Author's response

Title: Learning and calibrating per-location classifiers for visual place recognition
Authors: Gronat P., Sivic J., Pajdla T., Obozinski G.

We would like to thank the reviewers for their valuable comments and helpful suggestions on further improving the manuscript. We have carefully revised the manuscript and addressed the comments of reviewers. We first summarize the main changes in "Changes Summary" and then provide detailed responses to the reviewers' comments. For the reviewers' convenience, the main edits are highlighted in blue in the revised manuscript.

Changes Summary

Major changes:

- We have edited and significantly extended section 1. *Introduction* and section 2. *Related work* sections to more clearly position this submission with respect to the related work and outlined our main contributions.
- We have extended section 7 *Experimental setup and implementation details* to describe an additional dataset and details of obtaining additional positive examples.
- We have significantly extended section 8 *Results* to describe
    - Results of comparison with the method of (Knopp et al., ECCV 2010 [29]) included in Table 1.
    - Results on an additional dataset 24/7 Tokyo [50] (Included in a new Table 2 that now shows all Fisher vector results at one place).
    - New section 8.4 experimentally comparing with Linear discriminant analysis and PCA whitening baselines.
    - New section 8.5 describing the main failure modes.
    - New section 8.6 describing the scalability of our approach.

Minor changes:

- We have corrected BOW data points in Figure 9.
- Figure 1 has been resized to a single column version.
- Appendix has been extended.
- We have re-positioned figures and tables.

For the reviewer's convenience we attached a colored difference PDF between the previous paper version and this revision.

Reply to Reviewer #1:

This paper considers the problem of image geolocation, with a specific focus on street-level imagery (e.g. from Google Street View). Given a large collection of images each with a GPS tag, the paper proposes to build an exemplar SVM-type model for each individual training image, using other images (specifically, geospatially distant images that are visually similar) as negatives. But this introduces a number of problems, like that a huge number of SVM weight vectors need to be kept around, and that the outputs of all the SVMs need to be calibrated on some meaningful scale. The paper proposes ways of handling both of these problems, using a relatively straightforward but intuitive and effective statistical significance test for the calibration problem, and using a trick based on storing sparse support vectors instead of SVM weights for the former problem. Results are presented on StreetView images of Pittsburgh, using both bag of interest points and Fisher Vectors as features.

This manuscript is based on a CVPR 2013 paper by the same authors, but with substantial additional content, including much better results (mostly due to using Fisher Vectors), additional techniques to reduce memory use and computation time, and additional explanations, analysis, and discussion. I think the additional material here is clearly enough to warrant a journal version.

This is a good paper that should be accepted. The topic is interesting, the explanations are very clear, and the quality of writing is high. The techniques are intuitive and clean, if not particularly earth shattering. The experimental results are convincing. I really enjoyed reading the paper, more so than most any other manuscript I've reviewed recently.

I have a couple of relatively minor high-level complaints, most of which could be fixed relatively easily.

1.1

First, the intro states that one of the two novelties of the paper is "to cast the place recognition problem as a classification task and use the available geo-tags to train a classifier for each location in the database". But this isn't really a novelty, right? Others like [20] also build per-place classifiers using geotags. It would be good to be more precise -- there are plenty of contributions in this paper, but I think this one might be stated a little too generally.

Answer:

Thank you for pointing this out. To clarify this issue, we have extended the “Introduction” and “Related work” sections in order to provide wider context and more clearly distinguish our work from others. We have also modified the “Contribution” paragraph such that contribution of this work is defined more precisely.

In detail, Li et al., ICCV 2009 [30] addresses the problem of landmark classification where geo-tags are used for data collection and definition of landmarks clusters. However, their task is very different from the place recognition where the goal is to localize the query image anywhere on the map, not just into pre-defined set of landmarks. Hence, we must deal with the lack of the training data and relatively uniformly distributed imagery across the region of interest. Previous works did not tackle the place recognition problem as a classification task with single (very small number of) examples in the positive class. This formulation, which is new, leads to a non-trivial technical problem of classifier calibration, which we are solving in our work. We believe that our formulation has not been used for place recognition before.

1.2

Second, it's a little disappointing that the paper doesn't compare to results from other authors, especially since location estimation is by no means a new problem. The CVPR 2013 paper upon which this article is based [15], for example, does report a comparison to another author on the same dataset -- is there are a reason those results were dropped here? It seems like it would be a good thing to leave in.

Anyway, the authors go out of their way on Page 8 lines 29-41 to argue for why they are not testing on other datasets, saying that their "method is not directly to place recognition benchmarks that do not use GPS position as ground truth." I'm a bit confused here what argument the authors are trying to make.

Many (nearly all?) datasets do have GPS ground truth, e.g. [42], [20], [5], right? The domains are different, of course; for instance the Hays & Efros IM2GPS paper (which probably should be cited here, by the way) certainly had GPS ground truth. Perhaps the difference is that there it is hopeless to build per-location ground truth because they are trying to geolocate on a world-wide scale? But then other techniques like [20], which studies place recognition as a classification task with a discrete set of places, should pose no problem. Also, work in 3D reconstruction has also produced datasets with fine-grained location data, e.g. the Photo Tourism work of Snavely et al or the SFM with MRFs paper of Crandall et al (with the latter reporting quantitative camera position errors wrt ground truth).

Answer:
We have added the comparison with (Knopp et al., ECCV 2010 [29]) in Table 1. Please note the results are slightly different to the CVPR 2013 version of our work as we have now a slightly different query set. The different query set was the main reason to omit the original comparison, but we have now re-run our implementation of (Knopp et al.) on the new query set and reported the result.

The note that “the method is not directly applicable to place recognition benchmarks that do not use GPS position as ground truth” was targeted to the San Francisco landmark dataset [4] for which the ground truth, rather than by GPS, is defined based on cartographical id from a Geographical Information System. This definition introduces several problems, details are explained below in the comment (2.4). We have now removed the paragraph describing the San Francisco benchmark from the result section.

We agree that it is important to compare to other results and existing baselines. We have added an evaluation on the challenging 24/7 Tokyo data set that has appeared at CVPR 2015 and uses an evaluation based on GPS position.

We have also added citation to IM2GPS of Hays & Efros.

1.3
To be clear, I don't think an evaluation on another dataset is necessarily needed, because the current results are sufficiently convincing, at least for Street View imagery. But it's less good to try to argue for why one can't compare to existing baselines when that doesn't really seem true, especially because if anything, this argument paints the proposed approach in an unnecessarily bad light: if it's really true that this approach is not applicable to any existing benchmark, that implies that the paper may be trying to solve a problem that is unrealistically specific to a particular new task that the authors invented.

Answer:
We have removed the paragraph “Relation to other benchmarks” from the text and added evaluation on the 24/7 Toky benchmark (please see above).

1.4
Finally, as a very minor point, there are a few typos and grammar issues here and there that should be cleaned up in a final draft -- a few examples are below.

More detailed comments:

Page 1

- Ln 48: only few -> only a few
- Not a big deal, but Fig 1 takes up a huge amount of space, isn't very attractive, and doesn't really say very much. I wonder if a smaller, 1-column version wouldn't do?

Answer:

We have changed the figure to a single column version.

- minor style thing: probably change "street-view" to Street View, at least when referring specifically to the Google imagery.

Page 2

- ln 32, machine -> machines

Page 3:

- ln 3, "at the same time": maybe just remove this phrase -- I
was confused at first, thinking that you were talking about timestamps on photos.
- ln 20, "each linear SVM classifier learns a score sj": This is a little confusing; actually what is learned is the weight wj and bias bj, in order to accurately predict scores sj, right?
- ln 55 -- not sure what the "(2)" here is referring to -- is it referring to equation (2) or to some numeric value in the figure?

Page 5:

do not like it - ln 32 (4) -> in equation (4)
- ln 38 calibrate -> calibrated
- ln 51 much -> many
- I'm not sure that Algorithm 1 and 2 are really needed, given how straightforward the procedure is and how clearly it is explained in the text. On the other hand,
I guess it doesn't hurt if there is enough space, and perhaps authors already received feedback from someone that it is helpful for clarity.

Page 7:

- ln 21-22 "Using this representation, the memory footprint is sig- nificantly reduced." redundant with line 32, "As a result, the storage requirements are significantly reduced"
- ln 37-38: "However, we found this approach did not yield competitive results." Does this refer to uncompetitive sparsity (i.e. memory requirements) or uncompetitive accuracies of the resulting classifiers?

Answer:

We thank Reviewer 1 for all the comments. We have modified the text accordingly.

1.5

- Minor organizational suggestion: I think it might make sense to discuss how the Fisher vectors are stored first, since that is trivial (just storing the weight matrices). Then next discuss how this is not practical for bag-of-words and then talk about the proposed workaround of storing support vectors instead. The current organization is fine but sort of anti-climatic, since the way Fisher vectors are handled is just the trivial obvious thing to do.

Answer:

We have re-organized section 6 according to the suggestion.

1.6

Page 8:

- ln 24 "we, first" -> we first
- same sentence, random and, second, we -> random, and second, we - ln 17 then-normalized -> then normalized
- ln 50 this-normalized -> this normalized

Page 9:

- I'd suggest combining Tables 2 and 3 together, unless for some reason the numbers aren't comparable?

Answer:

We have merged Table 2 and Table 3 together and added new results on Tokyo 24/7 dataset.

Page 10:

- Ln 40: not perform -> not perform well

Page 13:

1.7a

- Fig 9: This curve is a little confusing and potentially misleading I think. the discussion and plot suggests that BOW has a fixed memory footprint whereas FV footprint is variable, but of course the # of words could be adjusted to change the space requirements, just like # of dimensions are adjusted for FVs. It seems that for some reason the authors did not think it was interesting or necessary to try other vocabulary sizes, which may very well be true, but it's misleading to imply that BOW has a fixed memory requirement in general.

Answer:

The memory footprint of the BOW representation is equal to the sum of (i) the memory footprint of the vocabulary and (ii) the memory footprint of the database. The memory footprint of the vocabulary is in practical situations negligible when compared to the memory footprint of the database. The memory footprint of the database depends on the number of features per image (assuming each feature in an image is quantized to a different visual word, which is almost true for large vocabularies). Hence, for practical situations (1M images with 2k features per image and 200k vocabulary size), the memory footprint of BOW is constant as a function of the size of vocabulary and linear as a function of the number of images in the database.

1.7b

Second, the FV curves seem to have 4 discrete points connected by lines, but I thought the FV descriptors were only tested at 3 dimensionalities (128, 512, 2048) [according to Pg 9 ln 59 and Pg 8 ln 9]? (Although pg 12 mentions 256-dimensional FVs so maybe these were tested too?)

Answer:

We have evaluated our experiments for 16k-dimensional FV on 25k Pittsburgh dataset. The results are now included in the submitted manuscript in Table 2. The value of 256 in discussion on Page 12 was interpolated from the graph in Figure 9. The text was modified to clarify this.

1.7c

Third, section 5 marketed the w-norm technique as being much more memory efficient than the p-val technique, but then from this figure it looks like there's not much difference? Even given the log scale on the x-axis, it looks like may a factor of ~20% or something.

Answer:

The p-val method requires to store the non-parametric calibration functions which, in our experimental setup, corresponds to 40% overhead w.r.t. the w-norm method (950MB vs 1350MB). As discussed in Section 5, the w-norm method is much more efficient compared to the p-val in terms of time complexity, not in terms of memory complexity. We have also noticed small inaccuracies in Figure 9 in BOW datapoints. The submitted manuscript contains a corrected plot.

1.7d

Since one of the key arguments is that the new techniques are more computationally efficient, it seems like a figure like Fig 9 that shows recall versus running time would also be useful. In many applications, runtime may matter much more than memory consumption.

Answer:

To provide general characterization of computational requirements of the methods, we analyze the dependence of both the offline and online phases of the image based localization on the size of image database in a new Section 8.6 “Scalability”. The actual computation times depend on details of implementations and hardware used, but we have also included example timings for a ballpark comparison.

Reply to Reviewer #2:

The article proposes an approach to visual place recognition, focusing on street images crawled from Google Streetview. The article casts this problem as one of exemplar classification: for each possible image in the dataset, annotated with geo-tag information, an exemplar linear classifier is learned. At test time, the query image is classified using these classifiers, and the localization information of the image with the largest score is propagated to the query image. As multiple classifiers need to compete, calibration of the scores is of high importance.

A preliminary version of this paper appeared in CVPR13. This article extends the conference article in several aspects, high lighting:

- a new form of calibration based on l2 normalization
- a memory-efficient classifier representation for BOW features
- experiments with Fisher vector descriptors
- Additional details, related work, pseudo-code, etc.

The technical additions with respect to the conference paper are not large, but overall I consider the additional material sufficient in terms of novelty.

The main strength of the paper is to recast the problem of visual place recognition, which is usually addressed as a large-scale instance-level retrieval, into a classification one, and to provide calibration schemes to make this.

