A. Freytag, E. Rodner, T. Darrell, and J. Denzler
Exemplar-specific Patch Features for Fine-grained Recognition. GCPR 2014
(c) Copyright by Springer. The final publication is available at link.springer.com

Exemplar-specific Patch Features for Fine-grained Recognition

Alexander Freytag[1⋆], Erik Rodner[1⋆], Trevor Darrell[2], and Joachim Denzler[1]

[1]Computer Vision Group, Friedrich Schiller University Jena, Germany
[2]UC Berkeley ICSI & EECS, United States

Abstract. In this paper, we present a new approach for fine-grained recognition or subordinate categorization, tasks where an algorithm needs to reliably differentiate between visually similar categories, *e.g.*, different bird species. While previous approaches aim at learning a single generic representation and models with increasing complexity, we propose an orthogonal approach that learns patch representations specifically tailored to every single test exemplar. Since we query a constant number of images similar to a given test image, we obtain very compact features and avoid large-scale training with all classes and examples. Our learned mid-level features are built on shape and color detectors estimated from discovered patches reflecting small highly discriminative structures in the queried images. We evaluate our approach for fine-grained recognition on the CUB-2011 birds dataset and show that high recognition rates can be obtained by model combination.

1 Introduction

Nearly all image categorization and object recognition systems are built on the general idea of using a set of patch detectors and their outputs as proper features for classification. This is the case for bag-of-features approaches, *e.g.*, [22], where the set of detectors is usually referred to as codebook or vocabulary of local features, it holds for recent deep convolutional networks, *e.g.*, [21,31], where detectors are convolutional filter masks learned on different levels, and it also holds for discriminative patch techniques, *e.g.*, [24,18,8]. In all cases, a single set of these detectors is learned and the intra-class variability needs to be tackled by choosing a large number of sparsely coded detectors [22] or by stacking them together into several layers [21].

In contrast, we show how to build patch-based feature representations specifically for each test example. Our approach allows focusing features and the set of patch detectors on the task of differentiating objects with a similar pose and similar global appearance. This ability is especially useful for fine-grained recognition tasks, where finding subtle differences is important. Throughout the paper, we use the CUB-2011 birds dataset [28] as a running example for fine-grained recognition scenarios. Trying to find suitable patch detectors for the aforementioned differences, within the whole training set, is a very complex task, which we significantly simplify by restricting the patch discovery to the K nearest neighbors leading to very compact feature representations.

⋆ A. Freytag and E. Rodner were supported by a FIT scholarship from the DAAD.

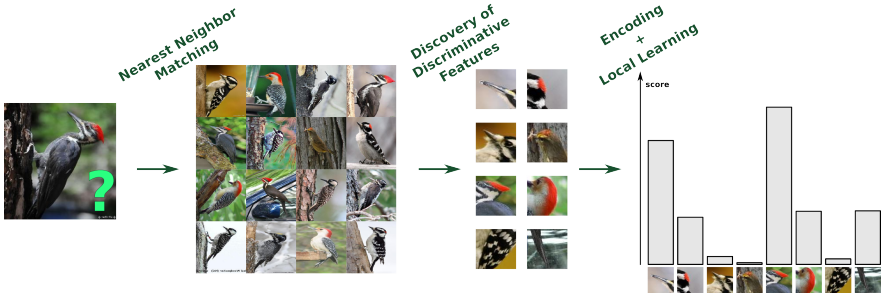


Fig. 1: Instead of training a global model, we seek for a given test image (*left*) its most similar images among all training samples (*middle left*). We then learn exemplar-specific representations by running our patch discovery on the retrieved image set and thereby figure out parts relevant for differentiation (*middle right*). Learned detectors are used to encode the images from the retrieved set and the query alike (*right*), to train a local model, and to finally classify the query image.

Fig. 1 gives an outline of this idea. In comparison to existing techniques, our approach offers the following main characteristics:

1. Convolution-based bootstrapping without restricting only to initial patches found by heuristic segmentation methods as in [18].
2. Exemplar-specific patch representations instead of global models.
3. A combination of exemplar-specific and semantic patches to improve on previous results on the CUB-2011 birds dataset.

