Neural Activation Constellations: Unsupervised Part Model Discovery with Convolutional Networks

Marcel Simon and Erik Rodner
Computer Vision Group, University of Jena, Germany*
http://www.inf-cv.uni-jena.de/constellation_model_revisited

Abstract

Part models of object categories are essential for challenging recognition tasks, where differences in categories are subtle and only reflected in appearances of small parts of the object. We present an approach that is able to learn part models in a completely unsupervised manner, without part annotations and even without given bounding boxes during learning. The key idea is to find constellations of neural activation patterns computed using convolutional neural networks. In our experiments, we outperform existing approaches for fine-grained recognition on the CUB200-2011, NA birds, Oxford PETS, and Oxford Flowers dataset in case no part or bounding box annotations are available and achieve state-of-the-art performance for the Stanford Dog dataset. We also show the benefits of neural constellation models as a data augmentation technique for fine-tuning. Furthermore, our paper unites the areas of generic and fine-grained classification, since our approach is suitable for both scenarios.

1. Introduction

Object parts play a crucial role in many recent approaches for fine-grained recognition. They allow for capturing very localized discriminative features of an object [18]. Learning part models is often either done in a completely supervised manner by providing part annotations [7, 40] or labeled bounding boxes [15, 29].

In contrast, we show how to learn part-models in a completely unsupervised manner, which drastically reduces annotation costs for learning. Our approach is based on learning constellations of neural activation patterns obtained from pre-learned convolutional neural networks (CNN). Fig. 1 shows an overview of our approach. Our part hypotheses are outputs of an intermediate CNN layer for which we compute neural activation maps [29, 30]. Unsupervised part models are either build by randomly selecting a subset of the part hypotheses or learned by estimating the parameters of a generative spatial part model. In the latter case, we implicitly find subsets of part hypotheses that "fire" consistently in a certain constellation in the images.

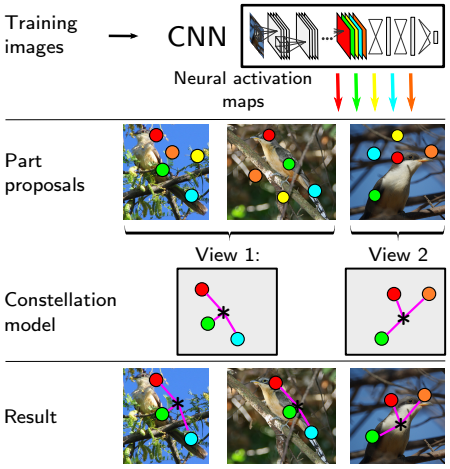

Figure 1. Overview of our approach. Deep neural activation maps are used to exploit the channels of a CNN as a part detector. We estimate a part model from completely unsupervised data by selecting part detectors that fire at similar relative locations. The created part models are then used to extract features at object parts for weakly-supervised classification.

Although creating a model for the spatial relationship of parts has already been introduced a decade ago [16, 14], these approaches face major difficulties due to the fact that part proposals are based on hand-engineered local descriptors and detectors without correspondence We overcome this problem by using implicit part detectors of a pre-learned CNN, which at the same time greatly simplifies the part-model training. As shown by [38], intermediate CNN outputs can often be linked to semantic parts of common objects and we are therefore using them as part proposals. Our part model learning has to select only a few parts for each view of an object from an already high quality pool of part proposals. This allows for a much simpler and faster part model creation without the need to explicitly consider appearance of the individual parts as done in pre-

*The authors thank NVIDIA for GPU hardware donations.

vious works [16, 1]. At the same time, we do not need any ground-truth part locations or bounding boxes.

The obtained approach and learning algorithm improves the state-of-the-art in fine-grained recognition on three datasets including CUB200-2011 [35] if no ground-truth part or bounding box annotations are available at all. In addition, we show how to use the same approach for generic object recognition on Caltech-256. This is a major difference to previous work on fine-grained recognition, since most approaches are not directly applicable to other tasks. For example, our approach is able to achieve state-of-the-art performance on Caltech-256 without the need for expensive dense evaluation on different scales of the image [31].

Furthermore, our work has impact beyond fine-grained recognition, since our method can also be used to guide data augmentation during fine-tuning for image classification. We demonstrate in our experiments that it even yields a more discriminative CNN compared to a CNN fine-tuned with ground-truth bounding boxes of the object.

In the next section, we give a brief overview over recent approaches in the areas of part constellation models and fine-grained classification. Sect. 3 reviews the approach of Simon *et al*. [29] for part proposal generation. In Sect. 4, we present our flexible unsupervised part discovery method. The remaining paper is dedicated to the experiments on several datasets (Sect. 5) and conclusions (Sect. 6).

2. Related work

Part constellation models Part constellation models describe the spatial relationship between object parts. There are many supervised methods for part model learning which rely on ground-truth part or bounding box annotations [41, 18, 29]. However, annotations are often not available or expensive to obtain. In contrast, the unsupervised setting does not require any annotation and relies on part proposals instead. It greatly differs from the supervised setting as the selection of useful parts is crucial. We focus on unsupervised approaches as these are the most related to our work.

One of the early works in this area is [42], where facial landmark detection was done by fusing single detections with a coupled ray model. Similar to our approach, a common reference point is used and the position of the other parts are described by a distribution of their relative polar coordinates. However, they rely on manually annotated parts while we focus on the unsupervised setting. Later on, Fergus *et al*. [16] and Fei-Fei *et al*. [14] build models based on generic SIFT interest point detections. The model includes the relative positions of the object parts as well as their relative scale and appearance. While their interest point detector delivers a number of detections without any semantics, each of the CNN-based part detectors we use correspond to a specific object part proposal already. This allows us to design the part selection much more efficient and to speed up the inference. The run time complexity compared to [16, 14] decreases from exponential in the number of modeled parts to linear time complexity. Similar computational limitations occur in other works as well, for example [27]. Especially in the case of a large number of part proposals this is a significant benefit.