However, the paper has several issues that should be addressed:

2.1

- Related work on visual place recognition: I found the related work on visual place recognition to be quite lackluster, in particular, in terms of the differences between the proposed approach and previous works. The article argues that one main difference is that previous works (including the previous version of this work [15], which may be a mistake?) focus on retrieval, while this work casts the problem as discriminative classification. Yet both views are tightly related, see next point.

Answer:

Thank you for pointing out the erroneous reference [15]. It was a typo and has been removed in the revised submission. We have also significantly extended the related work section. Please see below the detailed answers.

2.2(a)

- Retrieval vs classification: The distinction between retrieval and classification is not very clear to me in this context. After all, after computing the exemplar SVM, one can use the vector weights as an embedding of the dataset image that leads to a new vectorial representation (done eg in (Shrivastava et al.. SIGGRAPH Asia, 2011) [A], and consider the task as one of retrieving the dataset image nearest to the query in the embedded space. This is in fact consistent with the accuracy measures used during the experimental evaluation.

Answer:

We agree with the reviewer that many classification schemes can be viewed as a transformation of feature space followed by a nonlinear function, e.g. thresholding, to make the final decision. We also agree that retrieval, which uses the ordering in one-dimensional space after the transformation, can be seen as what is done in our paper. However, when comparing our approach to (Shrivastava et al.. SIGGRAPH Asia, 2011) [A], we find the following principal difference. In [A], a linear classifier is trained at query time (online), again and again for every query. This is computationally intensive process and is one of the main limitations of [A] (Please see section 5 “Limitations and Future Work” in [A]). In our case, we train a classifier for every database image beforehand in the off-line stage and hence make the on-line computation much faster. On the other hand, we need to be able to compare outcomes of different classifiers (feature transformations) to be able to compare different database images to the query. That calls for a classifier calibration, which we formulate and address in this work. We have added

this discussion into the paragraph “Per-exemplar support vector machines.” in the related work section.

2.2.(b)

Furthermore, in the case of learning the exemplar classifier using a squared Euclidean loss instead of a hinge loss, it is easy to show that the closed-form solution is equivalent to projecting the feature vectors with PCA and performing whitening, which, as shown for example by Jégou and Chum, ECCV 2012 [B], leads to much improved results in image retrieval. This paper also addresses the issue of normalization, and also suggests to l2 normalize the projected data.

Answer:

We agree with the reviewer that when using a squared Euclidean loss instead of hinge loss and neglecting SVM regularization, a per-exemplar linear classifier can be obtained by computing w and b by linear discriminant analysis (LDA) in closed form (details are, for example, in (Aubry et al., Transactions on Graphics, 2014 [4])). However, we are using a hinge loss as well as regularization and therefore this type of closed form solution cannot be used in our case. (Jegou and Chum, [B]) show the benefit of PCA whitening for relatively small BOW dimensions (up to 32k), combining multiple small BOW dictionaries and VLAD image representations. We now explicitly discuss the relation to linear discriminant analysis (LDA) and principal component analysis (PCA) in the related work section. We have also implemented linear discriminative analysis and PCA whitening and report the results in a new section “8.4 Comparison to linear discriminant analysis and whitening baselines”.

2.2.(c)

As one of the main distinctions between this work and previous ones is that it is cast as a classification one, this duality between the tasks should be discussed in more depth.

Answer:

We have revised the related work section to discuss this issue in more depth. In detail, the related work now discusses the relation of e-SVM, LDA and PCA whitening.

2.3

- Analysis of per-exemplar SVM and normalization of the weights. The appendix provides an analysis on why the exemplar SVM and the normalization of w are meaningful. The analysis focuses on the case of a true exemplar classifier, where only one positive sample is available. However, during the experimental evaluation, several positive samples per exemplar are in fact

used (see page 8, l 35-40, right column). Does the analysis still hold in this case? Additionally, more details about how exactly those additional positive samples are mined would be necessary.

Answer:
The analysis of the per-exemplar SVM shown in Appendix also holds for expanded positive examples as these extra positives have similar statistics to the original positive image (illumination, capturing conditions, the same camera, etc.). The aim of the analysis is to give an intuition what is the effect of regularization for per-exemplar SVM and why L2 re-normalization works. We have now modified the text in the first paragraph of the Appendix to clarify that the analysis also holds for expanded positives. Finally, we have added a new paragraph in Section 7.3 explaining how additional positive examples are obtained.

2.4
- Experimental evaluation. The method is tested on two datasets which, if I understand correctly, are in-house. Is this correct? Are there no public datasets where this method could be tested? Similarly, there is no comparison with any other work as far as I can see. No other method on visual place recognition could be fairly tested with this setup?

Answer:
We understand the call for comparison to other techniques on publicly available data sets. Therefore, we have added a new comparison to the revised manuscript. We now evaluate our method on the challenging 24/7 Tokyo dataset (Torii et al., CVPR 2015 [50]) with queries spanning very different illumination conditions. The results are described in Section 8 and shown in Table 2. The results demonstrate that proposed w-norm calibration method improves place recognition results over the Fisher vector baseline also on this new data.

However, we would also like to point out that the Pittsburgh research dataset that we use as a query set is not “in house”. The dataset is public, provided by Google on request for research purposes, please see e.g. [i]. It has also been used by other researchers, e.g. [ii, iii], to test image based localization. The database images used in our work were obtained from Google street-view website. Our exact set of images is now available upon email request at: http://www.di.ens.fr/willow/research/perlocation/

[i] Google company, “ICMLA 2011 StreetView Recognition Challenge”, http://www.icmla-conference.org/icmla11/challenge.htm”.

[ii] Le Barz, C., Thome, N., Cord, M., Herbin, S., & Sanfourche, M., "Global Robot Geo-localization Combining Image Retrieval and HMM-based Filtering." 6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles. 2014.
[iii] Le Barz, C., Thome, N., Cord, M., Herbin, S., & Sanfourche, M., "Exemplar based metric learning for robust visual localization."

We have also attempted to run on the San Francisco dataset [8]. However, the ground truth of San Francisco Dataset is not based on GPS but on estimated visibility and it is difficult to compare results of our approach with [8] on this data.

In detail, the San Francisco dataset [8] defines ground truth based on automatic labelling of buildings visible in each image using a GIS (Geographic Information System), rather than by GPS coordinates. In our analysis, this appears to be quite inaccurate providing a contrived “ground truth”. Our analysis shows that about 25% of query images are associated to a place that is further than 200m away from the actual query image position. In extreme cases, the distance is above 1300m.

Our method includes automatically mined far-away images into negative training sets. However, such images are often in the positive set using the San Francisco ground truth, which dramatically affects the measured performance. We considered creating a new GPS based ground truth for San Francisco data, but have finally decided not to as results with such as newly defined ground truth would not be comparable to any previous work. Instead, we have compared on the newly released 24/7 Tokyo dataset (Torii et al., CVPR 2015, [50]), which presents a new challenging benchmark across large changes in illumination and contains accurate positional ground truth for all queries.

2.5

- Bow vs Fisher vectors. It seems that FV clearly outperform the bag of word representations in terms of accuracy. Is there, in the context of place recognition, any advantage of the sparse bow over representations such as fisher vectors or CNN activation features? As it stands, there is currently a section on compression of bow features that does not truly have any relevance.

Answer:

We agree with the reviewer that the Fisher vector descriptors outperform the bag-of-visual-words representation. However, we kept the bag-of-visual-words representation in

the paper as we directly compare results (as requested by R1) with Knopp et al. ECCV 2010 [29] who is also based on the bag-of-visual-words model.

2.6

- Complementary of calibration methods. Are the two proposed calibration methods (l2 normalization of the weights and p-value) complementary? could the p-value calibration method be applied after the l2 normalization?

Answer:

Since e-SVM (with proper regularization) followed by L2 normalization yields a new image representation, one could hypothetically construct the p-val calibration functions from these new representations. However, the main drawback of the p-val method is its memory complexity. The main motivation behind developing calibration by re-normalization was to develop a scalable method providing comparable results, and combining the two methods would not be scalable.

2.7

- Comparison with standard data whitening. What are the results of applying PCA whitening + L2 norm instead of using an exemplar with a squared hinge loss? This would be interesting to see, as both representations are related.

Answer:

We have implemented and compared our approach with linear discriminative analysis and PCA whitening and report the results in a new section “8.4 Comparison to linear discriminant analysis and whitening baselines”.

Reply to Reviewer #3:

The paper proposes an approach for visual place recognition using per-location exemplar-SVMs. It is an extension of the authors' previous CVPR 2013 paper.

The authors cast the place recognition problem as a classification task and train per-location exemplar-SVMs, one for each unique location. When training an exemplar-SVM for a location, the positive is the image at that location, and the negatives are images that are geographically at least 200m away. When presented with a new test image, each exemplar-SVM is used to

compute a score on the test image, and the one that produces the highest score is used to predict the test image's location. Since the exemplar-SVMs outputs are not directly comparable, the authors propose two ways to calibrate the classifiers. The first uses the Neyman-Pearson framework for hypothesis testing; the main idea is to calibrate the classifiers so that each classifier rejects the same number of negative examples at the same level of the calibrated score. The second simply normalizes the classifier to have unit L2-norm. The authors also present a memory-efficient classifier representation for bag-of-words features that decomposes the W matrix (where each column is a learned exemplar-SVM weight vector) into the product of a sparse matrix of the original bag-of-words descriptors and a sparse matrix of the coefficients (since each exemplar-SVM weight vector can be represented as a linear combination of the support vectors, i.e., bag-of-words descriptors). The authors apply their approach to 25k and 55k Google Streetview images of Pittsburgh, covering roughly an area of 1.3x1.2 km^2. Results demonstrate that the proposed calibrations lead to significant improvement in location prediction compared to uncalibrated classifiers as well as the base features (SURF bag-of-words and SIFT fisher vectors).

[major comments]

Overall, the proposed approach is technically sound, and the writing is generally clear. I have a few suggestions that I think could improve the paper:

3.1

- I would like to see more discussion and justification as to why exemplar-SVMs are needed for visual place recognition. The main drawback of exemplar-SVMs is that they are computationally expensive for training as the number of classifiers scale linearly in the number of positive images. The authors have shown results on a relatively small dataset (25k and 55k images). I'm worried that this approach will not scale to larger regions (e.g., country-scale or world-scale); the authors in the intro state that "Google street-view of France alone contains more than 60 million panoramic images".
Could the authors comment on scalability to such large datasets?

Answer:
We have included a new section addressing the scalability of the method in Section 8.6 “Scalability”. In a brief summary, the exemplar-SVMs used in our work are linear SVMs. Hence the complexity of training for one database image is O(M), where M is the number of training examples. We use constant M(=500) and therefore the time complexity of training all classifiers

is linear w.r.t. the database size. When scaling up to the country-scale the bottleneck could be collecting the top hard negatives for each database image. However, even this could be done efficiently using approximate nearest neighbour search:
Jegou et al., Product Quantization for Nearest Neighbor Search, Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:33 , Issue: 1), 2010.

3.2

- Related to the above, I would like to see an experiment where the authors try using features computed with deep convolutional neural networks (pre-trained on ImageNet or MIT Places database). This could be done either with or without SVM training (i.e., if without, just taking L2 distance between the fc7 features). I think this would be a simple but interesting experiment to try, and it would be good to include discussion on what is observed.

Answer:
We have performed a simple experiment with off-the-shelf CNN descriptors on the Pittsburgh 25k dataset. We have extracted fc7 features of AlexNet style convolutional neural network pre-trained on ImageNet (using the MatConvnet Matlab library). As suggested above, we have extracted the 4096-dimensional fc7 descriptor from both the query and database images. Then we have measured the L2 distance between the descriptors in order to rank the database images for each query. The results are, however, not very good as shown in the table below:

Method: / Recall@K [%]	**1**	**2**	**5**	**10**	**20**
CNN fc7	0.2	0.3	0.8	1.5	2.5

We believe that this is because the FC7 CNN features are sensitive to object/scene categories and do not well discriminate between different places. We plan to investigate CNN features for place recognition as future work, but believe a more in-depth study of these results as well as investigation of how to make CNN descriptors work well for place recognition is beyond the scope of this paper.

3.3

- It would be good to show and discuss qualitative examples (similar to Figs 6 and 8) in which the baseline feature representations outperform the proposed approach. This would help the reader better understand the failure modes.

Answer:

We have examined improvement and failure cases of the w-norm method w.r.t. the Fisher vector baseline on the Pittsburght 25k datset. We analyzed cases for which the w-norm method increases the rank of the first true positive by the baseline and provide more detailed explanation of failure cases in new Section 8.5 “Analysis of improvement and failure cases”.

3.4

- p.10, L43-49: "When examining the results, ..." Could the authors please elaborate on this sentence? I am still unclear as to why the p-value calibration did not perform well for Fisher vectors.

Answer:

When examining the results we have observed that for bag-of-visual-words the cdf estimated on the database well represents the scores of (unseen) negative query images at test time. However, this is not the case for Fisher vectors, where the estimated cdf on the database does not represent well the scores of negative query images at test time. The scores of (unseen) negative query images often fall outside of the estimated cdf or at the very tail that is only sparsely sampled. As a result the estimated query image p-values for Fisher vectors are often over-confident and incorrect. We have included this explanation at the end of first paragraph of section 8.2.