First, we discuss related work in Sect. 2. Our automatic bootstrapping-based patch discovery is presented in Sect. 3. Steps towards exemplar-specific representations are given in Sect. 4 and the results of our experiments in the area of fine-grained recognition are evaluated in Sect. 5.

2 Related Work

There is a large body of literature on the topic of patch discovery, especially when also discriminative clustering and codebook learning methods are included (see [6] and references therein). In the following, we restrict ourselves to patch discovery methods related to our bootstrapping technique. Throughout the paper, we use the term *patch* and the notation $\boldsymbol{x}$ to refer to a small region, window, or block in an image, and the term *patch detector* and its notation $\boldsymbol{w}$ to refer to a template or linear classifier learned with a given set $\mathcal{M}(\boldsymbol{w})$ of patches.

Patch Discovery Patch discovery has been an important research field since the early works of [1] and [27]. Whereas [1] clusters similar patches of training images to obtain a vocabulary in an unsupervised manner, [27] finds class-specific patches by maximizing mutual information. Both papers (and related ones during the same time) use simple detectors based on gray values, which are hardly able to tackle the variability in natural images. The paper of [24] was the first one to present a patch discovery method

that made use of recent advances in object localization, such as HOG features and Exemplar-SVM models. The usefulness of patch discovery schemes for fine-grained recognition has been demonstrated recently in [23]. The work of [8] shows how to cast patch discovery as a non-convex optimization problem related to mode seeking. A common drawback is the time-consuming step of model learning with hard negative mining [24], and [18] presents a time-efficient version using whitened HOG features [15]. Their patch discovery scheme is based on bootstrapping a set of initial seed patches, which are previously derived from an unsupervised segmentation result. In contrast, we perform dense bootstrapping with convolutions, which allows for obtaining useful patch examples after a seeding step considering *every possible position* in the training images. Thus, our approach is more similar to latest techniques within the deep learning field[31,5] where discovery is done in a completely supervised manner and in several layers simultaneously.

Exemplar and Local Models State-of-the-art categorization techniques are usually based on huge representations and complex models. Local learning approaches aim at an orthogonal solution by learning models on the fly specifically tailored to every test image. Introduced in the early nineties [4], the main focus of those techniques is on a suitable trade-off between model capacity and number of training examples, especially for non-uniform distributions of samples in space. Some rare exceptions followed this idea through the years, *e.g.*, [30] for image categorization, or [16] for recognizing facial expressions. Similar in spirit are [14,13] showing how to transfer image annotations from nearest neighbors in the training set, which have been found by a global matching scheme. Whereas simple, those part detection-by-transferring methods are more flexible compared to global methods that can only tackle a limited number of viewpoints. However, current techniques assume a unique and constant feature space, *i.e.*, all approaches learn local models for fixed representations so far. Similar to our approach, [12] presented how to learn distance functions for every test sample to overcome this issue. In this paper, we even go one step further by learning image representations *and* classification models for every test sample on the fly, which allows focusing on patches important to differentiate quite similar birds already observed.

3 Discovering Mid-level Patch Representations

Our patch discovery scheme consists of three main parts: (1) finding initial seed patches, (2) learning patch detectors, and (3) convolution-based bootstrapping. The discovery is followed by a feature extraction step, where we generate features by spatially pooling patch detector outputs.

Finding Initial Seed Patches Finding initial seed patches is an important step to guide the following bootstrapping steps in the right direction. The purpose is similar to an interest point detection, which was often used in the earlier works on bag-of-features (see references cited in [19]). Here, we follow the idea of [18] and extract quadratic patches $\boldsymbol{x}^k$ of different sizes centered on the regions found by the region segmentation method of [11]. To further focus seed patches towards bird locations, we mask-out background regions using the pixelwise annotations provided with the dataset,