Yang *et al*. [37] select object part templates from a set of randomly initialized image patches. They build a part model based on co-occurrence, diversity, and fitness of the templates in a set of training images. The detected object parts are used for part-based fine-grained classification of birds. In our application, co-occurrence and fitness are rather weak properties for the selection of CNN-based part proposals. For example, detectors of frequently occurring background patterns such as leaves of a tree would likely be selected by their algorithm. Instead our work considers the spatial relationship in order to filter unrelated background detectors that fire on inconsistent relative locations.

Crandall *et al*. [11] improve part model learning by jointly considering object and scene-related parts. However, the number of combinations of possible views of an object and different background patterns is huge. In contrast, our approach selects the part proposals based on the relative positions which is simpler and effective since we only want to identify useful part proposals for classification.

In the area of detection, there are numerous approaches based on object parts. The deformable part model (DPM, [15]) is the most popular one. It learns part constellation models relative to the bounding box with a latent discriminative SVM model. Most detection methods require at least ground-truth bounding box annotations. In contrast, our approach does not require such annotations or any negative examples, since we learn the constellation model in a generative manner and by using object part proposals not restricted to a bounding box.

Fine-grained recognition with part models Fine-grained recognition focuses on visually very similar classes, where the different object categories sometimes differ only in minor details. Examples are bird species [35] or car models [21] recognition. Since the differences of small parts of the objects matter, localized feature extraction using a part model plays an important role.

One of the earliest work in the area of fine-grained recognition uses an ellipsoid to model the bird pose [13] and fuse obtained parts using very specific kernel functions [40]. Other works build on deformable part models [15]. For example, the deformable part descriptor method of [41] uses a supervised version of [15] for training deformable part models, which then allows for pose normalization by comparing corresponding parts. The work of [17] and [18] demonstrated nonparametric part detection for fine-grained recognition. The basic idea is to transfer human-annotated part positions from similar training examples obtained with

nearest neighbor matching. Chai *et al*. [8] use the detections of DPM and the segmentation output of GrabCut to predict part locations. Branson *et al*. [7] use the part locations to warp image patches into a pose-normalized representation. Zhang *et al*. [39] select object part detections from object proposals generated by Selective Search [33]. The mentioned methods use the obtained part locations to calculate localized features. Berg *et al*. [4] learns a linear classifier for each pair of parts and classes. The decision values from numerous of such classifiers are used as feature representation. While all these approaches work well in many tasks, they require ground-truth part annotations at training and often also at test time. In contrast, our approach does not rely on expensive annotated part locations and is fully unsupervised for part model learning instead. This also follows the recent shift of interest towards less annotation during training [39, 36, 29]. The method of Simon *et al*. [29] presents a method, which requires bounding boxes of the object during training rather than part annotations. They also make use of neural activation maps for part discovery, but although our approach does not need bounding boxes we are still able to improve over their results.

The unsupervised scenario that we tackle has also been considered by Xiao *et al*. [36]. They cluster the channels of the last convolutional layers of a CNN into groups. Patches for the object and each part are extracted based on the activation of each of these groups. The patches are used to classify the image. While their work requires a pre-trained classifier for the objects of interest, we only need a CNN that can be pre-trained on a weakly related object dataset.

3. Deep neural activation maps

CNNs have demonstrated an amazing potential to learn a complete classification pipeline from scratch without the need to manually define low level features. Recent CNN architectures [22, 31] consist of multiple layers of convolutions, pooling operations, full linear transformations and non-linear activations.

The convolutional layers convolve the input with numerous kernels. As shown by [38], the kernels of the convolutions in early layers are similar to the filter masks used in many popular low level feature descriptors like HOG or SIFT. Their work also shows that the later layers are sensitive to increasingly abstract patterns in the image. These patterns can even correspond to whole objects [30] or parts of objects [29] and this is exactly what we exploit.

The output f of a layer before the fully-connected layers is organized in multiple channels $1 \leq p \leq P$ with a two-dimensional arrangement of output elements, *i.e*. we denote f by $(f^{(p)}_{j,j'}(\boldsymbol{I}))$ where $\boldsymbol{I} \in \mathbb{R}^{W \times H}$ denotes the input image and j and j' are indices of the output elements in the channel. Fig. 2 shows examples of such a channel output for the last convolutional layer. As can be seen the output can be interpreted as detection scores of multiple object part detectors. Therefore, the CNN automatically learned implicit part detectors relevant for the dataset it was trained from. In this case, the visualized channel shows high outputs at locations corresponding to the head of birds and dogs.

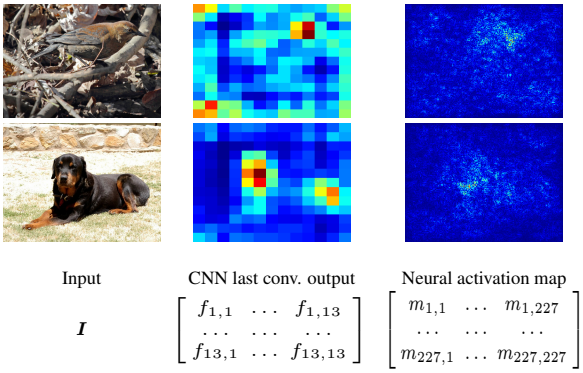

Figure 2. Examples for the output of a channel of the last convolutional layer and the corresponding neural activation maps for two images (index of the channel is skipped to ease notation). A deep red corresponds to high activation and a deep blue to no activation at all. Activation maps are available in higher resolution and better suited for part localization. Best viewed in color.