3.5

[minor comments]

- The figure and table placements could be improved. Some of the figures appear a few pages after they are referred to in text.
- p.9, L28: remove space after "section 8.1"
- p.9, L45: table 2 -> table 1
- p.9, L52: figure 8 -> figure 6 - p.10, L50: figure 6 -> figure 8

Answer:

We have re-organized the placement of tables and figures in submitted manuscript.

IJCV manuscript No.
(will be inserted by the editor)

Learning and calibrating per-location classifiers for visual place recognition

Petr Gronát · Josef Sivic · Tomáš Pajdla · Guillaume Obozinski

Received: date / Accepted: date

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. This is a challenging task due to changes in viewpoint, imaging conditions and the large size of the image database. The contribution of this work is two-fold. First, we cast the place recognition problem as a classification task and use the available geotags to train a classifier for each location in the database in a similar manner to per-exemplar SVMs in object recognition. Second, as only a few positive training examples are available for each location, we propose two methods to calibrate all the per-location SVM classifiers without the need for additional positive training data. The first method relies on p-values from statistical hypothesis testing and uses only the available negative training data. The second method performs an affine calibration by appropriately normalizing the learnt classifier hyperplane and does not need any additional labelled training data. We test the proposed place recognition method with the bag-of-visual-words and Fisher vector image representations suitable for large scale indexing. Experiments are performed on three datasets: 25,000 and 55,000 geotagged street view images of Pittsburgh, and the 24/7 Tokyo benchmark containing 76,000 images with varying illumination conditions. The results show improved place recognition accuracy of the learnt image representation over direct matching of raw image descriptors.

Grants or other notes about the article that should go on the front page should be placed here. General acknowledgments should be placed at the end of the article.

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

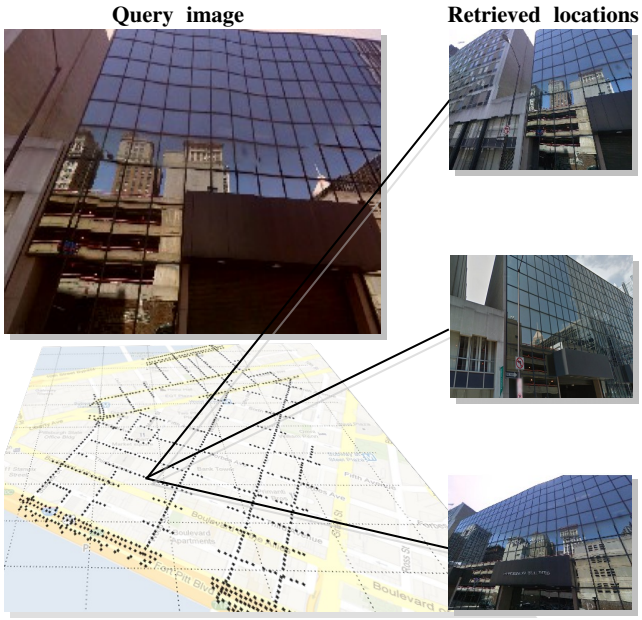


Fig. 1: The goal of this work is to localize a query photograph (left top) by finding other images of the same place in a large geotagged image database (right column). We cast the problem as a classification task and learn a classifier for each location in the database. We develop two procedures to calibrate the outputs of the large number of per-location classifiers without the need for additional labeled training data.

Keywords Place recognition · classifier calibration · Visual geo-localization

1 Introduction

Visual place recognition [11,29,43] is a challenging task as the query and database images may depict the same 3D structure (e.g. a building) from a different camera viewpoint, under different illumination, or the building can be partially occluded. In addition, the geotagged database may be very large. For example, we estimate that Google Street View of France alone contains more than 60 million panoramic images. It is, however, an important problem as automatic, accurate and fast visual place recognition would have many practical applications in robotics, augmented reality or navigation.

Similar to other work in large scale place recognition [11,29,43,52] and image retrieval [36,37,47,26], we describe each image by a set of local invariant features [5, 33] that are encoded and aggregated into a fixed-length single vector descriptor for each image. In particular, in this work we consider the sparse tf-idf weighted bag-of-visual-words representation [47,37] and the compact Fisher vector descriptors [26].

The resulting vectors are then normalized to have unit L_2 norm and the similarity between the query and a database vector is measured by their dot product. This representation has some desirable properties such as robustness to background clutter and partial occlusion. Efficient retrieval can then achieved using inverted file indexing [25].

While in image retrieval databases are typically unstructured collections of images, place recognition databases are usually structured: images have geotags, are localized on a map and depict a consistent 3D world. Knowing the structure of the database can lead to significant improvements in both speed and accuracy of place recognition. Examples include: (i) building an explicit 3D reconstruction of the scene [23,31,32]; (ii) constructing an image graph [6,38, 53], where images are nodes and edges connect close-by images on the map [51], or (iii) using the geotagged data as a form of supervision to select local features that characterize a certain location [29,43] or re-rank retrieved images [55].

In this work, we also take advantage of geotags as an available form of supervision and investigate whether the place recognition problem can be cast as a classification task. Learning visual classifiers has been investigated for landmark recognition [30] where consumer photographs were clustered into landmark classes based on geo-tags. In this work we wish to recognize individual street locations rather than a small number of landmarks, and as a consequence have only a few (1-5) photographs available for each place. In particular, we train a classifier *for each location on the map* in a similar manner to per-exemplar classification in object recognition [34].

This is beneficial as each classifier can learn which features are discriminative for a particular place. The classifiers are learnt offline. At query time, the query photograph is localized by transferring the GPS tag of the best scoring location classifier.

While learning classifiers for each place may be appealing, calibrating outputs of the individual classifiers is a critical issue. In object recognition [34], it is addressed in a separate calibration stage on a held-out set of training data. This is not possible in the place recognition set-up as only a small number, typically one to five, of positive training images are available for each location (e.g. Street View images viewing the same building facade). To address this issue, we propose two calibration methods. The first method relies on p-values from statistical hypothesis testing and uses only the available negative training data. The second method performs a simple affine calibration by appropriately normalizing the learnt classifiers and does not need any additional labelled calibration examples.

2 Related work

The task of geo-localizing a given input query photograph has recently received considerable attention. The output can be a coarse geo-localization on the level of continents and cities [14,22,27] or a name of the depicted landmark [30]. In this work we focus on visually recognizing the "same place" by finding an image in geo-tagged database that depicts the same building facade or street-corner as shown in the query [8,11,29,43,52,55].

This *visual place recognition* problem is typically treated as large-scale instance-level retrieval [11,8,29,43,52,55], where images are represented using local invariant features [33] encoded and aggregated into the bag-of-visual-words [10,47] or Fisher vector [26] representations. The image database can be further augmented by 3D point clouds [28], automatically reconstructed by large-scale structure from motion (SfM) [1,28], which enables accurate prediction of query image camera position [32,40]. In contrast, in this work we investigate learning a discriminative place-specific image representation. A similar idea has been recently explored in [6] who learn a graph-based discriminative representation for landmark image collections where typically many images are available for each landmark. In this work, we focus on street-level images such as Google Street View, which have greater coverage, but typically only one or a small number of images are available for each place. To address this issue we learn a discriminative re-weighting of the descriptor specific to each image in the database using per-exemplar support vector machine [34].

Per-exemplar support vector machines. The exemplar support vector machines (e-SVM) has been used in a number of visual recognition tasks including category-level recognition [34], cross-domain retrieval [45], scene parsing [48] or

as an initialization for more complex discriminative clustering models [14,46]. The main idea is to train a linear support vector machine (SVM) classifier from a single positive example and a large number of negatives. The intuition is that the resulting weight vector will give a higher weight to the discriminative dimensions of the positive training data point and will down weight dimensions that are non-discriminative with respect to the negative training data.

The exemplar support vector machine can be learnt at query time where the weight vector is used as a new query image representation [45]. However, this requires training a new classifier afresh for each query that is computationally very demanding. In this work, similar to [34] who learn per-exemplar object category representation, we learn per-exemplar classifiers for each place in the database off-line. A key advantage is that each per-exemplar classifier is trained independently and hence the learning can be heavily parallelized. The per-exemplar training brings, however, also an important drawback. As each classifier is trained independently a careful calibration of the resulting classifier scores is required [34].

Calibrating classifier scores. Several calibration approaches have been proposed in the literature (see [16] and references therein for a review). The most known consists of fitting a logistic regression to the output of the SVM [39]. This approach, however, has a major drawback as it imposes a parametric form (the logistic a.k.a. sigmoid function) of the likelihood ratio of the two classes, which typically leads to biased estimates of the calibrated scores. Another important calibration method is the isotonic regression [54], which allows for a non-parametric estimate of the output probability. Unfortunately, the fact that we have only a single positive example (or only very few of them which are almost identical, and which are all used for training) essentially prevents us from using any of these methods. To address these issues, we develop two classifier calibration methods that do not need additional labelled positive examples. Related to ours is also the recent work of Scheirer et al. [42] who develop a classifier calibration method for face attribute similarity search. Their method (discussed in more detail in section 4) also does not require labelled positive examples but, in contrast to us, uses a parametric model (the Weibull distribution) for the scores of negative examples.

Linear discriminant analysis and whitening. Our work is also related to linear discriminative transformations of feature space that have shown good performance in object recognition [17,21] and 2D-3D alignment [4,3]. While conceptually the idea of finding a discriminative projection of the original feature space is similar to our work, the main difference is in the used loss function. While we use hinge loss [44] to train the new discriminative representation of each place, [4,17,21] use the Euclidean loss. The advantage of using the Euclidean loss is that the discriminative projection can be computed in closed form. The resulting projection is tightly related to Linear Discriminant Analysis and whitening the feature space [4,17,21]. Such whitened representations have shown promise for image retrieval [24] or matching HOG [12] descriptors [13], however, we have found they do not perform well for place recognition.

Contributions. This paper has two main contributions. First, we cast the place recognition problem as a classification task where we use the available geo-tags as a weak form of supervision to train a classifier for each location in the database (section 3). These classifiers are subsequently used for ranking the database images at query time.

Second, as only a few positive training examples are available for each location, we propose two methods to calibrate all the per-location SVM classifiers without the need for additional positive training data. The first method (section 4) relies on p-values from statistical hypothesis testing. The second method (section 5) performs an affine calibration by appropriately normalizing the learnt decision hyperplane. We also describe a memory efficient classifier representation for the sparse bag-of-visual-word vectors (section 6) and experimentally demonstrate benefits of the proposed approach (section 7 and 8).

3 Per-location classifiers for place recognition

We are given an image descriptor $\mathbf{x}_j$, one for each database image j. This representation can be a sparse tf-idf weighted bag-of-visual-words vector [47] or a dense compact descriptor such as the Fisher vector (FV) [26]. The goal is to learn a score f_j for each database image j, so that, at test time, given the descriptor $\mathbf{q}$ of the query image, we can either retrieve the correct target image as the image j^* with the highest score

$$j^* = \arg\max_j f_j(\mathbf{q}) \tag{1}$$

or use these scores to rank candidate images and use geometric verification to identify the correct location in an n-best list. Instead of approaching the problem directly as a large multiclass classification problem, we tackle the problem by learning a per-exemplar linear SVM classifier [34] for each database image j. Similar to [29], we use the available geo-tags to construct the negative set $\mathcal{N}_j$ for each image j. The negative set is constructed so as to concentrate difficult negative examples, i.e. from images that are far away from the location of image j and similar to the target image as measured by the dot product between their feature vectors. The details of the construction procedure will be given in section 7. The positive set $\mathcal{P}_j$ is represented o single positive

example, which is $\mathbf{x}_j$ itself. Each SVM classifier produces a score s_j which is a priori not comparable with the score of the other classifiers. A calibration of these scores will therefore be key to convert them to comparable scores f_j. This calibration problem is more difficult than usual given that we only have a single positive example and will be addressed in section 4.

3.1 Learning per-location SVM classifiers

Each linear SVM classifier generates a score s_j of the form

$$s_j(\mathbf{q}) = \mathbf{q}^T \mathbf{w}_j + b_j \tag{2}$$

where $\mathbf{w}_j$ is a weight vector re-weighting contributions of individual visual words and b_j is the bias specific for image j. Given the training sets $\mathcal{P}_j$ and $\mathcal{N}_j$, the aim is to find a vector $\mathbf{w}_j$ and bias b_j such that the score difference between $\mathbf{x_j}$ and the closest neighbor from its negative set $\mathcal{N}_j$ is *maximized*. Learning the weight vector $\mathbf{w}_j$ and bias b_j is formulated as a minimization of the convex objective

$$\begin{aligned}\Omega(\mathbf{w}_j, b_j) = ||\mathbf{w}_j||^2 + C_1 \sum_{\mathbf{x}\in\mathcal{P}_j} h(\mathbf{w}_j^T \mathbf{x} + b_j) \\ + C_2 \sum_{\mathbf{x}\in\mathcal{N}_j} h(-\mathbf{w}_j^T \mathbf{x} - b_j),\end{aligned} \tag{3}$$

where the first term is the regularizer, the second term is the loss on the positive training data weighted by scalar parameter C_1, and the third term is the loss on the negative training data weighted by scalar parameter C_2. This is a standard SVM formulation (3), also used in exemplar-SVM [34]. In our case h is the squared hinge loss, which we found to work better in our setting than the standard hinge-loss. Parameters $\mathbf{w}_j$ and b_j are learned separately for each database image j in turn.