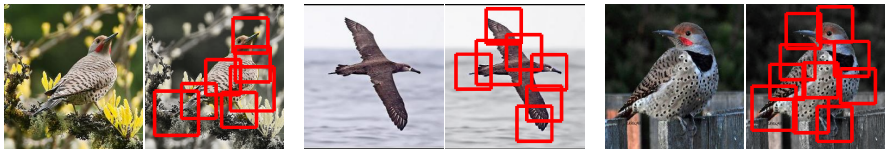


Fig. 2: Seeding results to initialize patches for discovery: seeding is based on unsupervised region segmentation [11] conducted with masked images.

but this is only an optional step. The first sets for the patch detectors $\boldsymbol{w}^k$ are then set to $\mathcal{M}(\boldsymbol{w}^k) = \{\boldsymbol{x}^k\}$. Fig. 2 shows some example regions extracted in this manner. Note that we are later on using convolutions to densely bootstrap these initial patches, therefore, the discovered patches are not restricted to initial seed patches as in [18].

Learning Patch Detectors Humans usually describe a birds appearance by a mixture of typical color and texture occurrences, *e.g.*, a dotted red belly or feathers with blue stripes. To meet this observation, we represent patches by histogram of oriented gradient features (HOG) and color feature histograms [25] computed for small cells of pixels. For a set of patches, we learn a linear patch detector and detection responses for unseen images are obtained by convolving the weight vector $\boldsymbol{w}$ of the learned model with computed feature planes of the image. As shown by [15], training HOG detectors can be done efficiently using standard Gaussian assumptions. Although only presented for HOG features, their technique can be applied to arbitrary features such as the combined HOG and color features we use in the experiments. Let us consider a single filter $\boldsymbol{w}$ that represents a classifier differentiating between positive (sub-images showing the patch) and negative examples. The paper of [15] assumes that positive and negative examples are Gaussian distributed with the same covariance matrix $\mathbf{S}_0$ and mean vector $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_0$, respectively. It can be shown that in this case, the optimal hyperplane separating positives and negatives can be calculated by $\boldsymbol{w} = \mathbf{S}_0^{-1}\left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right)$ [15]. Although the underlying assumptions leading to this equation might seem unrealistic, the resulting simple learning step is, at a closer look, a common feature whitening. It implicitly decorrelates all features using statistics of a large set of (negative) examples – an important step to deal with high correlations naturally arising, *e.g.*, between neighboring HOG cells [15].

Since every detector discriminates a tiny set of positive patches against everything else, the notion of 'everything else' can be shared by all detectors. Thus, the covariance matrix as well as the mean μ_0 of negative examples can be estimated from an arbitrary set of background images and features calculated therein. As a consequence, we can easily pre-compute it and re-use it for every patch model. In summary, the only remaining steps for learning a new patch detector from given examples is to average the features of positive patches and to solve a linear equation system.

Iterative Bootstrapping With Convolutions After learning a patch detector for each of the initial seed patches, we proceed with bootstrapping to obtain new useful training examples for each of the patch detectors. Bootstrapping can be performed in a supervised manner by restricting it to images of the category the initial seed patch was ex-

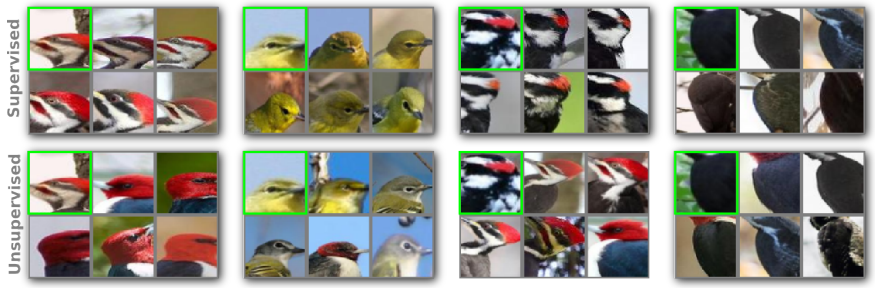


Fig. 3: Bootstrapping: initial seed patches are indicated with colored frames and the set of positive blocks for resulting detectors is displayed accordingly. While the technique is applicable to supervised and unsupervised scenarios, adding supervision prevents grouping of visually similar blocks from different categories.

tracted from [24,18,8], or unsupervised by using all training images. Both versions are evaluated in our experiments.