A disadvantage of the channel output is its resolution, which would not allow for precise localization of parts. Due to this reason, we follow the basic idea of [30] and [29] and compute *deep neural activation maps*. We calculate the gradient of the average output of the channel p with respect to the input image pixels $I_{x,y}$:

$$m^{(p)}_{x,y}(\boldsymbol{I}) = \frac{\partial}{\partial I_{x,y}} \sum_{j,j'} f^{(p)}_{j,j'}(\boldsymbol{I}) \tag{1}$$

The calculation can be easily achieved with a back-propagation pass [29]. The absolute value of the gradient shows which pixels in the image have the largest impact on the output of the channel. Similar to the actual output of the layer, it allows for localizing image areas this channel is sensitive to. However, the resolution of the deep neural activation maps is much higher (Fig. 2). In our experiments, we compute part proposal locations for a training image $\boldsymbol{I}_i$ from these maps by using the point of maximum activation:

$$\boldsymbol{\mu}_{i,p} = \operatorname*{argmax}_{x,y} \left| m^{(p)}_{x,y}(\boldsymbol{I}_i) \right| . \tag{2}$$

Each channel of the CNN delivers one neural activation map per image and we therefore obtain one part proposal per channel p. RGB images are handled by adding the absolute activation maps of each input channel. Hence we reduce a deep neural activation map to a 2D location and do not consider image patches for each part during the part model learning. In classification, however, image patches are extracted at predicted part locations for feature extraction.

The implicit part detectors are learned automatically during the training of the CNN. This is a huge benefit compared to other part discovery approaches like poselets [6], which do not necessarily produce parts useful for discrimination of classes a priori. In our case, the dataset used to train the CNN does not necessarily need to be the same as the final dataset and task for which we want to build part representations. In addition, determining the part proposals is nearly as fast as the classification with the CNN (only 110ms per image for 10 parts on a standard PC with GPU), which allows for real-time applications. A video visualizing a bird head detector based on this idea running at 10fps is available at our project website. We use the part proposals throughout the rest of this paper.

4. Unsupervised part model discovery

In this section, we show how to construct effective part models in an unsupervised manner given a set of training images of an object class. The resulting part model is used for localized feature extraction and subsequent fine-grained classification. In contrast to most previous work, we have a set of robust but not necessarily related part proposals and need to select useful ones for the current object class. Other approaches like DPM are faced with learning part detectors instead. The main consequence is that we do not need to care about expensive training of robust part detectors. Our task simplifies to a selection of useful detectors instead.

As input, we use the normalized part proposal locations $\boldsymbol{\mu}_{i,p} \in [0,1]^2$ for training image $i = 1,\dots,N$ and part proposal $p = 1,\dots,P$. The P part proposals correspond to the channels an intermediate output layer in a CNN and $\boldsymbol{\mu}_{i,p}$ is determined by calculating the activation map of channel p for input image i and locating the maximum response. If the activation map of a channel is equal to $\mathbf{0}$, the part proposal is considered hidden. This sparsity naturally occurs due to the rectified linear unit used as a nonlinear activation.

4.1. Random selection of parts

A simple method to build a part model with multiple parts is to select M random parts from all P proposals. For all training images, we then extract M feature vectors describing the image region around the part location. The features are stacked and a linear SVM is learned using image labels. This can even be combined with fine-tuning of the CNN used to extract the part features. Further details about part feature representations are given in Sect. 5.

In our experiments, we show that for generic object recognition random selection is indeed a valid technique. However, for fine-grained recognition, we need to select the parts that likely correspond to the same object and not a background artifact. Furthermore, using all proposals is not an option since the feature representation increases dramatically rendering training impractical. Therefore, we show in the following how to select only a few parts with a constellation model to boost classification performance and reduce computation time for feature calculation significantly.

4.2. Constellations of neural activations

The goal is to estimate a star shape model for a subset of selected proposals using the 2D locations of all part proposals of all training images. Similar to other popular part models like DPM [15], our model also incorporates multiple views $v = 1,\dots,V$ of the object of interest. For example, the front and the side view of a car is different and different parts are required to describe each view.

Each view consists of a selection of M part proposals denoted by the indicator variables $b_{v,p} \in \{0,1\}$ and we refer to them as parts. In addition, there is a set of corresponding shift vectors $\boldsymbol{d}_{v,p} \in [-1,1]^2$. The shift vectors are the ideal relative offset of part p to the common root location $\boldsymbol{a}_i$ of the object in image i. The $\boldsymbol{a}_i$ are latent variables since no object annotations are given during learning.

Another set of latent variables $s_{i,v} \in \{0,1\}$ denotes the view selection for each training image. We assume that there is only one target object visible in each image and hence only one view is selected for each image. Finally, $h_{i,p} \in \{0,1\}$ denotes if part p is visible in image i. In our case, the visibility of a part is provided by the part proposals and not estimated during learning.

