The whole is more than its parts? From explicit to implicit pose normalization

Marcel Simon, Erik Rodner, Trevor Darrell, *Member, IEEE*, and Joachim Denzler, *Member, IEEE*

Abstract—Fine-grained classification describes the automated recognition of visually similar object categories like birds species. Previous works were usually based on explicit pose normalization, *i.e.,* the detection and description of object parts. However, recent models based on a final global average or bilinear pooling have achieved a comparable accuracy without this concept. In this paper, we analyze the advantages of these approaches over generic CNNs and explicit pose normalization approaches. We also show how they can achieve an implicit normalization of the object pose. A novel visualization technique called *activation flow* is introduced to investigate limitations in pose handling in traditional CNNs like AlexNet and VGG. Afterward, we present and compare the explicit pose normalization approach *neural activation constellations* and a generalized framework for the final global average and bilinear pooling called α-*pooling*. We observe that the latter often achieves a higher accuracy improving common CNN models by up to 22.9%, but lacks the interpretability of the explicit approaches. We present a visualization approach for understanding and analyzing predictions of the model to address this issue. Furthermore, we show that our approaches for fine-grained recognition are beneficial for other fields like action recognition.

Index Terms—Fine-grained classification, Object recognition, Convolutional Neural Networks

1 Introduction

The tremendous progress in image classification over the past years allowed for automatically recognizing more and more object categories. This lead towards fine-grained classification, which is the recognition of visually similar object categories like bird species [1]. Distinguishing fine-grained categories is challenging due to the small visual differences between categories, scarce training data, and large intra-class variance.

Previous approaches usually detect parts and represent an object by a concatenation of part features. However, Lin *et al.* [2] show comparable recognition rates using second-order descriptors [3] for encoding local features extracted with a generic convolutional neural networks (CNNs). Similar results were obtained with global average pooling [4]. In this paper, we analyze the differences between these approaches to common CNN models and previous fine-grained recognition concepts and show that they achieve an implicit normalization of object poses.

We first analyze common CNN architectures without object pose handling like AlexNet and VGG using a novel visualization technique called *activation flow*. It traces the highest class score back to the most influential intermediate patterns of the CNN. We are able to visualize the learned decomposition of objects into parts and to quantify the importance of the foreground object in classification. The results show that CNN models with several fully-connected layers for the encoding of local features tend to lose the object focus if the object pose is rare. Hence they are usually inferior compared to fine-grained approaches.

Afterward, we present *neural activation constellations* as a reference explicit pose normalization approach. It discovers object parts in an unsupervised way by generating proposals using a pre-trained CNN. A part constellation model is then learned to identify the proposals related to the foreground objects. We build the classification model by using these for a part-based description and classification.

We compare this approach to α-*pooling*, which is used as a final local feature encoding step similar to global average [5] or bilinear pooling [2]. The global aggregation of local features followed by a linear classifier leads to a pairwise matching of local features in the similarity function of the classifier [6]. Hence we refer to these approaches as *implicit pose normalization* approaches. We observe that tasks like fine-grained recognition require only very few such matches for correct identification. In contrast, other tasks such as scenes recognition might require much more matches as the overall scene matters instead of single objects. We hence propose to learn the influence of the largest matchings by learning a parameter α. Depending on its value, the pooling strategy changes and both global average and single-stream bilinear pooling can be achieved as special cases.

The approaches increase the accuracy of popular CNN architectures like AlexNet [7], VGG-VD [8], and ResNet [5] on three fine-grained recognition tasks significantly by up to 22.9%. Our comparison reveals that the implicit pose normalization approach achieves slightly higher recognition rates. However, the interpretability is missing as it is not clear, which object parts contributed to a decision. We address this with a classification visualization showing the relationship of test and training image regions.