3.2 The need for calibrating classifier scores

Since the classification scores s_j are learned independently for each location j, they cannot be directly used for place recognition as in eq. (1). As illustrated in figure 2, for a given query $\mathbf{q}$, a classifier from an incorrect location (b) can have a higher score (eq. (2)) than the classifier from the target location (a). Indeed, the SVM score is a signed distance from the discriminating hyperplane and is a priori not comparable between different classifiers. This issue is addressed by calibrating scores of the learnt classifiers. The goal of the calibration is to convert the output of each classifier into a probability (or in general a "universal" score), which can be meaningfully compared across classifiers. In the following two sections we develop two classifier calibration methods that do not need additional labelled positive examples.

4 Non-parametric calibration of the SVM-scores from negative examples only

In this section we describe a classifier calibration method that exploits the availability of large amounts of negative data, i.e. images from other far away locations in the database. In particular, the method estimates the significance of the score of a test example compared to the typical score of the (plentifully available) negative examples. Intuitively, we will use a large dataset of negative examples to calibrate the individual classifiers so that they *reject the same number of negative examples* at each level of the calibrated score. We will expand this idea in detail using the concepts from hypothesis testing.

4.1 Calibration *via* significance levels

In the following, we view the problem of deciding whether a query image matches a given location based on the corresponding SVM score as a hypothesis testing problem. In particular, we appeal to ideas from the traditional frequentist hypothesis testing framework also known as Neyman-Pearson (NP) framework (see e.g. [7], chap. 8).

We define the null hypothesis as $H_0 = \{$the image is a random image$\}$ and the alternative as $H_1 = \{$the image matches the particular location$\}$. The NP framework focuses on the case where the distribution of the data under H_0 is well known, whereas the distribution under H_1 is not accessible or too complicated to model, which matches perfectly our setting.

In the NP framework, the *significance level* of a score is measured by the p-value or equivalently by the value of the cumulative density function (cdf) of the distribution of the negatives at a given score value. The cdf is the function F_0 defined by $F_0(s) = \mathbb{P}(S_0 \leq s)$, where S_0 is a random variable corresponding to the scores of negative data (see figure 3 for an illustration of the relation between the cdf and the density of the function). The cdf (or the corresponding p-value[1]) is naturally estimated by the empirical cumulative density function $\hat{F}_0$, which is computed as:

$$\hat{F}_0(s) = \frac{1}{N_c} \sum_{n=1}^{N_c} 1_{\{s_n \leq s\}}, \tag{4}$$

where $(s_n)_{1\leq n\leq N_c}$ are the SVM scores associated with N_c negative examples used for calibration. Note that no positive examples are involved in the construction of the cumu-

[1] The notion most commonly used in statistics is in fact the p-value. The p-value associated to a score is the quantity $\alpha(s)$ defined by $\alpha(s) = 1 - F_0(s)$; so the more significant the score is, the closer to 1 the cdf value is, and the closer to 0 the p-value is. To keep the presentation simple, we avoid the formulation in terms of p-values and we only talk of the probabilistic calibrated values obtained from the cdf F_0.

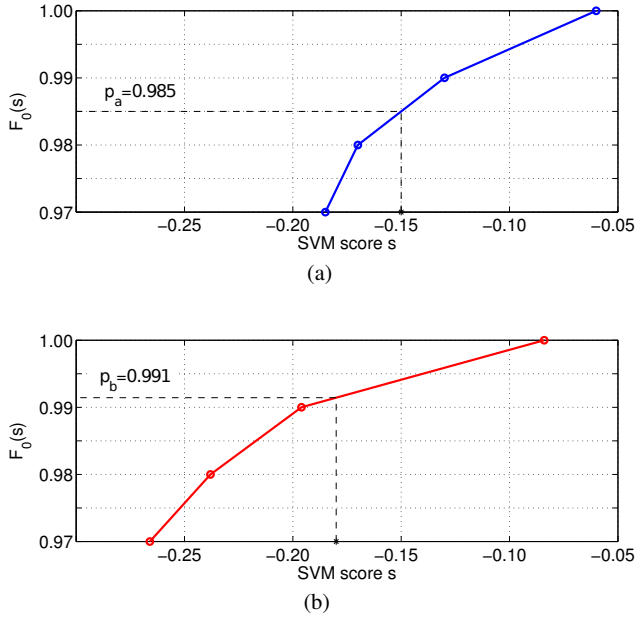


Fig. 2: **An illustration of the proposed normalization of SVM scores for database images.** In each plot, the x-axis shows the raw SVM score. The y-axis shows the calibrated output. For the given query, the raw SVM score of image (b) is lower than for image (a), but the calibrated score of image (b) is higher than for image (a).

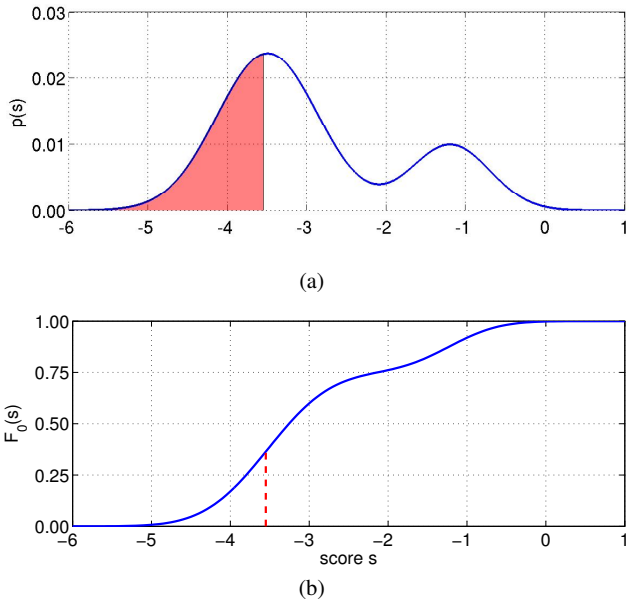


Fig. 3: **Cumulative density function.** Illustration of the relation between (a) the probability density of the random variable S_0 modeling the scores of the negative examples and (b) the corresponding cumulative density function $F_0(s) = \mathbb{P}(S_0 \leq s)$.

lative density function. $\hat{F}_0(s)$ is the fraction of the negative examples used for calibration (ideally held out negative examples) that have a score below a given value s. Computing $\hat{F}_0$ exactly would require to store all the SVM scores for all the calibration data for all classifiers, so in practice, we only keep a fraction of the larger scores. We also interpolate the empirical cdf between consecutive datapoints so that instead of being a staircase function it is a continuous piecewise linear function such as illustrated in figure 2. Given a query, we first compute its SVM score s_q and then compute the calibrated probability $f(q) = \hat{F}_0(s_q)$. We obtain a similar calibrated probability $f_j(q)$ for each of the SVMs associated with each of the target locations, which can now be ranked. Two other examples of score calibration functions are shown in figure 6 in section 8. Note that while figure 2 illustrates only few points on the cdf, the two plots in figure 6 show a complete cdf that contains on the order of $25k$ data points. Note also that the two cumulative density functions in figure 6 are similar but not identical.

4.2 Summary of the calibration procedure

For each trained place-specific classifier s_j we construct the empirical cumulative density function (4) of scores of the negative examples and keep only its top K values. This can be done offline and the procedure is summarized in Algorithm 1. At query time, given a query image descriptor $\mathbf{q}$, we compute the uncalibrated classification score $s_j(\mathbf{q})$ and then use the stored cdf values to compute the calibrated score $f_j(\mathbf{q})$. This procedure is performed for each database image j and is summarized in Algorithm 2. Finally, the best candidate database image is selected by equation (1). Alternatively, candidate database images can be also ranked according to the calibrated score.

Algorithm 1 P-value calibration: offline stage

Input: $\mathcal{X}$ ... column wise matrix of image descriptors
$\mathbf{w}_j, b_j$ ... learnt SVM weights and biases
Output: $\hat{F}_{0j}$ ... calibration functions

procedure P-VALUE CALIBRATION
  $N \leftarrow$ database size
  $\mathcal{X} \leftarrow$ descriptor matrix of negative examples
  for $\forall j \in 1 \ldots N$ **do**
    $N_c \leftarrow$ number of negative examples
    $\mathbf{w} \leftarrow$ learned SVM weight for image j
    $b \leftarrow$ learned SVM bias for image j
    $\boldsymbol{\sigma} \leftarrow \mathbf{w}^T\mathcal{X} + b$
    Compute the cdf:
    $\mathbf{s_j} \leftarrow$ sorted $\boldsymbol{\sigma}$ in descending order
    $\hat{F}_{0j} \leftarrow [N_c \ldots 0]/N_c$

Algorithm 2 P-value calibration: online stage

Input: $\mathbf{q}$ ... query image descriptor
$\mathbf{w}_j, b_j$ ... learnt SVM weights and biases
$\hat{F}_{0j}$ ... learnt calibration function
Output: $f_j(\mathbf{q})$ ... calibrated score

procedure CALIBRATING SCORES
  $\mathbf{q} \leftarrow$ query image descriptor
  $N \leftarrow$ database size
  for $\forall j \in 1 \dots N$ **do** // for each database image
    $\mathbf{w} \leftarrow$ learned SVM weight for image j
    $b \leftarrow$ learned SVM bias for image j
    $\hat{F}_0 \leftarrow \hat{F}_{0j}$ // Empirical cdf
    $\mathbf{s} \leftarrow \mathbf{s_j}$ // Corresponding sorted scores
    $s_q \leftarrow \mathbf{q}^T\mathbf{w} + b$ // compute uncalibrated classifier score
    Find n such that $s_n \leq s_q < s_{n+1}$
    Compute the interpolated empirical cdf value:
    $\hat{F}_0(s_q) \approx \hat{F}_0(s_n) + \frac{s_q - s_n}{s_{n+1} - s_n}(\hat{F}_0(s_{n+1}) - \hat{F}_0(s_n))$.
    $f_j(\mathbf{q}) = \hat{F}_0(s_q)$ // output the calibrated score

4.3 Discussion

It should be noted that basing the calibration only on the negative data has the advantage that we privilege precision over recall, which is justified given the imbalance of the available training data (many more negatives than positives). Indeed, since we are learning with a single positive example, intuitively, we cannot guarantee that the learned partition of the space will generalize well to other positives, whose scores in the test set can potentially drop significantly. By contrast, since we are learning from a comparatively large number of negative examples, we can trust the fact that new negative examples will stay in the half-space containing the negative training set, so that their scores are very unlikely to be large. Our method is therefore based on the fact that we can measure reliably how surprising a high score would be if it was the score of a negative example. This exactly means that we can control false positives (type I error) reasonably well but not false negatives (type II error or equivalently the power of our test/classifier), exactly as in the Neyman-Pearson framework.

An additional reason for not relying on positive examples for the calibration in our case is that (even if we had sufficiently many of them) the positive examples that we collect using location and geometric verification from the geotagged database typically have illumination conditions that are extremely similar to each other and not representative of the distribution of test positives which can have very different illuminations. This is because of the controlled nature of the capturing process of geotagged street-level imagery (e.g. Google Street View) used for experiments in this work. Close-by images are typically captured at a similar time (e.g. on the same day) and under similar imaging conditions.

Scheirer et al. [42] propose a method, which is related to ours, and calibrate SVM scores by computing the corresponding cdf value of a Weibull distribution fitted to the top negative scores. The main difficulty is that the Weibull model should be fitted only to the tail of the distribution of the negatives, which is in general difficult to identify. As a heuristic, Scheirer et al. propose to fit the Weibul model to false positives (i.e. the negative samples classified incorrectly as positives). But in our case, most of the exemplar SVMs that we are training have zero false positives in a held out set, which precludes the application of their method.

Finally, we should remark that we are not doing here calibration in the same sense of the word as the calibration based on logistic regression (or isotonic regression), since logistic regression estimates the probability of making a correct prediction by assigning a new data to class 1, while we are estimating how unlikely it would be for a negative example to have such a high score. The calibration with either methods yields "universal" scores in the sense that they are comparable from one SVM to another, but the calibrated values obtained from logistic regression are not comparable to the values obtained from our approach.

5 Affine calibration by normalizing the classification hyperplane

The non-parametric calibration method described in the previous section has two computational disadvantages, which make it hard to scale-up to very large datasets. First, the method requires storing the non-parametric model of the calibration function for each learned classifier. This has memory complexity of $O(NK)$, where N is the number of images (classifiers) in the database and K the number of stored elements of the non-parametric model. For typical values of $K = 1000$ and $N = 1$M this would require additional 4GB of memory, comparable to the size of the inverted index itself. Second, computing the cumulative density function requires applying all N learnt classifiers to the entire set of negative examples, which has also size N. As a result computing the cdf has complexity $O(N^2)$, which becomes quickly infeasible already for datasets with N larger than $100,000$.