Learning objective We identify the best model for the given training images by maximum a-posteriori estimation of all model and latent parameters $\boldsymbol{\Gamma} = (\boldsymbol{b}, \boldsymbol{d}, \boldsymbol{s}, \boldsymbol{a})$ from provided part proposal locations $\boldsymbol{\mu}$:

$$\hat{\boldsymbol{\Gamma}} = \operatorname{argmax}_{\boldsymbol{\Gamma}} \quad p\left(\boldsymbol{\Gamma} \mid \boldsymbol{\mu}\right). \tag{3}$$

In contrast to a marginalization of the latent variables, we obtain a very efficient learning algorithm. We apply Bayes' rule, use the typical assumption that training images and part proposals are independent given the model parameters [1], assume flat priors for $\boldsymbol{a}$ (no prior preference for the object's center) and $\boldsymbol{d}$ (no prior preference for part offsets), and independent priors for $\boldsymbol{b}$ and $\boldsymbol{s}$:

$$\begin{aligned} &\operatorname*{argmax}_{\boldsymbol{\Gamma}} \; p\left(\boldsymbol{\mu} \mid \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{s}, \boldsymbol{a}\right) \cdot p(\boldsymbol{b}) \cdot p(\boldsymbol{s}) \\ &= \operatorname*{argmax}_{\boldsymbol{\Gamma}} \prod_{i=1}^{N} \left(\prod_{p=1}^{P} p\left(\boldsymbol{\mu}_{i,p} \mid \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{s}, \boldsymbol{a}\right) \right) p(\boldsymbol{b}) \cdot p(\boldsymbol{s}) \end{aligned} \tag{4}$$

The term $p\left(\boldsymbol{\mu}_{i,p} \mid \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{s}, \boldsymbol{a}\right)$ is the distribution of the predicted part locations given the model. If the part p is used in view v of image i, we assume that the part location is normally distribution around the root location plus the shift vector, *i.e.* $\boldsymbol{\mu}_{i,p} \sim \mathcal{N}(\boldsymbol{d}_{v,p} + \boldsymbol{a}_i, \sigma^2_{v,p}\boldsymbol{E})$ with $\boldsymbol{E}$ denoting the identity matrix. If the part is not used, there is no prior information about the location and we assume it to be uniformly distributed over all possible image locations in $\boldsymbol{I}_i$.

Hence, the distribution is given by

$$p\left(\boldsymbol{\mu}_{i,p} \,|\, \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{s}, \boldsymbol{a}\right) = \prod_{v=1}^{V} \mathcal{N}\left(\boldsymbol{\mu}_{i,p} \,|\, \boldsymbol{a}_i + \boldsymbol{d}_{v,p}, \sigma_{v,p}^2 \boldsymbol{E}\right)^{t_{i,v,p}} \cdot \left(\frac{1}{|\boldsymbol{I}_i|}\right)^{1-t_{i,v,p}}, \quad (5)$$

where $t_{i,v,p} = s_{i,v} b_{v,p} h_{i,p} \in \{0,1\}$ indicates whether part p is used and visible in view v which is itself active in image i. The prior distribution for the part selection $\boldsymbol{b}$ only captures the constraint that M parts need to be selected, *i.e.* $\forall v : M = \sum_{p=1}^{P} b_{v,p}$. The prior for the view selection $\boldsymbol{s}$ incorporates our assumption that only a single view is active in training image i, *i.e.* $\forall i : 1 = \sum_{v=1}^{V} s_{i,v}$. In general, we denote the feasible set of variables as $\mathcal{M}$. Exploiting this and applying log simplifies Eq. (4) further:

$$\underset{\boldsymbol{\Gamma} \in \mathcal{M}}{\operatorname{argmin}} - \sum_{i=1}^{N} \sum_{p=1}^{P} \sum_{v=1}^{V} t_{i,v,p} \log \mathcal{N}\left(\boldsymbol{\mu}_{i,p} \,|\, \boldsymbol{a}_i + \boldsymbol{d}_{v,p}, \sigma_{v,p}^2\right)$$

In addition, we assume the variance $\sigma_{v,p}^2$ to be constant for all parts of all views. Hence, the final formulation of the optimization problem becomes

$$\underset{\boldsymbol{\Gamma} \in \mathcal{M}}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{p=1}^{P} \sum_{v=1}^{V} s_{i,v} b_{v,p} h_{i,p} \left\| \boldsymbol{\mu}_{i,p} - \boldsymbol{a}_i - \boldsymbol{d}_{v,p} \right\|^2 \quad (6)$$

Optimization Eq. (6) is solved by alternately optimizing each of the model variables $\boldsymbol{b}$ and $\boldsymbol{d}$, as well as the latent variables $\boldsymbol{a}$ and $\boldsymbol{s}$, independently, similar to the standard EM algorithm. For each of the variables $\boldsymbol{b}$ and $\boldsymbol{s}$, we can calculate the optimal value by sorting error terms. For example, $b_{v,p}$ is calculated by analyzing

$$\underset{\boldsymbol{b} \in \Gamma_{\boldsymbol{b}}}{\operatorname{argmin}} \sum_{p=1}^{P} \sum_{v=1}^{V} b_{v,p} \underbrace{\left(\sum_{i=1}^{N} s_{i,v} h_{i,p} \left\| \boldsymbol{\mu}_i^p - a_i - \boldsymbol{d}_{v,p} \right\|^2 \right)}_{E(v,p)} \quad (7)$$

This optimization can be intuitively solved. First, each view is considered independently, as we select a fixed number of parts for each view without considering the others. For each part proposal, we calculate $E\left(v,p\right)$. This term describes, how well the part proposal p fits to the view v. If its value is small, then the part proposal fits well to the view and should be selected. We now calculate $E\left(v,p\right)$ for all parts of view v and select the M parts with the smallest value. In a similar manner, the view selection $\boldsymbol{s}$ can be determined.