Our comparison also shows how to express the explicit pose normalization as implicit pose normalization approach

- *M. Simon and J. Denzler are with the Computer Vision Group, Friedrich-Schiller-Universität Jena, Germany. E-mail: marcel.simon, joachim.denzler@uni-jena.de*
- *E. Rodner is with Carl-Zeiss Corporate Research, Jena*
- *T. Darrell is with the Department of Electrical Engineering and Computer Science (CS Division), UC Berkeley. E-mail: trevor@cs.berkeley.edu.*

Manuscript received XXX

and vice versa, which reveals additional differences. For example, only the explicit pose normalization approach can exploit a valuable constellation model, while only α-pooling exploits the full detection map instead of the peak location only. Finally, we compare the transferability of the approaches to another domain. In action recognition on the Stanford 40 actions dataset [9], especially α-pooling showed an advantage reaching up to 87.7%.

This paper combines our previous publications [10], [11], [12] and adds a wide range of additional aspects and analysis. In particular, our *first contribution* is a novel visualization scheme called activation flow for analyzing learned object decompositions of CNNs. It is a valuable basis for qualitative and quantitative analysis of trained models.

Our *second contribution* is an evaluation of the influence of rare object poses on the recognition process. We show that rare poses cause CNNs to partially lose the focus on foreground objects in bird classification. This insight motivates that handling rare object poses is a key component for improving image classification models.

The *third contribution* is the presentation of our previous work in the context of explicit and implicit pose normalization. We compare both concepts and show advantages of each, which will allow for improving the approaches by transferring ideas in the future.

Our *fourth contribution* is an extensive evaluation using new datasets, an additional CNN model ResNet-50, and an updated normalization scheme using the matrix root [13] for α-pooling. The ResNet model uses global average pooling and hence already implicitly normalizes object poses as explained in Section 5. We show that our approaches nevertheless improve its accuracy.

Following this introduction, Section 2 reviews relevant previous work followed by the analysis of generic CNN models using activation flow in Section 3. Our approaches for explicit and implicit pose normalization are presented and compared in Sections 4 and 5. The results of the evaluation and an ablation study are shown in Section 6.

2 Related work

Our work relates to publications on image classification, fine-grained classification, part discovery and selection, and visualization of learned representations. In the following, we list and compare a selection of relevant works.

Image classification. Fine-grained classification is a special case of image classification. Prior to 2012, most computer vision approaches were based on local feature descriptors like scale-invariant feature transform (SIFT) [14] or histogram of oriented gradients (HoG) [15] combined with encodings like bag-of-words [16], fisher vectors [17], [18], or VLAD [19]. In addition, spatial relationships were modeled, for example, by spatial pyramid matching [20]. Due to their success in recent years, CNNs trained for image classification tasks like the ILSVRC classification dataset [21] are now widely used. Examples are AlexNet [7], VGG-VD [8], Inception [22], and ResNet [5]. Based on these, several extensions and improvements have been presented, including pre-activation ResNets [23], wide ResNets [24], highway networks [25], leaky rectified linear units [26], and spatial pyramid pooling [27]. We use generic classification models as a basis and show how to improve accuracy for the task of fine-grained classification.

Fine-grained classification. Fine-grained recognition received a notable amount of attention in recent years. Starting with part-based models like [28], [29], it developed to a separate field with its own, usually part-based, approaches [30], [31], [32] and datasets like Oxford flowers [33], CUB200-2011 [1], Stanford dogs [34], and the Oxford IIIT pets dataset [35].

Early works explore a large variety of ideas, which mostly belong to the area of explicit pose normalization approaches. For example, Zhang *et al.* [36] present pose pooling kernels based on poselet detections of two images and Yao *et al.* [37] use classifiers trained on responses of generated class templates. Branson *et al.* [30] study interactive classification based on part-related attributes. Soon most publications exploited part location annotations to further improve the accuracy. For example, Liu *et al.* [32] train detectors for parts of a dog face using ground-truth annotations and compute localized descriptions. Göring *et al.* [38] transfer part location annotations from training images with similar object pose. Branson *et al.* [39] gain robustness against pose variation by warping patches of object parts to a canonical pose. Zhang *et al.* [40] learn a hierarchical detection model, which incorporates deformations of the parts, in order to increase the accuracy of detection and hence also of classification.