During bootstrapping, every patch detector is applied to every corresponding image, such that we obtain scores for every possible patch in all images. Note that this is different from the approach of [18], where only seed patches are considered as potential candidates for bootstrapping. Our method instead allows for discovering important patches not found by the unsupervised segmentation in the beginning.

Given the detection responses, we now seek for possible positive patches to increase the training set size of every detector leading to increased generalization abilities. In contrast to adding a fixed number m of the highest scored examples in each bootstrapping step, we add *at most* m examples. In particular, we do not add training examples that received a detection score worse than a certain percentage γ of the minimum score achieved by examples already used as positive training examples:

$$\mathcal{A}^{t+1}(\boldsymbol{w}^k) = \{\boldsymbol{x} \mid (\boldsymbol{w}^k)^T\boldsymbol{x} > \gamma \cdot \min_{\tilde{\boldsymbol{x}} \in \mathcal{M}^t(\boldsymbol{w}^k)} (\boldsymbol{w}^k)^T\tilde{\boldsymbol{x}}\}. \tag{1}$$

We denoted with $\mathcal{A}^{t+1}(\boldsymbol{w}^k)$ the set of accepted blocks for detector $\boldsymbol{w}^k$ in bootstrapping round $t+1$. If $\mathcal{A}^{t+1}$ is empty, we consider $\boldsymbol{w}^k$ as converged. Otherwise, we select m examples of $\mathcal{A}^{t+1}(\boldsymbol{w}^k)$ with highest score and add them to the set of positive samples:

$$\mathcal{M}^{t+1}(\boldsymbol{w}^k) = \text{top}_m\left(\mathcal{A}^{t+1}\left(\boldsymbol{w}^k\right)\right) \cup \mathcal{M}^t(\boldsymbol{w}^k) \ . \tag{2}$$

The parameter γ controls the exploration/exploitation trade-off during discovery and we set γ to 0.75 according to preliminary experiments on smaller datasets. Intuitively, larger values for γ prevent patch detectors from being "blurred" during bootstrapping by outliers. Furthermore, this strategy also leads to a convergence during bootstrapping without the necessity of specifying a fixed number of bootstrapping iterations as done in [18,24]. Visualizations for the process of bootstrapping are given in Fig. 3 for supervised and unsupervised scenarios. Initial seeding blocks for every detector are indicated with colored frames.

To finally remove non-discriminative patch detectors, previous approaches introduced supervised feature selection techniques, such as entropy-rank curves used by [18] or mutual information proposed by [24]. We skip this additional selection step and rely on the final classification model for the selection of relevant dimensions. Since bootstrapping is performed independently for each patch, early pruning would also not lead to an advantage in computation time, but would limit the number of features that can be used by a classifier later on drastically.

Extracting Features Based on Discovered Patch Detectors After discovering a set of patch detectors, it now remains to build a final feature representation for an unseen image. Every detector is trained on just a couple of training patches and fires only on an extremely small number of windows. Thus, the maximum score achieved for an image serves as an indicator whether or not the corresponding patch occurs. Consequently, max-pooling detection responses over the entire image leads to a feature vector with as many dimensions as detectors discovered previously.

Note that since computing detection results can be interpreted as a convolution of images and learned weight vectors of detector models, the overall pipeline shows an interesting parallel to deep learning techniques currently prominent. In direct comparison, we fix the lower layers and instead of learning planes associated with mid-level features by back-propagation, we instead bootstrap patch detectors, which could be also used in unsupervised settings. Given the great results recent approaches obtained by replacing handcrafted representation with rich representations learned in deep architectures [31,5], it would be interesting to see the proposed patch discovery scheme running in a local learning manner on those representations instead of HOG and color names only, *i.e.*, to fine-tune pre-trained deep architectures for *every single test image* and obtaining patches from an additional convolutional layer.