To address these issues we first describe an affine calibration model that calibrates the classifier score with a simple linear function defined by only two parameters: its slope and offset, greatly reducing the required storage. Second, we show that the parameters of the affine calibration function can be obtained by normalizing the learnt classification hyper-plane without applying the classifiers on the negative data and thus bringing down the computational complexity to $O(N)$. As a result, computing and storing the calibration functions becomes feasible for very large datasets with 1M images.

5.1 Affine calibration model

Using the affine calibration model we transform the uncalibrated score $s_j(\mathbf{q})$ of query $\mathbf{q}$ with a linear function

$$f_j(\mathbf{q}) = \alpha_j s_j(\mathbf{q}) + \beta_j, \quad (5)$$

where $f_j(\mathbf{q})$ is the output calibrated score, and α_j and β_j are scalar calibration parameters specific to each classifier j. In this work we use linear classifiers, hence substituting for $s_j(\mathbf{q})$ the linear classifier from (2) results also in a linear calibrated classifier

$$f_j(\mathbf{q}) = \widetilde{\mathbf{w}}_j^T \mathbf{q} + \widetilde{b}_j, \quad (6)$$

where $\widetilde{\mathbf{w}}_j = \alpha_j \mathbf{w}_j$ and $\widetilde{b}_j = \alpha_j b_j + \beta_j$. Note that the calibrated classifier (6) has the same form as the original classifier (2) and hence this representation does not require any additional storage compared to storing the original classifier. The question remains how to set the parameters α_j and β_j of the calibration function (5), which is discussed next.

5.2 Calibration by normalization

Parameters of the affine calibration function (5) could be learnt from negative training data in a similar manner to, for example, [3]. We have tried to estimate the parameters in a similar manner by fitting a line to the tail of the cdf, however this procedure did not yield satisfactory results. In addition, as discussed above, in our case this requires running all N classifiers on all N images, which is prohibitive for large datasets. Instead, we have found that a good calibration can be obtained by normalizing the learnt hyperplane $\mathbf{w}$. In particular, we set

$$\alpha_j = \frac{1}{||\mathbf{w}_j||}, \quad (7)$$

$$\beta_j = -b_j \alpha_j, \quad (8)$$

where $\mathbf{w}_j$ and b_j are the parameters of the learnt SVM hyperplane for location j and $||\mathbf{w}||$ is the L_2 norm of $\mathbf{w}$. Given this choice of α_j and β_j the calibrated classification score (6) reduces to

$$f_j(\mathbf{q}) = \frac{1}{||\mathbf{w}_j||} \mathbf{w}_j^T \mathbf{q} = \widetilde{\mathbf{w}}_j^T \mathbf{q}. \quad (9)$$

The intuition is that when $\mathbf{q}$ is L_2 normalized, equation (9) is equivalent to computing the normalized dot-product between vectors $\mathbf{q}$ and $\mathbf{w}$. This was found to work well in image retrieval [47] or matching whitened HOG descriptors [13]. In this work we investigate whether this intuition about descriptor matching can be used as a form of calibration for the learnt place-specific classifier. Note that this form of calibration by normalization is scalable to very large datasets as it (i) requires only $O(N)$ computations offline to pre-compute the calibration parameters for each of the N learnt classifiers (equations (7) and (8)), and (ii) does not need any additional storage or computation at query time as the calibration parameters can be included in the classifier (6). In Appendix we examine the per-exemplar SVM cost and give an additional intuition why calibration by renormalization works.

6 Memory efficient classifier representation

We learn a linear discriminative classifier with weight vector $\mathbf{w}_j$ and bias b_j for each image j in the database. These classifier parameters become the new representation for each image. In this section we discuss how the classifier parameters can be stored in a memory efficient manner that is amenable for indexing. The goal is to apply all the learnt classifiers to the query descriptor $\mathbf{q}$

$$\mathbf{s} = \mathbf{q}^T \mathcal{W} + \mathbf{b}, \quad (10)$$

where $\mathcal{W}$ is $d \times N$ matrix storing all the learnt $\mathbf{w}_j$ classifiers as columns, $\mathbf{b}$ is a $1 \times N$ vector storing all the learnt bias values b_j, $\mathbf{q}$ is the input query descriptor, $\mathbf{s}$ is a $1 \times N$ vector of output scores for all classifiers in the database, N is the number of images in the database and d is the dimensionality of the image representation. As discussed in detail in section 7 we investigate two different image representations: (i) the compact Fisher vectors [26] and (ii) the bag-of-visual-word vectors [47]. The learnt classifiers for these two image descriptors have different statistics and require different methods for storing and indexing. Next, we discuss the classifier representations for the two types of image representations.

Fisher vectors: The Fisher vector descriptors are not sparse, but have a relatively low-dimension $d \in \{128, 512, 2048\}$ hence it is possible to store directly the (non-sparse) matrix $\mathcal{W}$ containing the learnt classifier parameters $\mathbf{w}$. In this work we exhaustively compute the classifier scores for all images in the database (given by equation (10)) using efficient (but exact) matrix-vector multiplication routines. However, this computation can be further sped-up using product quantization indexing as described in [25].

Bag-of-visual-words: In the bag-of-visual-words representation, each image is represented by a high dimensional vector $\mathbf{x}$, where the dimensionality d is typically $100,000$, but the vector is very sparse with only about 2,000 non-zero entries. The learnt $\mathbf{w}_j$ are of the same (high) dimension d but are *not sparse*. As a result, directly storing the learnt classifiers becomes quickly infeasible. To illustrate this, consider a database of $N = 1,000,000$ images. Storing the original descriptors with about 2,000 non-zero entries for each image would take around 8GB. However, directly storing the

learnt non-sparse $100,000 \times 1,000,000$ matrix $\mathcal{W}$ would require 400GB of memory. To address this issue we have developed an alternative indexing structure taking advantage of the dual form of the linear classifier as a sparse linear combination of a small number of support vectors [44]. The key observation is that the number of support vectors k is significantly lower then dimensionality d of the original image descriptor. In the following we omit index j for clarity. In detail, we represent each $\mathbf{w}$ by its corresponding coefficients α_i of the linear combination of the support vectors (individual image descriptors) $\mathbf{x}_i$ such that

$$\mathbf{w} = \sum_i \alpha_i \mathbf{x}_i = \mathcal{X} \cdot \boldsymbol{\alpha}, \tag{11}$$

where α_i, the elements of vector $\boldsymbol{\alpha}$, are coefficients of the linear combination of the training data points $\mathbf{x}_i$ and the matrix $\mathcal{X}$ contains (as columns) descriptors of the entire database. Note that the vector $\boldsymbol{\alpha}$ is sparse and the number of non-zero elements depends on the number of support vectors k.

As a result, matrix $\mathcal{W}$ containing all learned classifier weights can be expressed in the dual form as

$$\mathcal{W} = \mathcal{X}\mathcal{A}, \tag{12}$$

where $\mathcal{X}$ is the (sparse) matrix of the bag-of-visual-words image descriptors and $\mathcal{A}$ is the (sparse) matrix of α coefficients, where each column corresponds to vector $\boldsymbol{\alpha}$ from (11). Instead of storing all (non-sparse) weight vectors $\mathcal{W}$, which has memory complexity $O(dN)$ where d (= 100,000) is the dimensionality of the image representation and N is the size of the database, we store two sparse matrices $\mathcal{X}$ and $\mathcal{A}$, which has memory complexity $O(mN + kN)$ where m (=2,000) is the number on non-zero elements in the original bag-of-visual-word descriptors, and k is the typical number of support vectors. In our case k is about the size of the training data which is around 500. As a result, the storage requirements are significantly reduced. For example, for a database of 1M images the dual representation requires only about 10 GB of storage compared to 400GB for directly storing classifiers $\mathcal{W}$. Note that sparsity can be imposed directly on the learnt classifiers $\mathbf{w}$ by appropriate regularization [44]. However, we found this approach did not yield competitive results in terms of accuracy.

7 Experimental setup and implementation details

In this section we describe the experimental datasets, outline the two types of used image descriptors, and finally give implementation details of the classifier learning procedure.

7.1 Image datasets

Experiments are performed on two datasets, the Pittsburgh place recognition dataset [20] and the Tokyo 24/7 dataset [50].

Pittsburgh dataset. The first dataset contains Google Street View panoramas downloaded from the Internet covering an area of 1.3×1.2 km^2 of the city of Pittsburgh (U.S.). Similar to [8], we generate for each panorama 12 overlapping perspective views corresponding to two different elevation angles $4°$ and $28°$ to capture both the street-level scene and the building façades. This results in a total of 24 perspective views each with $90°$ FOV and resolution of 960×720 pixels. In this manner we generate two versions of this dataset. The first version covers a smaller area and contains 25k perspective images. The second larger dataset contains 55k images. As a query set with known ground truth GPS positions, we use 8999 panoramas from the Google Street View research dataset [18], 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. We generate a test query set such that we first select a panorama at random, and second, we generate a perspective image with a random orientation and random elevation pitch. This way we synthesize 4,000 query test images. Both the query and database images are available upon request at [19].

24/7 Tokyo dataset. The 24/7 Tokyo dataset [50] contains Google Street View panoramas downloaded from the Internet covering an area of 1.6×1.6 km^2 of the city of Tokyo. The dataset contains 76k perspective views. The query set contains 315 query images from 105 distinct locations captured by different types of camera phones. This dataset is very challenging as each location is captured at three different times: during day, at sunset and during night. The dataset is available upon request at [49].

7.2 Image descriptors

We perform experiments with two types of image descriptors: the sparse high-dimensional bag-of-visual-word vectors [47] and the compact (not-sparse) Fisher vectors [26]. Details of each are given next.

Bag-of-visual-word representation. We extract SURF descriptors [5] for each image and learn a vocabulary of 100k visual words by approximate k-means clustering [37] from a subset of features from $5,000$ randomly selected database images. Then, a tf-idf weighted vector [47] is computed for each image by assigning each descriptor to the nearest cluster center. Finally, all database vectors are normalized to have unit L_2 norm.

Fisher vectors. Following [26] we project the extracted 128-dimensional rootSIFT [2] descriptors to 64 dimensions using PCA. The projection matrix is learnt on a set of descriptors from 5,000 randomly selected database images. This has also the effect of decorrelating the rootSIFT descriptor. The 64-dimensional descriptors are then aggregated into Fisher vectors using a Gaussian mixture model with $N = 256$ components, which results in a $2 \times 256 \times 64 = 32,768$-dimensional descriptor for each image. The Gaussian mixture model is learnt from descriptors extracted from 5,000 randomly sampled database images. The high-dimensional Fisher vector descriptors are then projected down to dimension using PCA learnt from all available images in the database. The resulting low dimensional Fisher vectors are then normalized to have unit L2-norm, which we found to be important in practice.

7.3 Parameters of per-location classifier learning

To learn the exemplar support vector machine for each database image j, the positive and negative training data are 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 m; (ii) sorting the images by decreasing value of similarity to image j measured by the dot product between their respective descriptors; (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 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 descriptor $\mathbf{x}_j$ of the target image j.

We found that for the bag-of-visual-words representation it was useful to further expand [9] positive training set by close by images that view the same scene structures. These images can be identified by geometric verification [37] as follows. We first build a graph where each image in the database represents a node and an edge represents a spatial adjacency in the world. An edge is present if the positions of the two images are within $50m$ of each other. Then, we score each edge by the number of geometrically verified matches [37]. Finally, we remove edges with score below a threshold of $t_m = 40$ matches. It is worth noting that the graph contains many isolated nodes. This typically indicates that the viewpoint change between two adjacent panoramas is large. For each image in the database, we include between zero and five extra positive examples that are directly connected in the graph.

For the support vector machine classifier (SVM) training we use `libsvm` [15]. We use the same C_1 and C_2 parameters for all per-exemplar classifiers, but find the optimal value of the parameters for each image representation by a cross-validation evaluating performance on a held out query set.

For the calibration by re-normalization, we L_2 normalize the learned $\mathbf{w}_j$ using equation (9) and use this normalized vector as the new image descriptor $\mathbf{x}'_j$ for image j. At query time we compute the descriptor $\mathbf{q}$ of the query image and measure its similarity score to the learnt descriptors $\mathbf{x}'_j$ for each database image by equation (1).

For the p-value calibration, we take the learnt classifier for each database image j and compute its SVM score for all other database images to construct its empirical cumulative density function (4). We keep only the top 1,000 values that, in turn, represent the calibration function. At query time, given the query descriptor $\mathbf{q}$, we compute the SVM score (2) for each database image j, and compute its calibrated SVM score f_j (4).