As annotating object parts is tedious, later works propose approaches for the case that only the bounding boxes are annotated [41], [42]. Donahue *et al.* [43] and Razavian *et al.* [44] evaluate the classification of images cropped to the bounding box with CNNs. Gavves *et al.* [42] and Krause *et al.* [41] use co-segmentation or alignment to identify corresponding image regions. There are also more generic approaches, which turned out to work well on fine-grained tasks. For example, Yao *et al.* [31] learn a decision tree, which uses image regions at each node. Chai *et al.* [45] aggregate feature within foreground masks generated with co-segmentation. In contrast to all these approaches, we do not rely on any ground-truth location annotation, neither part locations nor bounding boxes.

Starting with publications like [46] and our work [11] presented in Section 4, there is a noticeable increase in the number of publications, which do not use any extra annotations beside the class labels. Xiao *et al.* [46] cluster channels of a convolutional layer in a CNN and interpret each cluster as an object part and use it for part detection. Zhang *et al.* [47] mine part detections in a similar manner from convolutional layers as well and try to learn traditional detectors from these afterward. Wang *et al.* [48] exploit the hierarchical structure of classes, which is often present in fine-grained tasks. Jaderberg *et al.* [49] estimate transformation parameters for the input image to focus the classification on relevant image regions. Liu *et al.* [50] combine the information from two layers, one for description and one for localization. Compared to these approaches, our neural activation constellations approach is able to select only the foreground related channels of a CNN. It hence allows for focusing the classification on the foreground object, which is desirable in our task. In addition, it is easy to apply to any given pre-trained CNN model without laborious adjusting

of learning hyperparameters. As shown in the experiments, it improves the accuracy of very different CNN architectures without changing hyperparameters.

Implicit pose normalization has only recently achieved the level of accuracy of explicit pose normalization. In particular, Lin *et al.* [2] show the effectiveness of the second-order pooling strategy [3] for fine-grained tasks. They proposed a two-stream approach which combines two possibly distinct local features to a single combined representation using the outer product. We will refer to the case of two identical streams as it achieves comparable accuracy. Krause *et al.* [51] show that also global average pooling of the Inception architecture [52] can achieve competitive performance. We introduce the term implicit pose normalization for these approaches, as pose normalization is obtained by a pairwise matching of all local features in the similarity function of the classifier. Our approach generalizes the global average and second-order pooling used in these approaches and automatically learns the pooling strategy from data.

Part constellation models. Spatial relationships between object parts are described by constellation models. Early works include, for example, Zobel *et al.* [28], which fuse single part detection of face landmarks with a coupled ray model. In this work, however, we focus on approaches which generate parts and their relationship in an unsupervised manner as part annotations are expensive to obtain.

Fergus *et al.* [29] and Fei-Fei *et al.* [53] model the spatial relationship between generic SIFT interest point detections, while Riabchenko *et al.* [54] use Gabor filters. The generic interest point detector used in these works does not provide any connection between the detections in two images. In contrast, our generated part detector proposals provide related detections across images. This also allows us to decrease the exponential runtime complexity in the number of modeled parts present in these approaches to linear complexity in our approach.

Yang *et al.* [55] select part proposals using co-occurrence, diversity, and fitness as criteria. Crandall *et al.* [56] incorporate co-occurring background patterns into the part selection. Both approaches lead to a selection of background parts, which is usually not desirable in fine-grained classification. In contrast, we are able to identify more foreground parts due to the modeled spatial relationship.

Discovered object parts are also used in detection frameworks like the deformable parts model [57]. The filter masks of the parts are heuristically initialized and optimized for a maximal detection score. Similar to the works based on generic interest point detectors [29], [53], the correspondence between parts of different views is missing. Zhang *et al.* [58] address this by learning a separate classifier for each object view. In contrast, our approach leverages the complete training data as our model can share parts across multiple views of an object.

Visualization of learned representations. CNNs are often seen as a "black box" and hence several ideas have been presented to understand learned models. The approaches can be roughly categorized into visualizing learned concepts and visualizing a prediction. Learned concepts can be analyzed by visualizing parameter values, *e.g.*, the filter masks of the first layer of a neural network [7], [59]. Other works maximize the output of a selected (intermediate) channel starting from a real [60], [61] or random image [61], [62], [63], [64] or select image patches achieving the highest output from a dataset [59]. Predictions can be visualized using relevance or attention maps [65], [66], [67]. It is also possible to obtain the most relevant training images using the representer theorem for classifiers [68], [69].