The root points $\boldsymbol{a}$ are obtained for fixed $\boldsymbol{b}$, $\boldsymbol{s}$, and $\boldsymbol{d}$ by

$$\hat{\boldsymbol{a}}_i = \sum_{v,p} t_{i,v,p} \left(\boldsymbol{\mu}_i^p - \boldsymbol{d}_{v,p}\right) / \left(\sum_{v',p'} t_{i,v',p'} \right). \quad (8)$$

Similarly, we obtain the shift vectors $\hat{\boldsymbol{d}}_{v,p}$:

$$\hat{\boldsymbol{d}}_{v,p} = \sum_{i=1}^{N} t_{i',v,p} \cdot \left(\boldsymbol{\mu}_{i,p} - \boldsymbol{a}_i\right) / \left(\sum_{i'=1}^{N} t_{i',v,p} \right). \quad (9)$$

The formulas are intuitive as, for example, the shift vectors $\boldsymbol{d}_{v,p}$ are assigned the mean offset between root point $\boldsymbol{a}_i$ and predicted part location $\boldsymbol{\mu}_{i,p}$. The mean, however, is only calculated for images in which part p is used.

This kind of optimization is comparable to the EM-algorithm and thus shares the same challenges. Especially the initialization of the variables is crucial. We initialize $\boldsymbol{a}$ to be the center of the image and $\boldsymbol{s}$ as well as $\boldsymbol{b}$ randomly to an assignment of views and selection of parts for each view, respectively. The initialization of $\boldsymbol{d}$ is avoided by calculating it first. The value of $\boldsymbol{b}$ is used to determine convergence. This optimization is repeated with different initializations and the result with the best objective value is used.

Inference The inference step for an unseen test image is similar to the calculations during training. The parameters $\boldsymbol{s}$ and $\boldsymbol{a}$ are iteratively estimated by solving Eq. (7) and (8) for fixed learned model parameters $\boldsymbol{b}$ and $\boldsymbol{d}$. The visibility is again provided directly by the neural activation maps.

5. Experiments

The experiments cover three main aspects and applications of our approach. First, we present a data augmentation technique based on the part models of our approach for fine-tuning, which outperforms fine-tuning on bounding boxes. Second, we apply our approach to fine-grained classification, a task in which most current approaches rely on ground-truth part annotations [7, 39, 29]. Finally, we show how to use the same approach for generic image classification, too, and present the benefits in this area. Code for our method will be made available.

5.1. Experimental setup

Datasets We use five different datasets in the experiments. For fine-grained classification, we evaluate our approach on CUB200-2011 [35] (200 classes, 11788 images), NA birds [34] (555 classes, 48562 images), Stanford dogs [20] (120 classes, 20580 images), Oxford flowers 102 [24] (102 classes, 8189 images), and Oxford-IIIT Pets [25] (37 classes, 7349 images). We use the provided split into training and test and follow the evaluation protocol of the corresponding papers. Hence we report the overall accuracy on CUB200-2001 and the mean class-wise accuracy on all other datasets. For the task of generic object recognition, we evaluate on Caltech 256 [19], which contains 30607 images of a diverse set of 256 common objects. We follow the evaluation protocol of [31] and randomly select 60 training images and use the rest for testing.

CNNs and parameters Two different CNN architectures were used in our experiments: the widely used architecture of Krizhevsky *et al.* [22] (AlexNet) and the more accurate one of Simonyan *et al.* [31] (VGG19). In case of NA birds,

we use GoogLeNet [32]. For details about the architecture, we kindly refer the reader to the corresponding papers. It is important to note that our approach can be used with any CNN. Features were calculated using the *relu6*, *relu7* and *pool5/7x7_s1* layer, respectively. For the localization of parts, the *pool5* layer was used. This layer consists of 256 and 512 channels resulting in 256 and 512 part proposals, respectively. In case of the CUB200-2011, NA birds, Oxford dogs, pets and flowers datasets, fine-tuning with our proposed data augmentation technique is used. We use two-step fine-tuning [7] starting with a learning rate of 0.001 and decrease it to 0.0001 when there is no change in the loss anymore. In case of Stanford dogs, the evaluation with CNNs pre-trained on ILSVRC 2012 images is biased as the complete dataset is a subset of the ILSVRC 2012 training image set. Hence, we remove the testing images of Stanford dogs from the training set of ILSVRC 2012 and learned a CNN from scratch on this modified dataset. The trained model is available on our website for easy comparison with this work.

If not mentioned otherwise, the learned part models use 5 views and 10 parts per view. A model is learned for each class separately. The part model learning is repeated 5 times and the model with the best objective value was taken. We count in how many images each part is used and select the 10 most often selected parts for use in classification.

Classification framework We use the part-based classification approach presented by Simon *et al*. [29]. Given the predicted localization of all selected parts, we crop square boxes centered at each part and calculate features for all of them. The size of these boxes is given by $\sqrt{\lambda \cdot W \cdot H}$, $\lambda \in \left\{\frac{1}{5}, \frac{1}{16}\right\}$, where W and H are the width and height of the uncropped image, respectively. If a part is not visible, the features calculated on a mean image are used instead. This kind of imputation has comparable performance to zero imputation, but yields in a slight performance gain in some cases. In case of CUB200-2011, we also estimate a bounding box for each image. Selective Search [33] is applied to each image to generate bounding box proposals. Each proposal is classified by the CNN and the proposal with the highest classification confidence is used as estimated bounding box.