4 Exemplar-specific Mid-level Features

In order to reliably differentiate between subordinate classes, the identification of relevant features is among the most crucial aspects [7,20,10]. Although pose-alignment techniques [14,13] can almost eliminate effects based on significant pose variations, *e.g.*, of highly deformable objects like birds, they still have to struggle with the identification of discriminative features to encode the yet pose-aligned objects. Early papers in this area used off-the-shelf features such as bag-of-visual-word statistics [20] and more recent approaches further focused extraction on manually defined regions of interest for training samples [10,14]. Additionally, feature learning techniques have been proposed to distinguish object classes in an offline training step, by asking users [7], train 1-vs-1-features [3], or seek for a subset of useful random patches [9]. Still, all of these methods aim at calculating a unique general representation able to differentiate all training data as good as possible. Coupled with powerful post-processing techniques, *e.g.*, linear embeddings in high dimensional spaces [26], state-of-the-art systems usually work in feature spaces of hundreds of thousands of dimensions, while being trained on orders of magnitude less training samples (denoted with N).

We propose to follow an orthogonal path and find for every unseen image a compact, informative, and image specific feature representation. Thus, we start by querying for a

new test image its $K \ll N$ most similar training samples using Euclidean distance and combined HOG and color name (CN) features [25]. Fig. 1 displays this first step. All of these images are then used for the patch discovery described in the previous section. This results in a set of patch detectors that are specific for the current test image and especially for the global shape and pose of the object in it. The set of patch detectors is then used to compute feature representations for the test image as well as for the neighbors. Finally, a linear SVM classifier is trained on the neighbor images and used to predict the category for the test image. In terms of asymptotic runtimes, our local learning approach scales with $\mathcal{O}(K^2)$ during testing for patch discovery, and needs no training step. In contrast, discovering patch detectors for a global model demands at least $\mathcal{O}(N^2)$ and $\mathcal{O}(N)$ time during training and testing, respectively.

Combination of Discovered Patches and Semantic Parts As shown in [14], the performance of fine-grained recognition can be drastically improved when the location of semantic parts can be estimated, such as the head or back position for bird recognition. Therefore, we combine our approach with the exemplar-specific part prediction method proposed by [14]. The combination of both exemplar-specific classification approaches is done by late fusion. In particular, we are combining estimated class probabilities with linear combination $\mathcal{S}(\boldsymbol{x}) = \lambda \cdot \mathcal{S}_{\text{semantic}}(\boldsymbol{x}) + (1-\lambda)\mathcal{S}_{\text{discovery}}(\boldsymbol{x})$. We denoted with $\mathcal{S}$ class probabilities obtained via late-fusing probabilities $\mathcal{S}_{\text{semantic}}$ computed with a model using semantic part transfer [14] and probability scores $\mathcal{S}_{\text{discovery}}$ obtained from a model learned on discovered patches. The combination weight $\lambda \in [0, 1]$ serves as trade-off parameter and can be learned with leave-one-out estimation. It is important to note that in this paper, we optimize this parameter on the test set to simply show the potential of a combination.

5 Experiments

We evaluate our approach for fine-grained recognition on the CUB-2011 dataset [28] and use the provided split for training and testing. Following evaluation standards, we use the whole dataset CUB-2011-200 with all classes and the CUB-2011-14 dataset with only 14 classes as done in [10]. In the first part, we are interested in the accuracy using a global patch discovery with all its different flavors, whereas the exemplar-specific extension proposed in Sect. 4 is evaluated in the second part. Finally, we show how combining decisions of models learned on either semantic or discovered parts pays off and results in improved classification performance compared to state-of-the-art results on this dataset. For experimental details, we refer to the supplementary material and our source code, which is available at `http://www.inf-cv.uni-jena.de/fine_grained_recognition` .

5.1 Evaluation of Global Patch Discovery

We ran the patch discovery technique described in Sect. 3 with both supervised and unsupervised bootstrapping (with a maximum of 5 iterations). With the discovered patch

[1] Note that in [14], reported results are overall recognition rates averaged over all samples, whereas we report average recognition rates (averaged over class accuracies).