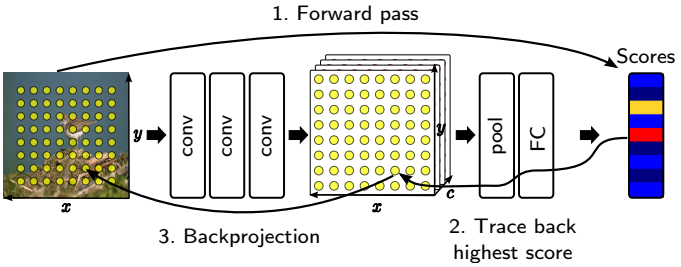

Fig. 1. Outline of the activation flow calculation. We first perform a regular forward pass through the network and determine the highest scoring class. Afterward, we trace back the most contributing elements from previous layers. For each identified element, we project its location on the feature map back to the input image for visualization purposes.

The activation flow presented in this work visualizes predictions. Previous papers focus on analyzing the relevant input regions or analyzing specific output elements. However, they do not investigate the relationship between the intermediate representations learned by a model. In contrast, we show the learned decomposition of a given object into corresponding mid-level representations. Furthermore, while previous works stop at mid-level object representations, our work can visualize the full hierarchy from full object to low-level edges.

3 Activation Flow for Analyzing Learned CNN Models

In this section, we analyze generic CNN architectures and their ability for fine-grained recognition. We use a novel approach called *activation flow* to visualize the hierarchy of intermediate patterns relevant for a prediction. We show that rare object poses are still challenging for CNNs containing several fully-connected layers.

Approach. Our approach is based on tracing the highest class score back to those intermediate elements of the CNN, which has the largest influence on this output. Fig. 1 shows a simplified outline of the generation process. The visualization is computed for a single image. We first perform a forward pass to compute the highest scoring class. We currently do not consider a possibly similar second highest class score for clarity in the figure. Afterward, we recursively go back layer by layer. At each layer, we identify the input element with the largest positive contribution to the output element selected in the layer before.

For example, suppose class 131 has the highest score for an input image in a VGG-VD model. We start at layer `fc8` and identify the input element of this layer, which had the largest positive contribution to the score of class 131. Suppose this is element 3821. The input element 3821 of `fc8` is the output of the previous layer `fc7`. Hence we now continue to the layer `fc7` and identify its input element with the largest positive contribution to the output element 3821. The recursion can be continued until the input image.

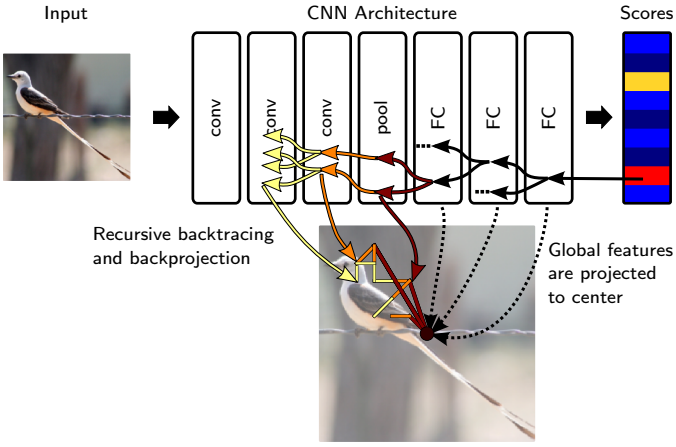

Fig. 2. Details about the visualization. We trace back the two highest contribution elements for each element selected before and project all elements as a tree to the input image. Earlier layers are marked with brighter colors.

Fig. 3. Activation flow for VGG-VG on CUB200-2011 birds. The top row depicts common poses, which were correctly classified by the network, and the bottom row failure cases with rare poses.