8 Results

We evaluate the proposed per-location classifier learning approach on two different image descriptors: the bag-of-visual-words model (section 8.1) and Fisher vectors (section 8.2). We also compare the recognition accuracy of the two learnt representations relative to their compactness measured by their memory footprint (section 8.3). Finally, we compare results to linear discriminant analysis (LDA) and whitening baselines (section 8.4), outline the main failure modes (section 8.5) and discuss the scalability of our method (section 8.6). Since the ground truth GPS position for each query image is available, for each method we measure performance using the percentage of correctly recognized queries (Recall) similarly to, e.g., [8,29,41]. We deem the query as correctly localized if at least one of the top K retrieved database images is within 20 meters from the ground truth position of the query.

8.1 Bag-of-visual-words model

Results for the bag-of-visual-words image representation are shown in table 1. Learning per-location classifiers with either calibration method (*p-val* and *w-norm*) clearly improves over the standard bag-of-visual-words baseline (BOW) that does not perform any learning. In addition, both calibration methods significantly improve over the learnt SVM classifiers without any calibration (BOW SVM no calib) underscoring the importance of calibration for the independently learnt per-location classifiers. In table 1, we also compare performance to our implementation of the confuser suppression approach (Conf. supp.) of [29] that, in each database image, detects and removes features that frequently appear at other far-away locations (using parameters $t = 3.5$ and

Method:	25k Pittsburgh				
recall@K [%]	1	2	5	10	20
BOW SVM no calib.	6.4	8.1	13.5	17.5	20.5
BOW	28.7	35.7	45.8	53.7	61.5
BOW Conf. supp [29]	29.6	37.3	48.9	59.3	69.2
BOW w-norm	31.8	38.7	49.7	**60.2**	**69.4**
BOW p-val	**33.0**	**40.3**	**50.2**	58.7	66.4

Table 1: **Evaluation of the learnt bag-of-visual-words representation on the Pittsburgh 25k dataset.** The table shows the fraction of correctly recognized queries (recall@K) for the different values of $K \in \{1, 2, 5, 10, 20\}$ retrieved database images. The learnt representations (BOW w-norm and BOW p-val) outperform the raw bag-of-visual-words baseline (BOW) as well as the learnt representation without calibration (BOW SVM no calib).

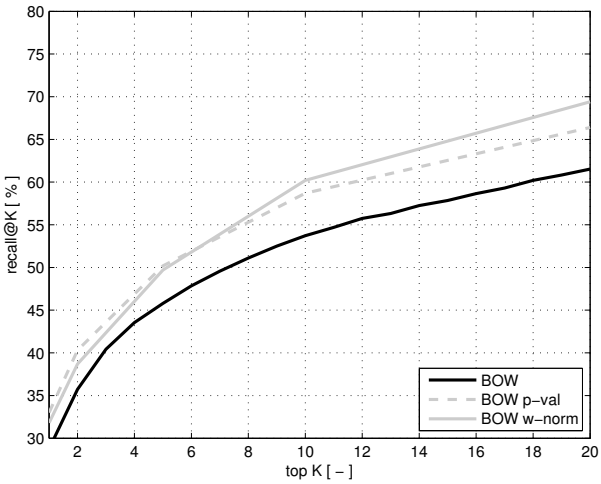


Fig. 4: **Evaluation of the learnt bag-of-visual-words representation on the Pittsburgh 25k [20] dataset.** The graph shows the fraction of correctly recognized queries (recall@K, y-axis) vs. the number of top K retrieved database images for the raw bag-of-visual-words baseline (BOW) and the learnt representation with two different calibration methods (p-val and w-norm).

$w = 70$). The results show an improvement by our method, specially at recall@1.

Inspecting the detailed plots in figure 4 we further note that the *p-val* calibration performs slightly better than the *w-norm* calibration for shorter top K shortlists but this effect is reversed for larger K. This could be attributed to the fact that the *p-val* calibration uses the negative data to control false positive errors, but has less control over false negatives, as discussed in section 4.3.

In figure 6 we visualize the learnt SVM weights on BOW for *p-val*. We visualize the contribution of each feature to the SVM score for the corresponding query image. Red circles represent features with negative weights while green circles correspond to features with positive weights. The area of each circle is proportional to the contribution of the corresponding feature to the SVM score. For instance for the left figure notice that the correctly localized queries (c) contain more green colored features than queries from other places (b) and (a). Query (b) gets a high score because the building has orange and white stripes similar to the the sun-blinds of the bakery, which are features that also have large positive weights in the query image (c) of the correct place. In the top row we visualize the calibration of raw SVM score for three different queries. The calibration function of the target image j is shown in the blue and the corresponding SVM scores of the three queries are denoted by red circles. Notice that both images (b) and (c) have high calibrated score even their respective SVM score was different.

Finally, examples of correctly and incorrectly localized queries are shown in figure 9.

8.2 Fisher vectors

Results of the proposed per-location learning method for the Fisher vector image representation for different dimensions are shown in table 2 and figure 5. Similar to bag-of-visual-words, the learnt representation (w-norm) significantly improves the place recognition performance over the baseline Fisher vector (FV) matching without learning. The improvements are consistent across different lengths of shortlist K and for different dimensionality of the Fisher vector representation. We report results only for the w-norm calibration as we found that the p-val calibration did not perform well for the learnt Fisher vector classifiers (top 1 recall of 25.3% compared to baseline performance of 33.6% for dimension 128). When examining the results we have observed that for bag-of-visual-words the cdf estimated on the database well represents the scores of (unseen) negative query images at test time. However, this is not the case for Fisher vectors where estimated cdf on the database does not represent well the scores of negative query images at test time. The scores of (unseen) negative query images often fall outside of the estimated cdf or at the very tail that is only sparsely sampled. As a result the estimated query image p-values for Fisher vectors are often over-confident and incorrect. Notice that the proposed per-location learning method consistently improves performance over the raw Fisher vector descriptors on the larger Pittsburgh 55k dataset and the challenging 24/7 Tokyo dataset (76k images). Examples of correctly and incorrectly localized queries are shown in figure 8. Next, we compare the performance of the two learnt representations relative to their memory footprints.

recall@K [%]	1	2	5	10	20
Method / Dataset:	25k Pittsburgh				
FV128	33.6	41.8	52.0	59.8	67.7
FV128 w-norm	**38.3**	**47.5**	**57.7**	**65.8**	**72.7**
FV512	44.3	51.7	61.4	68.7	75.2
FV512 w-norm	**47.6**	**55.4**	**65.1**	**72.4**	**78.8**
FV2048	46.9	54.1	63.8	70.5	76.8
FV2048 w-norm	**50.2**	**57.3**	**67.0**	**73.8**	**78.0**
FV16384	45.3	54.1	63.8	69.4	75.3
FV16384 w-norm	**49.3**	**56.0**	**65.9**	**72.5**	**76.8**
	55k Pittsburgh				
FV128	10.9	14.1	20.2	26.4	33.2
FV128 w-norm	**13.5**	**17.7**	**25.0**	**31.8**	**39.0**
FV512	17.3	21.1	28.4	34.2	40.3
FV512 w-norm	**19.8**	**25.1**	**32.7**	**38.7**	**46.0**
FV2048	19.2	23.5	29.9	35.2	41.9
FV2048 w-norm	**20.8**	**25.9**	**33.1**	**38.7**	**45.9**
	24/7 Tokyo				
FV128	14.2	20.0	27.9	34.2	41.5
FV128 w-norm	**16.9**	**22.0**	**29.6**	**37.2**	**44.8**
FV512	35.2	40.3	43.8	48.2	57.1
FV512 w-norm	**36.1**	**42.0**	**46.8**	**52.8**	**61.4**
FV2048	37.4	42.5	48.5	53.9	58.7
FV2048 w-norm	**42.9**	**46.7**	**52.8**	**58.8**	**66.7**
FV4096	42.9	46.3	54.0	59.0	64.8
FV4096 w-norm	**44.3**	**47.1**	**54.7**	**61.1**	**66.5**

Table 2: **Evaluation of the learnt Fisher vector representation on the Pittsburgh [20] and 24/7 Tokyo [50] datasets.** The table shows the fraction of correctly recognized queries (recall@K) for the different values of $K \in \{1, 2, 5, 10, 20\}$ retrieved database images. The learnt Fisher vector representation (FV *w-norm*) consistently improves over the standard Fisher vector matching baseline (FV) for all target dimensions.

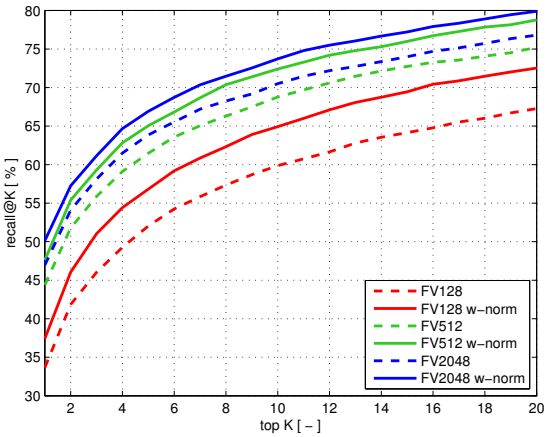


Fig. 5: **Evaluation of the learnt Fisher vector representation on the Pittsburgh 25k [20] dataset.** The graph shows the fraction of correctly recognized queries (recall@K, y-axis) vs. the number of top K retrieved database images for the raw Fisher vector baseline (FV) for different dimensions compared to the learnt representation (*w-norm*). Note the consistent improvements over all lengths of shortlist K for all dimensions.

8.3 Analysis of recognition accuracy vs. compactness

Here we analyze the recognition accuracy of the learnt representations vs. their compactness measured by their memory footprint on the Pittsburgh 25k image dataset. Ideally, we wish to learn a more compact representation, that still improves the recognition accuracy. However, usually there is a trade-off between the discriminative power of the representation and its size, where having a more compact representation reduces the recognition accuracy [26]. We observe a similar behavior but our learnt representation results in a higher recognition accuracy for a given size, or alternatively, significantly reduces the size of the representation for a given accuracy. The results are summarized in figure 7. The figure shows the recognition performance (y-axis) for the different dimensionality of the Fisher vector representation, which corresponds to different memory footprints (x-axis). For example, for $d = 128$ the memory footprint is about 24 MB, whereas for $d = 2048$ the memory footprint is about 384 MB. Note that the x-axis is in log-scale. The bag-of-visual-words representation has a fixed dimensionality (and fixed memory footprint) and hence each bag-of-visual-words method is shown only as a single point on the graph. For Fisher vectors, the results demonstrate that for a given level of accuracy (y-axis) the proposed method learns a more compact (lower-dimensional) representation (x-axis). For example, our learnt 128-dimensional descriptor (memory footprint of 24 MB) achieves a similar accuracy (around 65%) as the 256-dimensional raw Fisher descriptor (memory footprint of 51MB, interpolated from figure 7). This corresponds to 50% memory savings for the same level of recognition performance. Note that similar to [26], we observe decrease in performance at high-dimensions for both the FV baseline and our method. The results also demonstrate the benefits of using the compact FV descriptors compared to the bag-of-visual-words baseline achieving significantly better recognition accuracy for a similar memory footprint.

8.4 Comparison to linear discriminant analysis (LDA) and whitening baselines

We have compared our method to the linear discriminant analysis (LDA) [4,21,17] and whitening [24] baselines. Results are reported on the Pittsburgh 25k dataset. The LDA baseline finds a discriminative linear projection of the feature space by minimizing an Euclidean loss rather than hinge loss used in our work. In detail, following [4] we have used all available database to learn the covariance matrix and used calibrated LDA score (see [4] eq. 11) to obtain a classifier