The features of each part, the uncropped image and the estimated bounding box are stacked and classified using a linear SVM. In case of CUB200-2011, flipped training images were used as well. Hyperparameters were optimized using cross-validation on the training data of CUB200-2011 and used for the other datasets as well.

5.2. Data augmentation using part proposals

Fine-tuning is the adaption of a pre-learned CNN to a domain specific dataset. It significantly boosts the performance in many tasks [3]. Since the domain specific datasets are often small and thus the training of a CNN is prone to overfitting, the training set is artificially enlarged by using "data augmentation". A common technique used for example by [22, 31] is random cropping of a large fixed sized image patch. This is especially effective if the training images are cropped to the object of interest. If the images are not cropped and no ground-truth bounding box is available, uncropped images can be used instead. However, fine-tuning is less effective as shown in Tab. 1. Since ground-truth bounding box annotations are often not available or expensive to obtain, we propose to fine-tune on object proposals filtered by a novel selection scheme instead.

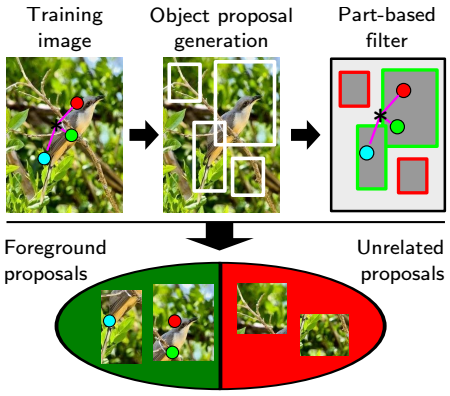

Figure 3. Overview of our approach to filter object proposals for fine-tuning of CNNs. Best viewed in color.

Train. Anno.	Method	Accuracy
Bbox	Fine-tuning on cropped images	67.24%
None	No fine-tuning	63.77%
None	Fine-tuning on uncropped images	66.10%
None	Fine-tuning on filtered part proposals	**67.97%**

Table 1. Influence of the augmentation technique used for fine-tuning in case of AlexNet on CUB200-2011. Classification accuracies were obtained by using 8 parts as described in Sect. 5.3.

An overview of our approach is shown in Fig. 3.

First, we select for each training image the five parts of the corresponding view, which fit the model best. Second, numerous object proposals are generated using Selective Search [33]. These proposals are very noisy, *i.e*. many only contain background and not the object of interest. We count how many of the predicted parts are inside of each proposal and select only proposals containing at least three parts. The remaining patches, ≈ 48 on average in case of CUB200-2011, are high quality image regions containing the object of interest. Finally, fine-tuning is performed using the filtered proposals of all training images.

The result of this approach is shown in Tab. 1. Fine-tuning on these patches provides not only a gain even compared to fine-tuning on cropped images, it also eliminates the need for ground-truth bonding box annotations.

Train. Anno.	Test Anno.	Method	Accuracy
Parts	Bbox	Bbox CNN features	56.00%
Parts	Bbox	Berg *et al*. [4]	56.78%
Parts	Bbox	Goering *et al*. [18]	57.84%
Parts	Bbox	Chai *et al*. [8]	59.40%
Parts	Bbox	Simon *et al*. [29]	62.53%
Parts	Bbox	Donahue *et al*. [12]	64.96%
Parts	None	Simon *et al*. [29]	60.55%
Parts	None	Zhang *et al*. [39]	73.50%
Parts	None	Branson *et al*. [7]	75.70%
Bbox	None	Simon *et al*. [29]	53.75%
None	None	Xaio *et al*. [36] (AlexNet)	69.70%
None	None	Xaio *et al*. [36] (VGG19)	77.90%
None	None	No parts (AlexNet)	52.20%
None	None	Ours, rand., Sect. 4.1 (AlexNet)	$60.30 \pm 0.74\%$
None	None	Ours, const., Sect. 4.2 (AlexNet)	68.50%
None	None	No parts (VGG19)	71.94%
None	None	Ours, rand., Sect. 4.1 (VGG19)	$79.44 \pm 0.56\%$
None	None	Ours, const., Sect. 4.2 (VGG19)	**81.01%**

Table 2. Species categorization performance on CUB200-2011.

Train. Anno.	Test Anno.	Method	Accuracy
Parts	Parts	Horn *et al*. [34]	75.0%
None	None	No parts (GoogLeNet)	63.9%
None	None	Ours, const., Sect. 4.2 (GoogLeNet)	**76.3%**

Table 3. Species categorization performance on NA Birds.

Method	Accuracy
Chai *et al*. [8]	45.60%
Gavves *et al*. [17]	50.10%
Chen *et al*. [10]	52.00%
Google LeNet ft [28]	75.00%
No parts (AlexNet)	55.90%
Ours, rand., Sect. 4.1 (AlexNet)	$63.29 \pm 0.97\%$
Ours, const., Sect. 4.2 (AlexNet)	68.61%

Table 4. Species categorization performance on Stanford dogs.

5.3. Fine-grained recognition without annotations

Most approaches in the area of fine-grained recognition rely on additional annotation like ground-truth part locations or bounding boxes. Recent works distinguish between several settings based on the amount of annotations required. The approaches either use part annotations, only bounding box annotations, or no annotation at all. In addition, the required annotation in training is distinguished from the annotation required at test time. Our approach only uses the class labels of the training images without additional annotation.