Each selected element in this process can be connected to a position in the input image. This is possible because the intermediate output of a CNN can be interpreted as a two-dimensional multi-channel feature map as shown in the center of Fig. 1. Each point on the feature map can be projected to a point in the input image. For an accurate projection, it is important to consider the padding, filter sizes, and strides used in convolutions and pooling operations.

Fig. 2 shows more details of the visualization. Except for the class scores, we always identify the l highest elements and recursively trace back each of them individually. This results in a tree of elements, where each element is associated with a certain layer and position in the input image. In the figure, we visualize the tree using colors ranging from black for the last layers to red and yellow for earlier layers.

Fully connected layers are a special case. They compute only a single global feature vector for the whole image and hence all spatial information is lost. We project elements of these layers to the image center for visualization purposes. However, there is no clear corresponding location for these elements. In the supplementary material, we also present results when projecting only the convolutional layers.

Fig. 3 shows the activation flow for images of the CUB200-2011 birds dataset and a fine-tuned VGG-VD model. In the top row, we show the activation flow for images showing common bird poses. The flow focuses on the object itself and even covers semantic parts like the birds head, belly, and tail. The bottom row shows the flow for images with rarer bird poses. The network's focus shifts away from the object towards background patterns.

Our visualization can be used to analyze limitations of CNN architectures. For example, the recognition of the top row images is supported by a wide range of body parts. Hence, their prediction seems more trustworthy compared to the images in the bottom row. For example, in the bottom right image, the prediction seems to be based mostly on the background. This suggests that the part detection failed and an approach for accurate part localization might help to improve the recognition process.

Quantifying object focus. We are interested in whether rare object poses cause CNNs to lose the focus on the foreground object. The presented activation flow is a basis for this. We compute two measures: a quantification of how well a model is able to recognize the full object pose and a measure of pose rarity.

The quantification uses the presented activation flow, where we warp the input image to fit the input shape of the CNN. To keep computation time reasonable, we compute for each selected output element the $l = 2$ highest contributing input elements and limit the recursion to seven convolution and pooling operations. Given the projected locations of all selected elements of the activation flow, we compute how many of these elements are located within the object. The ground-truth object segmentation is used for this purpose. The percentage of elements inside the object is used as the measure in the plot and will be called *coverage* in the following. In the supplementary material, we explore other measures such as the mean distance of elements to the foreground object. Please note that we ignore fully-connected layers here as their elements do not have a clear correspondence to a position in the input image.

The rarity of a pose is computed from ground-truth part annotations. The locations of all parts in an image are concatenated. The resulting vectors of all training images are clustered using k-means. The locations are normalized with respect to the bounding box and hence robust against translations and object scale. The cluster centers computed with k-means represent common object poses. We then define the rarity of the object pose in a new image by the minimal distance of its part locations to the cluster centers. The distance is computed as L^2-distance of all visible parts in the image. We compute the distance to both the original pose and its flipped variant and use the smaller value. The L^2-distance does not fully represent our understanding of pose differences. However, missing 3D coordinates and models prevent applying more sophisticated distance metrics.

The result for AlexNet and the CUB200-2011 birds dataset is shown in Fig. 4. We can observe that there is an inverse relationship between foreground object focus and pose rarity. This means the rarer the pose, the lower is the focus of the activation flow on the foreground object. The

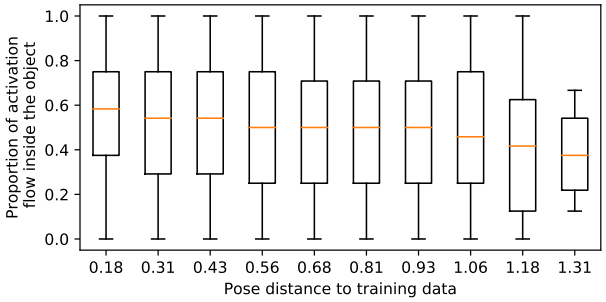

Fig. 4. The percentage of elements of the activation flow located on the object (coverage) versus pose rarity.

TABLE 1
Mean coverage and classification accuracy of different network architectures using CUB200-2011.