Table 1: Fine-grained recognition results on the CUB-2011 dataset.

Approach	CUB-2011-14	CUB-2011-200
Wah *et al.* [28]	–	10.25%
Our approach (global, unsupervised)	54.01%	-
Our approach (global, supervised)	58.01%	39.35%
Our approach ($K = 150$, supervised)	63.95%	34.16%
Style-awareness [23]	-	38.31%
PDL [17]	-	38.91%
Template learning [29]	-	43.67%
DPD [32]	-	50.98%
POOF [3]	70.10%	56.78%
Goering *et al.* [14][1]	73.39%	57.99%
Our approach ($K = 150$) + [14]	**76.64%**	**58.55%**
Our approach (global) + [14]	**78.25%**	**60.81%**

Table 2: Results on the CUB-2011-14 dataset **without** exemplar-specific discovery.

Approach	**CUB-2011-14**
Bootstrapping: on seeding blocks only [18]	46.93%
none	55.64%
Selection: entropy-rank criterion with merging [18]	26.96%
representative, without singletons [24]	58.62%
Our approach (globally discovered parts, convolution-based, no selection)	**58.01%**
Seeding: *human annotated semantic parts*	57.89%

detectors at hand, we encode training images as described previously. Classification accuracies are given in the first rows of Table 1.

First of all, we observe that the supervised bootstrapping clearly outperforms its unsupervised counterpart. Obviously, supervision during bootstrapping avoids rather similar patches with high discrimination abilities being grouped together as can be seen in Fig. 3. Although similar, tiny details still make some patches different from others, *e.g.*, the size of the red dot in the left group of patches. Based on these observations and the fact that unsupervised bootstrapping requires increased computation times compared to the supervised variant, we use supervised bootstrapping in all following evaluations only. Obtained accuracies are in the range of current results from patch discovery techniques such as [23], although not as competitive as latest techniques using ground truth part annotations and additional expert knowledge [3,14].

In Table 2, we analyzed different steps of our approach and compared to alternatives proposed in [18]. It can be seen that our convolution-based bootstrapping outperforms the bootstrapping by [18] which is conducted on seeding blocks only as well as a simple baseline that uses every seeding block as a patch detector without any bootstrapping involved. Furthermore, we can also see that the selection criteria proposed by [18,24] hurt the performance in our case. Interestingly, our part discovery scheme reaches a performance on par with a baseline that uses manually selected semantic parts. We further displayed detection response maps in Fig. 4 obtained by applying our discovered patch detectors on unseen test images.

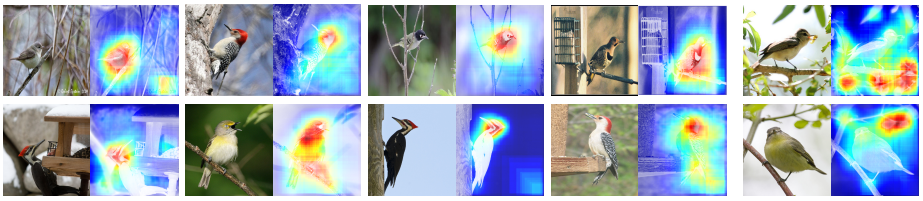


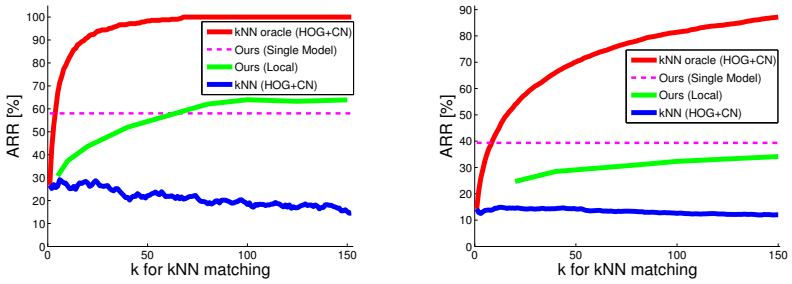
Fig. 4: Detection responses of discovered patch detectors on previously unseen test images. High scores are indicated by warm colors. The very right column displays cases where detectors are distracted by background patterns. Best viewed in color.