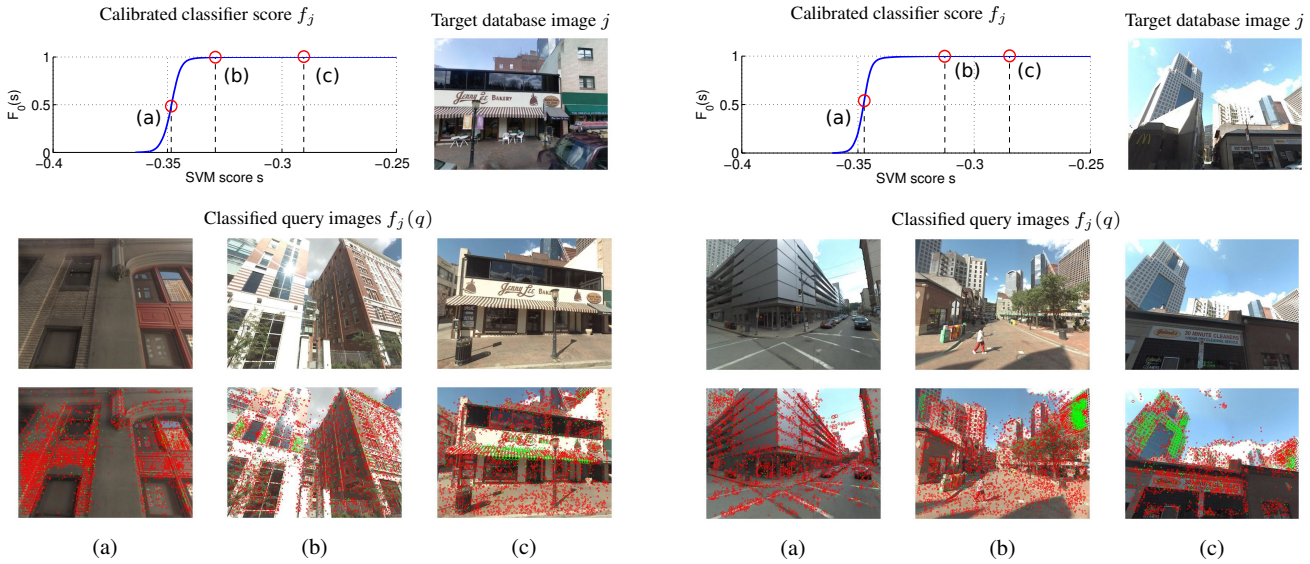


Fig. 6: **A visualization of learnt feature weights for two database images. In each panel:** *first row:* (Right) Target database image j. (Left) Cumulative density function (or calibrated score) learnt for the SVM scores of the corresponding classifier f_j; three query images displayed on the *second row* are represented by their SVM scores and cdf values $F_0(s)$, denoted (a)-(c) on the graph. *Third row:* A visualization of the contribution of each feature to the SVM score for the corresponding query image. Red circles represent features with negative weights while green circles correspond to features with positive weights. The area of each circle is proportional to the contribution of the corresponding feature to the SVM score. Notice that the correctly localized queries (c) contain more green colored features than queries from other places (b) and (a). Please note also that the calibration *cdf*s in the left and right panel are similar but not identical. *Left panel:* Query (b) gets a high score because the building has orange and white stripes similar to the the sun-blinds of the bakery, which are features that also have large positive weights in the query image (c) of the correct place. *Right panel:* Query (b) is in fact also an image of the same location with a portion of the left skyscraper in the target image detected in the upper left corner and the side of the rightmost building in the target image detected in the top right corner. Both are clearly detected by the method as indicated by a large quantity of green circles in the corresponding regions.

for each database image. We have applied the LDA method on the 128-dimensional Fisher vector descriptor but have obtained significantly worse performance (31.9% for recall@1) than our method (recall@1 of 38.3%). We believe the better performance of our method can be attributed to (i) the use of hinge-loss and (ii) training using the top scoring hard negative examples that are specific for each place. Next, we compare results to PCA compression followed by whitening as suggested in [24]. For bag-of-visual-words, we follow [24] and compare performance to PCA whitening to a target dimension of 4096. We have observed performance drop compared to the raw bag-of-visual-words baseline (28.7% to 26.1% for recall@1). We hypothesize this could be attributed to the large dictionary size used in our work (100k), whereas [24] report improved results for single dictionary whitening only for dictionaries of up to 32k visual words. Finally, we have also applied PCA whitening on Fisher vector descriptors of dimensions 128, 512 and 2048, but have not observed significant improvements over the baseline raw descriptors. In fact, for the highest dimension (2048) we have observed a performance drop (49.6% to 41.3%), which could be attributed to amplification of low-energy noise as also reported in [24].

8.5 Analysis of improvement and failure cases

We have examined the improvement and failures of the w-norm method w.r.t. the Fisher vector baseline on the Pittsburgh 25k dataset. We analyzed the cases for which the w-norm method improves the rank of the first true positive compared to the baseline and for which the rank of the first true positive is made worse. In detail, considering a shortlist of the size 20 we want to identify when: (i) an image with the rank of 20 is attracted into the short list (improvement), and (ii) an image with the rank of ≤ 20 is pushed out of the short list using our method (failure).

We observe that in 237 cases a low-ranked true positive image by the baseline (ranked $40-70$) is attracted into the short list by the w-norm method, resulting in an improvement. Note that in many other cases our method improves ranking but here we only count the cases for which the base-

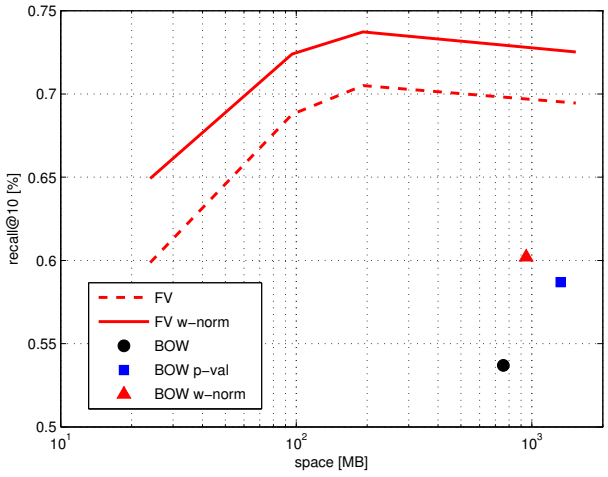


Fig. 7: **The recognition performance vs. the memory requirements for the Pittsburgh 25k dataset.** The fraction of correctly localized queries at the top 10 retrieved images (y-axis) vs. the memory footprint (x-axis) for the different representations. For Fisher vectors, the learnt descriptor (FV *w-renorm*) clearly outperforms the raw Fisher vector descriptor (FV) for all dimensions corresponding to different memory footprints (x-axis). Learnt per-location representations for the bag-of-visual-words model (BOW p-val and BOW w-norm) also improve performance over the raw bag-of-visual-words (BOW). However, the Fisher vectors provide much better recognition performance for the same memory footprint.

line method does not have any true positives in the top 20 short-list. On the other hand, in 39 cases our method makes the results worse and removes a correct image from the top 20 short-list but typically only to top 40.

Finally, we observe that aggravation typically occurs on hard examples where the baseline performance is already bad. When visually inspecting the failure cases we observed that our method typically fails on queries containing a big portion of the sky clouds or vegetation, narrow streets or tunnels and sometimes retrieves images capturing the same building from a different viewpoint or a larger distance.

8.6 Scalability

In the offline stage, our method collects hard negative examples for each location in the database, which are consequently used to train exemplar SVM classifiers. As only a constant number of examples (1-5 positives and 500 negatives) is used to train each per-location classifier the overall complexity of training is linear, $O(N)$, i.e. we need to train one classifier (with constant training time) for each of N images in the database. The bottleneck of the offline stage is collecting the negative examples that is quadratic $O(N^2)$ in the database size. In other words, for each of N database images, we need to find the top 500 most similar negatives among all N database images. However, we believe that even finding negatives can be scaled-up to very large datasets with standard compression techniques such as product quantization (PQ) [25] combined with sub-linear approximate nearest neighbor search [35].

At query time our method needs to compute the calibrated e-SVM score (equation (2)) of the query for each image in the database. In the case of w-norm method the calibration weights can be included in the classifier weight matrix, as discussed in section 6. For the p-val calibration method, each e-SVM score must be calibrated using K stored values of the non-parametric CDF model. This requires a search for the two closest values and subsequent interpolation, which yields complexity of $O(N \log K)$. Since K is only a constant both the w-norm and p-val methods have a linear time complexity (in the size of the database) at query time but with different constants. However, in practice the constant in the p-val method can be quite large. The actual running time per query is $340ms$ for the bag-of-visual-words representation with p-val calibration and $3ms$ for the FV128 descriptor with w-norm calibration. Both timings are on the 25k Pittsburgh dataset on a desktop with CPU Intel Xenon E5 using a single thread. Hence in practice, the p-val method may be scalable only to medium size datasets. For the w-norm method, the query time can be further sped-up using sub-linear approximate nearest neighbor search [35] on compressed descriptors [25], making the method scalable to very large datasets.

Fig. 8: **Examples of correctly and incorrectly localized queries for the learnt bag-of-visual-words representation.** Each example shows a query image (left) together with correct (green) and incorrect (red) matches from the database obtained by learnt bag-of-visual-words representation *p-val* method (top) and the standard bag-of-visual-words baseline (bottom). Note that the proposed method is able to recognize the place depicted in the query image despite changes in viewpoint, illumination and partial occlusion by other objects (trees, lamps) and buildings. Note also that bag-of-visual-words baseline is often confused by repeating patterns on facades and walls.

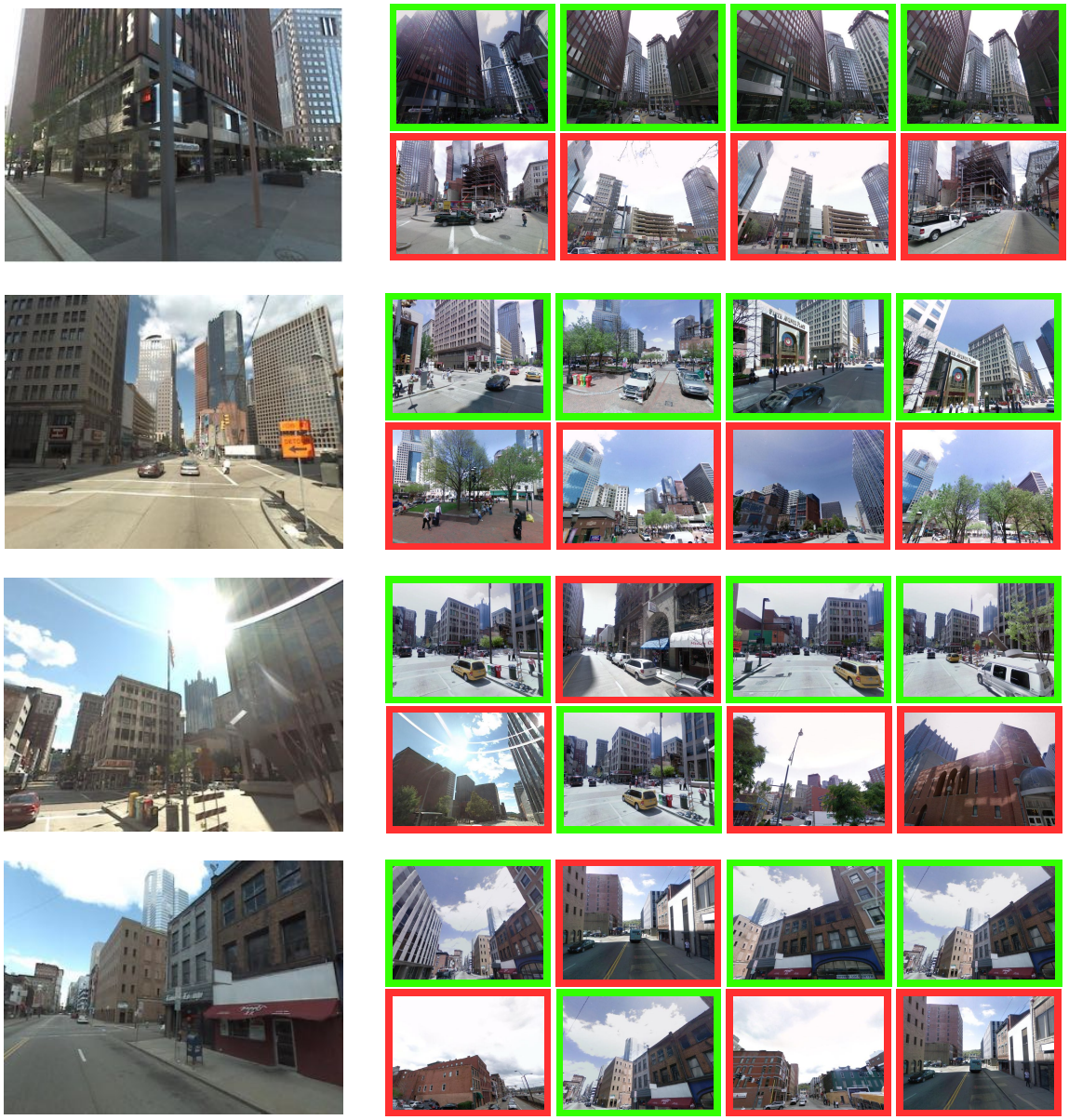


Fig. 9: **Examples of correctly and incorrectly localized queries for the learnt Fisher vector representation.** Each example shows a query image (left) together with correct (green) and incorrect (red) matches from the database obtained by the learnt Fisher vector representation *w-norm* method (top) and the standard Fisher vector baseline (bottom) for dimension 128. Note that the proposed method is able to recognize the place depicted in the query image despite changes in viewpoint, illumination and partial occlusion by other objects (trees, lamps) and buildings. Note that the baseline methods often finds images depicting the same buildings but in a distance whereas our learnt representation often finds a closer view better matching the content of the query.

9 Conclusions

We have shown that place recognition can be cast as a classification problem and have used geotags as a readily-available supervision to train an ensemble of classifiers, one for each location in the database. As only few positive examples are available for each location, we have developed two procedures to calibrate the output of each classifier without the need for additional positive training data. We have shown that learning per-location representations improves the place recognition performance over the raw bag-of-visual-words and Fisher vector matching baselines. The developed calibration methods are not specific to place recognition and can be useful for other per-exemplar classification tasks, where only a small number of positive examples are available [34].