Architecture	Accuracy	Coverage
AlexNet	52.20%	47.02%
VGG-VD	71.94%	58.96%
ResNet-50	80.35%	65.20%

same conclusion can also be derived from Fig. 3, where we showed examples of such rare bird poses. However, the variance in the plot is fairly high suggesting that other factors influence the coverage as well. We hence also compute the classification accuracy for the 100 most common and most rare poses for further evidence. While rare poses are only recognized in 42% of the cases, the recognition rate on common poses is 57%.

In Table 1, we compare different network architecture using the mean coverage and classification accuracy. The mean coverage increases with more complex and more accurate models suggesting that a higher classification accuracy is correlated with a larger foreground object focus.

Observations. Figs. 3 and 4 and Table 1 let us draw three important conclusions. First, the widely used AlexNet and VGG-VD models base their decisions on the appearance of a wide range of body parts. However, the bird's pose influences the focus during classification. Rare poses, shown in Fig. 4 on the right, seem to cause a loss in foreground object focus. We explain this with the lack of training data. For common bird poses, many training images exist and hence the fully-connected layers learned these poses. Uncommon bird poses, in contrast, might never occur in the training data at all. Hence the CNN is not able to recognize the bird using the appearance of the complete animal. It focuses only on often seen less deformable subparts like the head.

Second, the comparison between network architectures shows that there is a correlation between model accuracy and focus on the foreground object. Hence object focus might be beneficial for improving classification accuracy and increasing the object focus further is a promising idea to further increase accuracy. It is in particular important to handle uncommon bird poses. In this work, we use explicit and implicit pose normalization to achieve this goal. The ResNet-50 model shown in the table already achieves a certain robustness due to the global average pooling, which

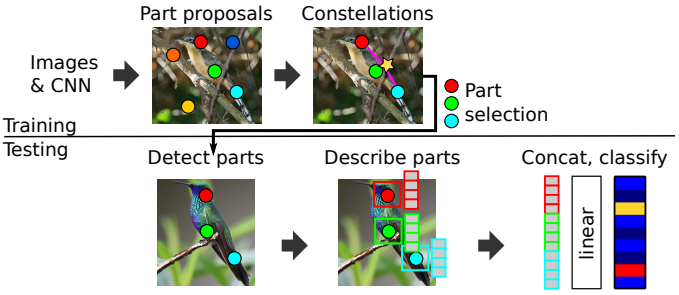

Fig. 5. The framework of our explicit pose normalization approach.

can be seen by the larger coverage value.

Third, the most important layer for handling pose variations seems to be last layer, which aggregates the local features to a single global representation. This layer corresponds to the long dark red lines in the activation flow, which connect the location of `pool5` elements and its consecutive layer `fc6` in our case. It aggregates local part-level descriptors to a single global object-level representation. In our part discovery, we are interested in abstract object parts just below the object level. We hence use the input of this last aggregation layer for part discovery in our approach.

4 Explicit Pose Normalization with Neural Activation Constellations

Section 3 showed the limitations of AlexNet and VGG-VD when handling objects with large pose variation. In this section, we show how to address this issue with explicit pose normalization. It can be seen as a replacement for the ensembles of poses learned by the fully-connected layers.

4.1 Part-based representation

The framework of our explicit pose normalization approach is shown in Fig. 5. The classification pipeline is based on localizing discovered parts, extracting local features, and predicting the class scores using a combination of all part features. The object parts are discovered by generating part detector proposals from intermediate representations of a pre-trained CNN. Afterward, an unsupervised part constellation model is learned to select the most relevant proposals for the foreground objects. This approach is called *neural activation constellations* and presented in the following Sections 4.2 and 4.3. For feature extraction, we use the intermediate activations of a CNN. The representations of all parts and the global image are concatenated and passed on to a linear classifier.

4.2 Part detector discovery

Object parts can be generated in an unsupervised way using a pre-trained CNN. As shown in previous work [59], channels of a CNN are related to certain patterns in the input image. We exploit this knowledge by using the channels of a convolutional layer as part detectors. Given a selected channel and an input image, we compute a *neural activation map* as the flattened gradient of the summed channel output with respect to the input image. The result is a detection heat map, in which large absolute values correspond to high

detection scores. The part location for the image is then computed as the location of highest response in the map. We provide more details on the computation as well as a visualization of neural activation maps in the supplementary material.