CUB200-2001 The results of fine-grained recognition on CUB200-2011 are shown in Tab. 2. We present three different results for every CNN architecture. "No parts" corresponds to global image features only. "Ours, rand." and "Ours, const." are the approaches presented in Sect. 4.1 and 4.2. As can be seen in the table, our approach improves the work of Xiao *et al*. [36] by 3.1%, an error decrease of more than 16%. It is important to note that their work requires a pre-trained classifier for birds in order to select useful patches for fine-tuning. In addition, the authors confirmed that they used a much larger bird subset of ImageNet for pre-training of their CNN. In contrast, our work is easier to adapt to other datasets as we only require a generic pre-trained CNN and no domain specific outside training data. The gap between our approach and the third best result in this setting by Simon *et al*. [29] is even higher with more than 27% difference. The table also shows results for the use of no parts and random part selection. As can be seen, even random part selection improves the accuracy by 8% on average compared to the use of no parts. The presented part selection scheme boosts the performance even further to 68.5% using AlexNet and 81.01% using VGG19.

NA birds The results of our approach on the relatively new NA birds dataset are shown in Tab. 3. The accuracy without using any parts is only 63.9%. Similar to the CUB200-2011 dataset, there is a clear advantage of using parts selected by our approach with an accuracy of 76.3%. Interestingly, the accuracy is very close to the one on CUB200, while there are more than 2.5 times more classes in NA birds. We outperform the baseline provided the authors using the approach of [7] even though we are not using any kind of part annotation.

Stanford dogs The accuracy on Stanford dogs is given in Tab. 4. To the best of our knowledge, there is only one work showing results for a CNN trained from scratch excluding the testing images of Stanford dogs. Sermanent *et al*. [28] fine-tuned the architecture of their very deep Google LeNet to obtain 75% accuracy. In our experiments, we used the much weaker architecture of Krizhevsky *et al*. and still reached 68.61%. Compared to the other non-deep architectures, this means an improvement of more than 16%.

Oxford pets and flowers The results for the Oxford flowers and pets dataset are shown in Tab. 5 and 6. Our approach consistently outperforms previous work by a large margin on both datasets. Similar to the other datasets, randomly selected parts already improve the accuracy by up to 4%. Our approach significantly improves this even further and achieves 95.35% and 91.60%, respectively.

Influence of the number of parts Fig. 7 provides insight into the influence of the number of parts used in classification. We compare to random part to the part constellation model based selection. In contrast to the previous experiments, one patch is extracted per part using $\lambda = \frac{1}{10}$. While random parts increase the accuracy for any amount of parts, the presented scheme clearly selects more relevant parts and

Method	Accuracy
Angelova *et al*. [2]	80.66%
Murray *et al*. [23]	84.60%
Razavian *et al*. [26]	86.80%
Azizpour *et al*. [3]	91.30%
No parts (AlexNet)	90.35%
Ours, rand., Sect. 4.1 (AlexNet)	$90.32 \pm 0.18\%$
Ours, const., Sect. 4.2 (AlexNet)	91.74%
No parts (VGG19)	93.07%
Ours, rand., Sect. 4.1 (VGG19)	$94.20 \pm 0.23\%$
Ours, const., Sect. 4.2 (VGG19)	**95.34%**

Table 5. Classification performance on Oxford 102 flowers.

Method	Accuracy
Bo *et al*. [5].	53.40%
Angelova *et al*. [2].	54.30%
Murray *et al*. [23].	56.80%
Azizpour *et al*. [3].	88.10%
No parts (AlexNet)	78.55%
Ours, rand., Sect. 4.1 (AlexNet)	$82.70 \pm 1.64\%$
Ours, const., Sect. 4.2 (AlexNet)	85.20%
No parts (VGG19)	88.76%
Ours, rand., Sect. 4.1 (VGG19)	$90.42 \pm 0.94\%$
Ours, const., Sect. 4.2 (VGG19)	**91.60%**

Table 6. Species categorization performance on Oxford-IIIT Pets.

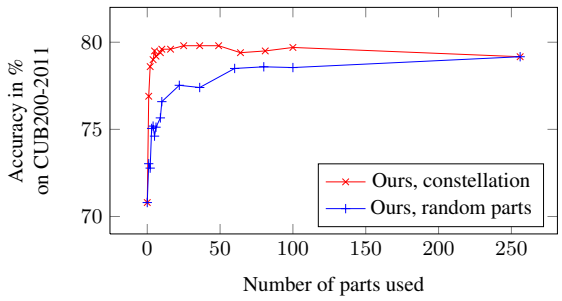

Table 7. Influence of the number of parts on the accuracy on CUB200-2011. One patch was extracted for each part proposal.

helps to greatly improve the accuracy.

5.4. From fine-grained to generic classification

Almost all current approaches in fine-grained recognition are specialized algorithms and it is hardly possible to apply them to generic classification tasks. The main reason is the common assumption in fine-grained recognition that there are shared semantic parts for all objects. Does that mean that all the rich knowledge in the area of fine-grained recognition will never be useful for other areas? Are fine-grained and generic classification so different? In our opinion, the answer is a clear no and the proposed approach is a good example for that.

Method	Accuracy
Zeiler *et al*. [38]	74.20%
Chatfield *et al*. [9]	78.82%
Simonyan *et al*. [31] + VGG19	85.10%
No parts (AlexNet)	71.44%
Ours, rand., Sect. 4.1 (AlexNet)	72.39%
Ours, const., Sect. 4.2 (AlexNet)	72.57%
No parts (VGG19)	82.44%
Ours, const., Sect. 4.2 (VGG19)	84.10%

Table 8. Accuracy on the Caltech 256 dataset with 60 training images per category.