5.2 Evaluation of Exemplar-specific Representations

Choosing the Number of Neighbors Our fine-grained recognition approach using exemplar-specific patch representations and classifiers is limited by the neighbors found by global matching. When an example of the correct category is not present in the set of neighbors, we are logically not able to predict this category for the current test image. Therefore, we first analyze the quality of the matching scheme and results are given in Fig. 5. We plot the performance of an oracle method, which reflects a perfect classification when at least one example of the correct category is among the neighbors, and the performance of a plain majority vote classification. As can be seen, a small set of neighbors is sufficient for the CUB-2011-14 dataset to provide training examples of the correct class, which is not the case for the larger dataset CUB-2011-200. The majority vote classification is unlikely to already provide proper classification results due to the simple features used for global matching. However, it took us by surprise that this simple kNN-technique already improved over the first baseline ever given for this dataset [28] by more then 4 percent accuracy (first row in Table 1).

Evaluation on CUB-2011 Since the matching accuracy is sufficient to reduce the training images to a reasonable subset, we are now interested in the performance of the whole local learning pipeline. The results for the 14 and 200 class sets are given in Fig. 5a and Fig. 5b, respectively. Interestingly, the local learning approach with $k \geq 80$ neighbors outperforms the global learning approach on the small dataset. This is indeed remarkable, since only a fraction of dimensions is used (an overview of numbers of discovered patch detectors is given in the supplementary material). For the large dataset, however, matching seems to be the limiting factor, which can be already guessed from Fig. 5b. Consequently, non-euclidean distance matching [30] or techniques borrowed from image retrieval [2] would probably overcome current limitations here. Nonetheless, local learning results in quite impressive results given the fact that the dimensionality is about 2.0% compared to a global model. Thus, we conclude that the discovered patch detectors form a compact and informative representation.

Combining Patch Discovery and Part Transfer In a final experiment, we combined our approach with the semantic part transfer of [14] and the results are given in the lower part of Table 1. Although our combination technique is a simple weighted



(a) CUB-2011-14 (b) CUB-2011-200

Fig. 5: Accuracy of simple k-NN matching on CUB2011 for a perfect oracle, majority voting, and our approach for different sizes of the query set.

combination of class probabilities, it can be seen that we are able to obtain state-of-the-art performance among approaches with fixed feature representations for both the global and the exemplar-specific model, and thus can draw advantage from two complementary encryption techniques. Interestingly, our local variant is inferior to its global counterpart when being combined with the semantic part transfer. Our intuition is that this is mainly due to the way of transferring part annotations in [14] which also relies on a nearest neighbor search as done in local learning. Furthermore, it should be noted that the results of approaches that learn the underlying feature representations in a supervised manner with convolutional neural networks [31,5] recently obtained higher recognition rates on this dataset. Using these representations together with our discovery scheme is future work.

6 Conclusions

In this paper, we tackled the challenging problem of fine-grained recognition of bird species. Our approach consists of two key ingredients: a novel patch discovery technique and a new variant of local representation learning. For the introduced discovery scheme, we proposed an iterative bootstrapping technique to group re-occurring and informative subimages to patch detectors. In contrast to other papers in this area, our method performs dense bootstrapping without restricting itself to segmentation results and it is suitable both for unsupervised and supervised settings. To overcome computational burdens during learning and to further focus on relevant patches, our second contribution is a novel local representation learning formulation. Thereby, for every test image we learn classification models and image representations jointly by using a subset of training data most similar to the test image. Results on fine-grained recognition tasks have shown that the combination of discovered part and semantic part representations leads to a further boost in performance.