4.3 Unsupervised selection of detector proposals

We obtain one part detector proposal for each channel in the last convolutional layer of the CNN. This set of all proposals will be called *parts*. We now describe an unsupervised approach for selecting the most relevant parts, called *neural activation constellations*, which does not require any location annotations. We assume that relevant parts appear in a consistent relative location to each other. This exists for semantic parts, but not for unrelated background parts.

Constellation model. The relationship between parts is modeled with a *part constellation model*. We estimate a multi-view star shape part model $\mathbf{\Gamma}$, which includes a part selection and the spatial relationship between these parts.

Our model is defined by $\mathbf{\Gamma} = (\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{a})$, consisting of the view selection $\boldsymbol{s}$, part selection $\boldsymbol{b}$, shift vectors $\boldsymbol{d}$, and anchor points $\boldsymbol{a}$. Similar to other popular part models like DPM [57], our model also incorporates multiple views of the modeled objects. For example, the front and the side view of a car are different and different parts are required to describe each view. The $\boldsymbol{s}_n^v \in \{0,1\}$, $1 \leq v \leq V$, are latent variables indicating the view selection for training image n. We assume that there is only one target object visible in each image and hence only one view is selected for each image, *i.e.*, $\sum_{v=1}^{V} \boldsymbol{s}_n^v = 1$. Each view consists of a selection of part proposals denoted by P indicator variables $\boldsymbol{b}_v^p \in \{0,1\}$, $1 \leq p \leq P$, one for each of the parts. The anchor points $\boldsymbol{a}_n \in [0,1]^2$ capture the location of the object center in image n and are latent as no object annotations are given during learning. The shift vectors $\boldsymbol{d}_v^p \in [-1,1]^2$ model the relative offset of part p to the common root location $\boldsymbol{a}_n$.

Each part detector p provides one detected location $\hat{\boldsymbol{\mu}}_n^p$ per image n. The presence or absence of a part detection is denoted by the corresponding $\boldsymbol{h}_n^p \in \{0,1\}$. We now obtain the optimal model parameters $\mathbf{\Gamma}$ using a maximum a posteriori estimation given the detected locations $\hat{\boldsymbol{\mu}}$

$$\mathbf{\Gamma}^* = \operatorname*{argmax}_{\mathbf{\Gamma}} \mathbb{P}(\mathbf{\Gamma} \mid \hat{\boldsymbol{\mu}}) \,. \tag{1}$$

In contrast to a marginalization of the latent variables, we obtain an efficient learning algorithm. We apply Bayes' rule, the common assumption that training images and part proposals are independent given the model parameters [70], and independent priors for $\boldsymbol{b}$ and $\boldsymbol{s}$. In addition, we use a flat prior for $\boldsymbol{a}$ and $\boldsymbol{d}$, which means that there is no prior preference for object locations and part offsets, and obtain

$$\mathbf{\Gamma}^* = \operatorname*{argmax}_{\mathbf{\Gamma}} \prod_{n=1}^{N} \left(\prod_{p=1}^{P} \mathbb{P}(\hat{\boldsymbol{\mu}}_n^p \mid \mathbf{\Gamma}) \right) \cdot \mathbb{P}(\boldsymbol{b}) \cdot \mathbb{P}(\boldsymbol{s}) \,. \tag{2}$$