Acknowledgements This work was supported by the MSR-INRIA laboratory, the EIT-ICT labs, Google, the ERC project LEAP and the EC project PRoViDE FP7-SPACE-2012-312377. Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL or the U.S. Government.

References

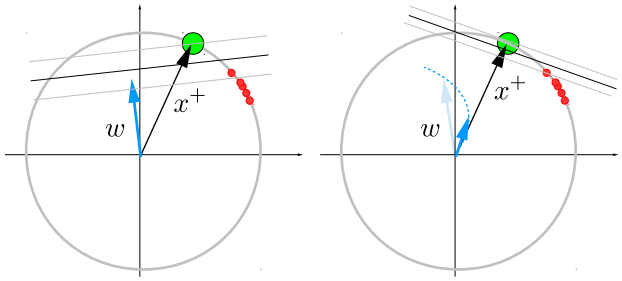
1. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building Rome in a day. In: ICCV, pp. 72–79 (2009)
2. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: IEEE PAMI (2012)
3. Aubry, M., Maturana, D., Efros, A., Russell, B., Sivic, J.: Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In: CVPR (2014)
4. Aubry, M., Russell, B., Sivic, J.: Painting-to-3D model alignment via discriminative visual elements. ACM Transactions on Graphics (2014)
5. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: ECCV (2006)
6. Cao, S., Snavely, N.: Graph-based discriminative learning for location recognition. In: CVPR, pp. 700–707. IEEE (2013)
7. Casella, G., Berger, R.: Statistical inference (2001)
8. 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)
9. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
10. 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)
11. Cummins, M., Newman, P.: Highly scalable appearance-only SLAM - FAB-MAP 2.0. In: Proceedings of Robotics: Science and Systems. Seattle, USA (2009)
12. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: CVPR (2005)
13. Doersch, C., Gupta, A., Efros, A.A.: Mid-level visual element discovery as discriminative mode seeking. In: NIPS (2013)
14. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.A.: What makes paris look like paris? SIGGRAPH **31**(4) (2012)
15. 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)
16. Gebel, M., Weihs, C.: Calibrating classifier scores into probabilities. Advances in Data Analysis pp. 141–148 (2007)
17. Gharbi, M., Malisiewicz, T., Paris, S., Durand, F.: A Gaussian approximation of feature space for fast image similarity. Tech. rep., MIT (2012)
18. Google: Icmla 2011 streetview recognition challenge. URL http://www.icmla-conference.org/icmla11/challenge.htm
19. Gronát, P.: Project webpage: Learning and calibrating per-location classifiers for visual place recognition. URL http://www.di.ens.fr/willow/research/perlocation/
20. Gronat, P., Obozinski, G., Sivic, J., Pajdla, T.: Learning and calibrating per-location classifiers for visual place recognition. In: CVPR (2013)
21. Hariharan, B., Malik, J., Ramanan, D.: Discriminative decorrelation for clustering and classification. In: ECCV (2012)
22. Hays, J., Efros, A.A.: im2gps: estimating geographic information from a single image. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2008)
23. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR (2009)
24. Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In: ECCV, pp. 774–787 (2012)
25. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(1), 117–128 (2011)
26. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE PAMI **34**, 1704–1716 (2012)
27. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A., Hertzmann, A.: Image sequence geolocation with human travel priors. In: IEEE 12th International Conference on Computer Vision (ICCV), pp. 253–260 (2009)
28. Klingner, B., Martin, D., Roseborough, J.: Street view motion-from-structure-from-motion. In: ICCV (2013)
29. Knopp, J., Sivic, J., Pajdla, T.: Avoidng confusing features in place recognition. In: ECCV (2010)
30. Li, Y., Crandall, D., Huttenlocher, D.: Landmark classification in large-scale image collections. In: ICCV (2009)
31. Li, Y., Snavely, N., Huttenlocher, D.: Location recognition using prioritized feature matching. In: ECCV (2010)
32. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3d point clouds. In: ECCV (2012)
33. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
34. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV (2011)
35. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36** (2014)
36. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
37. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
38. Philbin, J., Sivic, J., Zisserman, A.: Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. IJCV (2010)
39. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers (1999)

40. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: ECCV (2012)
41. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: Proc. BMVC (2012)
42. Scheirer, W., Kumar, N., Belhumeur, P.N., Boult, T.E.: Multi-attribute spaces: Calibration for attribute fusion and similarity search. In: CVPR (2012)
43. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR (2007)
44. Scholkopf, B., Smola, A.: Learning with kernels. MIT press, Cambridge (2002)
45. Shrivastava, A., Malisiewicz, T., Gupta, A., Efros, A.A.: Data-driven visual similarity for cross-domain image matching. In: SIGGRAPH ASIA (2011)
46. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV (2012)
47. 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
48. Tighe, J., Lazebnik, S.: Finding things: Image parsing with regions and per-exemplar detectors. In: CVPR (2013)
49. Torii, A.: Project webpage: 24/7 place recognition by view synthesis. URL http://www.ok.ctrl.titech.ac.jp/ torii/project/247/
50. Torii, A., Arandjelović, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: CVPR (2015)
51. Torii, A., Sivic, J., Pajdla, T.: Visual localization by linear combination of image descriptors. In: IEEE Workshop on Mobile Vision (2011)
52. Torii, A., Sivic, J., Pajdla, T., Okutomi, M.: Visual place recognition with repetitive structures. In: CVPR (2013)
53. Turcot, P., Lowe, D.: Better matching with fewer features: The selection of useful features in large database recognition problem. In: WS-LAVD, ICCV (2009)
54. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: ACM SIGKDD (2002)
55. Zamir, A., Shah, M.: Accurate image localization based on google maps street view. In: ECCV (2010)

Appendix

In section 8 we show that the simple calibration by normalization often results in surprisingly good place recognition performance without the need for any additional positive or negative calibration data. In this appendix, we give a possible explanation why this simple calibration works. We focus on the case of a single positive training example, i.e. when training set $\mathcal{P} = \mathbf{x}^+$, which is the typical case for place recognition where only one positive example is available for each place. The analysis holds also for the case of multiple expanded positive examples as in our case the positive examples are coming from the same database of Street View images, and hence have very similar statistics (illumination, capturing conditions, the same camera, etc.).

In particular, we first analyze the SVM objective and show that the learnt hyperplane $\mathbf{w}$ can be interpreted as a new descriptor $\mathbf{x}^*$ that replaces the original positive example $\mathbf{x}^+$ and is re-weighted to increase its separation from the negative data. Second, we show that when $\mathbf{x}^*$ is normalized, i.e. $\mathbf{x}^* = \frac{\mathbf{w}}{||\mathbf{w}||}$, the dot-product $\mathbf{q}^T\mathbf{x}^*$ corresponds to measuring the cosine of the angle between the (normalized) query descriptor $\mathbf{q}$ and the new descriptor $\mathbf{x}^*$, which was found to work well in the literature for descriptor matching, as discussed in section 5.2. The two steps are given next.



(a) $C_2 > 0$ (b) $C_2 \to 0$

Fig. 10: **An illustration of the effect of decreasing parameter C_2 in the exemplar support vector machine objective.** The positive exemplar $\mathbf{x}^+$ is shown in green. The negative data points are shown in red. All training data is L2 normalized to lie on a hyper-sphere. (a) For $C_2 > 0$, the normal $\mathbf{w}$ of the optimal hyper-plane moves away from the direction given by the positive example $\mathbf{x}^+$ in a manner that reduces the loss on the negative data. (b) As the parameter C_2 decreases the learnt $\mathbf{w}$ becomes parallel to the positive training example $\mathbf{x}^+$ and its magnitude $||\mathbf{w}||$ goes to 0.

Analysis of per-exemplar SVM objective

For a single positive example $\mathcal{P} = \mathbf{x}^+$, the per-exemplar SVM objective (3) can be written as

$$\Omega(\mathbf{w}, b) = ||\mathbf{w}||^2 + C_1 \cdot h(\mathbf{w}^T\mathbf{x}^+ + b) + C_2 \sum_{\mathbf{x}\in\mathcal{N}} h(-\mathbf{w}^T\mathbf{x} - b). \quad (13)$$

In the following, we analyze the objective (13) and provide intuition why *re-normalized* weight vector $\mathbf{w}$ can be interpreted as a new descriptor. In particular, we show first that when the weight C_2 of the negative data in objective (13) goes to zero the learnt normalized $\widetilde{\mathbf{w}}$ is identical to the original positive training data point $\mathbf{x}^+$. Second, when $C_2 > 0$, the learnt vector $\widetilde{\mathbf{w}}$ moves away from the positive vector $\mathbf{x}^+$ to increase its separation from the negative data. The two cases are detailed next.

Case I: $C_2 \to 0$. The goal is to show that when the weight C_2 of the negative data in objective (13) goes towards zero, the resulting hyperplane vector $\mathbf{w}$ is parallel with the vector of positive training descriptor $\mathbf{x}^+$. When $\mathbf{w}$ is normalized to have unit L2 norm the two vectors are identical. First, let us decompose $\mathbf{w}$ into parallel and orthogonal part with respect to the positive training data point $\mathbf{x}^+$, i.e. $w = \mathbf{w}^\perp + \mathbf{w}^{||}$, where $(\mathbf{w}^\perp)^T\mathbf{x}^+ = 0$. Next, we observe that when the

weight of the negative data diminishes ($C_2 \rightarrow 0$), any non-zero component $\mathbf{w}^{\perp}$ will increase the value of the objective. As a result, for $C_2 \rightarrow 0$ the objective is minimized by $\mathbf{w}^{||}$, i.e. the optimal $\mathbf{w}$ is parallel with $\mathbf{x}^{+}$.

In detail, for $\mathbf{w} = \mathbf{w}^{\perp} + \mathbf{w}^{||}$, the objective (3) can be written as

$$||\mathbf{w}^{\perp} + \mathbf{w}^{||}||^2 + C_1 \cdot h\left((\mathbf{w}^{\perp} + \mathbf{w}^{||})^T \mathbf{x}^{+} + b\right) + C_2 \sum_{\mathbf{x} \in \mathcal{N}} h\left(-(\mathbf{w}^{\perp} + \mathbf{w}^{||})^T \mathbf{x} - b\right). \tag{14}$$

Note that the orthogonal part $\mathbf{w}^{\perp}$ does not change the value of the second term in (14) because $(\mathbf{w}^{\perp} + \mathbf{w}^{||})^T \mathbf{x}^{+} = (\mathbf{w}^{||})^T \mathbf{x}^{+}$, and hence (14) reduces to

$$||\mathbf{w}^{\perp} + \mathbf{w}^{||}||^2 + C_1 \cdot h\left(\mathbf{w}^{||T} \mathbf{x}^{+} + b_j\right) + C_2 \sum_{\mathbf{x} \in \mathcal{N}} h\left(-(\mathbf{w}^{\perp} + \mathbf{w}^{||})^T \mathbf{x} - b\right). \tag{15}$$

In the limit case as $C_2 \rightarrow 0$ any non-zero component $\mathbf{w}^{\perp}$ will increase the value of the objective (15). This can be seen by noting that the third term vanishes when $C_2 \rightarrow 0$ and hence the objective is dominated by the first two terms. Further, the second term in (15) is independent of $\mathbf{w}^{\perp}$. Finally, the first term will always increase for any non-zero value of $w^{\perp}$ as $||\mathbf{w}^{\perp} + \mathbf{w}^{||}||^2 \geq ||\mathbf{w}^{||}||$ for any $\mathbf{w}^{\perp} \neq 0$.

As a result, in the limit case when $C_2 \rightarrow 0$ the optimal $\mathbf{w}$ is parallel with $\mathbf{x}^{+}$. Note also, that when C_2 is exactly equal to zero, $C_2 = 0$, the optimal $\mathbf{w}$ vanishes, i.e. the objective (15) is minimized by trivial solution $||\mathbf{w}|| = 0$ and $b = -1$. The effect of decreasing the parameter C_2 is illustrated in figure 10.

Case II: $C_2 > 0$. When the weight C_2 of the negative data in the objective (15) increases the direction of the optimal w will be different from $\mathbf{w}^{||}$ and will change to take into account the loss on the negative data points. Explicitly writing the hinge-loss $h(x) = \max(1 - x, 0)$ in the last term of (15), we see that $\mathbf{w}$ will move in the direction that reduces $\sum_{\mathbf{x} \in \mathcal{N}} \max\left(1 + \mathbf{w}^T x + b, 0\right)$, i.e. that reduces the dot product $\mathbf{w}^T \mathbf{x}$ on the negative examples that are active (support vectors).

The need for normalization of $\mathbf{w}$

Above we have shown that the learnt hyperplane $\mathbf{w}$ moves away from the positive example $\mathbf{x}^{+}$ in a manner that reduces the loss on the negative data. The aim is to use this learnt vector $\mathbf{w}$ as a new descriptor $\mathbf{x}^{*}$ replacing the original positive example $\mathbf{x}^{+}$. However, we wish to measure the cosine of the angle between the the new descriptor $\mathbf{x}^{*}$and the query image $\mathbf{q}$. This is equivalent to the normalized dot product, hence the vector $\mathbf{w}$ needs to be normalized.