There are two main challenges for applying fine-grained classification approaches to other tasks. First, the semantic part detectors need to be replaced by more abstract interest point detectors. Second, the selection or training of useful interest point detectors needs to consider that each object class has its own unique shape and set of semantic parts. Our approach can be applied to generic classification tasks in a natural way. The first challenge is already solved by using the part detectors of a CNN trained to distinguish a huge number of classes. Because of these properties, part proposals can be seen as generic interest point detectors with a focus on a special pattern. In contrast to semantic parts, they are not necessarily only recognizing a specific part of a specific object. Instead, they capture interesting points of many different kinds of objects. The second challenge is tackled by building class-wise part models and selecting part proposals that are shared among most classes. However, even a random selection of part detectors turns out to increase the classification accuracy already.

Caltech 256 The results of our approach on Caltech 256 are shown in Tab. 8. The proposed methods improves the baseline of global features without oversampling by 1% in case of AlexNet and 1.6% in case of VGG19. While Simonyan *et al*. achieves slightly higher performance, their approach is also much more expensive due to dense evaluation of the whole CNN over all possible crops at three different scales. Their best result of 86.2% is achieved by using a fusion of two CNN models, which is not done in our case and consequently not comparable. The results clearly shows that replacing semantic part detectors by more generic detectors can be enough to apply fine-grained classification approaches in other areas. Many current approaches in generic image classification rely on "blind" parts. For example, spatial pyramids or other oversampling methods are equivalent to part detectors that always detect something at a fixed position in the image. Replacing these "blind" detections by more sophisticated ones in combination with class-wise part models is a natural improvement.

6. Conclusions

This paper presents an unsupervised approach for the selection of generic parts for fine-grained and generic image classification. Given a CNN pre-trained for classification, we exploit the learned inherit part detectors for generic part detection. A part constellation model is estimated by analyzing the predicted part locations for all training images. The resulting model contains a selection of useful part proposals as well as their spatial relationship in different views of the object of interest.

We use this part model for part-based image classification in fine-grained and generic object recognition. In contrast to many recent fine-grained works, our approach surpasses the state-of-the-art in this area and is beneficial for other tasks like data augmentation and generic object classification as well. This is supported by, among other results, a recognition rate of 81.0% on CUB200-2011 without additional annotation and 84.1% accuracy on Caltech 256.

In our future work, we plan to use the deep neural activation maps directly as probability maps while maintaining the speed of our current approach. The estimation of object scale would allow for applying our approach to datasets in which objects only cover a small part of the image. Our current limitation is the assumption that a single channel corresponds to a object part. A combination of channels can be considered to improve localization accuracy. In addition, we plan to learn the constellation models and the subsequent classification jointly in a common framework.

7. Changelog

- V3: Added results for NA birds
- V2: Updated to camera ready version

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021, 2009. 1, 4

[2] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *CVPR*, 2013. 8

[3] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. *CoRR*, abs/1406.5774, 2014. 6, 8

[4] T. Berg and P. Belhumeur. POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013. 3, 7

[5] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *CVPR*, 2013. 8

[6] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 4

[7] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Improved bird species categorization using pose normalized deep convolutional nets. In *BMVC*, 2014. 1, 3, 5, 6, 7

[8] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013. 3, 7

[9] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 8

[10] G. Chen, J. Yang, H. Jin, E. Shechtman, J. Brandt, and T. Han. Selective pooling vector for fine-grained recognition. In *WACV*, 2015. 7

[11] D. J. Crandall and D. P. Huttenlocher. Composite models of objects and scenes for category recognition. In *CVPR*, 2007. 2

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 7

[13] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2009. 2

[14] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28(4), 2006. 1, 2

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9), 2010. 1, 2, 4

[16] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, 2003. 1, 2

[17] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013. 2, 7

[18] C. Göring, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *CVPR*, 2014. 1, 2, 7

[19] G. Griffin, A. Holub, and P. Perona. Website of the caltech 256 dataset. http://www.vision.caltech.edu/Image_Datasets/Caltech256/, 2007. 5

[20] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *ICCVW*, Colorado Springs, CO, 2011. 5

[21] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D object representations for fine-grained categorization. In *ICCVW*, 2013. 2

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3, 5, 6

[23] N. Murray and F. Perronnin. Generalized max pooling. In *CVPR*, 2014. 8

[24] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 5

[25] O. M. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Cats and dogs. In *CVPR*, 2012. 5

[26] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPRW*, 2014. 8

[27] E. Riabchenko, J.-K. Kamarainen, and K. Chen. Density-aware part-based object detection with positive examples. In *ICPR*, 2014. 2

[28] P. Sermanet, A. Frome, and E. Real. Attention for fine-grained categorization. *arXiv preprint arXiv:1412.7054*, 2014. 7

[29] M. Simon, E. Rodner, and J. Denzler. Part detector discovery in deep convolutional neural networks. In *ACCV*, 2014. 1, 2, 3, 5, 6, 7

[30] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*, 2014. 1, 3

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 3, 5, 6, 8

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 6

[33] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2), 2013. 3, 6

[34] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*, pages 595–604, 2015. 5, 7

[35] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 5

[36] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015. 3, 7

[37] S. Yang, L. Bo, J. Wang, and L. Shapiro. Unsupervised template learning for fine-grained object recognition. In *NIPS*, 2012. 2

[38] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 1, 3, 8

[39] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*. Springer, 2014. 3, 5, 7

[40] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *CVPR*, 2012. 1, 2

[41] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013. 2

[42] M. Zobel, A. Gebhard, D. Paulus, J. Denzler, and H. Niemann. Robust facial feature localization by coupled features. In *Automatic Face and Gesture Recognition*, pages 2–7, 2000. 2