References

1. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: European Conference on Computer Vision (ECCV). pp. 113–127 (2002)
2. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2911–2918 (2012)
3. Berg, T., Belhumeur, P.N.: Poof: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 955 – 962 (2013)
4. Bottou, L., Vapnik, V.: Local learning algorithms. Neural computation 4(6), 888–900 (1992)
5. Branson, S., Van Horn, G., Belongie, S., Perona, P.: Improved bird species categorization using pose normalized deep convolutional nets. In: British Machine Vision Conference (BMVC) (2014)
6. Coates, A., Ng, A.Y.: The importance of encoding versus training with sparse coding and vector quantization. In: International Conference on Machine Learning (ICML). pp. 921–928 (2011)
7. Deng, J., Krause, J., Fei-Fei, L.: Fine-grained crowdsourcing for fine-grained recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 580–587 (2013)
8. Doersch, C., Gupta, A., Efros, A.A.: Mid-level visual element discovery by discriminative mean-shift. In: Neural Information Processing Systems (NIPS). pp. 1–8 (2013)
9. Duan, K., Parikh, D., Crandall, D., Grauman, K.: Discovering localized attributes for fine-grained recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3474–3481 (2012)
10. Farrell, R., Oza, O., Zhang, N., Morariu, V.I., Darrell, T., Davis, L.S.: Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In: International Conference on Computer Vision (ICCV). pp. 161–168 (2011)
11. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. International Journal of Computer Vision (IJCV) 59, 167–181 (2004)
12. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: International Conference on Computer Vision (ICCV). pp. 1–8 (2007)
13. Gavves, E., Fernando, B., Snoek, C., Smeulders, A., Tuytelaars, T.: Fine-grained categorization by alignments. In: International Conference on Computer Vision (ICCV). pp. 1–8 (2013)
14. Göring, C., Rodner, E., Freytag, A., Denzler, J.: Nonparametric part transfer for fine-grained recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2014)
15. Hariharan, B., Malik, J., Ramanan, D.: Discriminative decorrelation for clustering and classfication. In: European Conference on Computer Vision (ECCV) (2012)
16. Ionescu, R., Popescu, M., Grozea, C.: Local learning to improve bag of visual words model for facial expression recognition. In: International Conference on Machine Learning - Workshop on Representation Learning (ICML-WS) (2013)
17. Jia, Y., Vinyals, O., Darrell, T.: Pooling-invariant image feature learning. CoRR abs/1302.5056 (2013)
18. Juneja, M., Vedaldi, A., Jawahar, C., Zisserman, A.: Blocks that shout: Distinctive parts for scene classification. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 923–930 (2013)
19. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: International Conference on Computer Vision (ICCV). vol. 1, pp. 604–610 (2005)

20. Khan, F.S., Van De Weijer, J., Bagdanov, A.D., Vanrell, M.: Portmanteau vocabularies for multi-cue image representation. In: Neural Information Processing Systems (NIPS). pp. 1323–1331 (2011)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems (NIPS). vol. 1, p. 4 (2012)
22. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2169–2178 (2006)
23. Lee, Y.J., Efros, A.A., Hebert, M.: Style-aware mid-level representation for discovering visual connections in space and time. In: International Conference on Computer Vision (ICCV). pp. 1857–1864 (2013)
24. Singh, S., Gupta, A., Efros, A.: Unsupervised discovery of mid-level discriminative patches. In: European Conference on Computer Vision (ECCV). pp. 73–86 (2012)
25. Van De Weijer, J., Schmid, C.: Applying color names to image description. In: International Conference on Image Processing (ICIP). vol. 3, pp. III–493 (2007)
26. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 34(3), 480–492 (2012)
27. Vidal-Naquet, M., Ullman, S.: Object recognition with informative features and linear classification. In: International Conference on Computer Vision (ICCV). pp. 281–288 (2003)
28. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
29. Yang, S., Bo, L., Wang, J., Shapiro, L.: Unsupervised template learning for fine-grained object recognition. In: Neural Information Processing Systems (NIPS). pp. 3131–3139 (2012)
30. Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2126–2136 (2006)
31. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based r-cnns for fine-grained category detection. In: European Conference on Computer Vision (ECCV) (2014)
32. Zhang, N., Farrell, R., Iandola, F., Darrell, T.: Deformable part descriptors for fine-grained recognition and attribute prediction. In: International Conference on Computer Vision (ICCV) (2013)