The term $\mathbb{P}(\hat{\boldsymbol{\mu}}_n^p \mid \mathbf{\Gamma})$ is the distribution of the predicted part locations given the model. If part p is used in view v of image n, we assume that the part location is normally distributed around the root location plus the shift vector, *i.e.*, $\hat{\boldsymbol{\mu}}_n^p \sim \mathcal{N}(\boldsymbol{a}_n + \boldsymbol{d}_v^p, (\sigma_v^p)^2 \boldsymbol{I})$ with $\boldsymbol{I}$ 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 image $\boldsymbol{x}^{(n)}$. Hence the distribution is given by

$$\mathbb{P}(\hat{\boldsymbol{\mu}} \mid \mathbf{\Gamma}) = \prod_{p=1}^{P} \mathcal{N}(\hat{\boldsymbol{\mu}}_n^p \mid \boldsymbol{a}_n + \boldsymbol{d}_v^p, (\sigma_v^p)^2 \boldsymbol{I})^{\boldsymbol{t}_{n,p}^v} \left(\frac{1}{|\boldsymbol{x}^{(n)}|} \right)^{1-\boldsymbol{t}_{n,p}^v} \tag{3}$$

where $\boldsymbol{t}_{n,p}^v = \boldsymbol{b}_v^p \cdot \boldsymbol{h}_n^p \cdot \boldsymbol{s}_n^v \in \{0,1\}$ indicates whether part p is used and visible in view v, which is itself active in image n. The prior distribution for the part selection $\boldsymbol{b}$ only captures the constraint that M parts need to be selected, *i.e.*, $\forall v : M = \sum_{p=1}^{P} \boldsymbol{b}_v^p$. The prior for the view selection $\boldsymbol{s}_n^v$ incorporates our constraint that a single view is selected for each image.

We will denote the induced set of feasible models by $\mathcal{M}$. 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

$$\operatorname*{argmin}_{\mathbf{\Gamma} \in \mathcal{M}} \sum_{n=1}^{N} \sum_{p=1}^{P} \sum_{v=1}^{V} \boldsymbol{t}_{n,p}^v \|\hat{\boldsymbol{\mu}}_n^p - (\boldsymbol{a}_n + \boldsymbol{d}_v^p)\|^2 \,. \tag{4}$$

In the supplementary material, we provide an intuitive interpretation and visualization of this optimization problem.

Eq. (4) is solved by alternating optimization of the model variables $\boldsymbol{b}$ and $\boldsymbol{d}$ as well as the latent variables $\boldsymbol{a}$ and $\boldsymbol{s}$ similar to the standard EM algorithm. For both $\boldsymbol{b}$ and $\boldsymbol{s}$, we can calculate the optimal value by sorting error terms. For example, $\boldsymbol{b}$ is calculated by analyzing

$$\operatorname*{argmin}_{\mathbf{\Gamma} \in \mathcal{M}} \sum_{p=1}^{P} \sum_{v=1}^{V} \boldsymbol{b}_v^p \underbrace{\left(\sum_{n=1}^{N} \boldsymbol{h}_n^p \boldsymbol{s}_n^v \|\hat{\boldsymbol{\mu}}_n^p - \boldsymbol{a}_n - \boldsymbol{d}_v^p\|^2 \right)}_{E(v,p) \geq 0} \,. \tag{5}$$

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 $E(v,p)$. This term describes, how well the part proposal p fits the view v. If its value is small, then the part proposal fits well to the view and should be selected. As $E(v,p) \geq 0$, the optimal parts are the ones with the smallest $E(v,p)$ for each view. In a similar manner, the view selection $\boldsymbol{s}$ can be determined.

The shift vectors $\boldsymbol{d}_v^{p*}$ for fixed $\boldsymbol{b}$, $\boldsymbol{s}$, and $\boldsymbol{a}$ are obtained by setting the derivative to 0:

$$\boldsymbol{d}_v^{p*} = \sum_{n=1}^{N} \boldsymbol{t}_{n,p}^v \cdot (\hat{\boldsymbol{\mu}}_n^p - \boldsymbol{a}_n) / \left(\sum_{n'=1}^{N} \boldsymbol{t}_{n',p}^v \right) . \tag{6}$$

The formulas are intuitive as the $\boldsymbol{d}_v^p$ are assigned the mean offset between anchor point $\boldsymbol{a}_n$ and observed part location $\hat{\boldsymbol{\mu}}_n^p$. The mean, however, is only calculated for images in which part p is used, *i.e.*, $\boldsymbol{t}_{n,p}^v = 1$. The anchor points can be obtained in a similar way.

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. 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 $\mathbf{\Gamma}$ with the best objective value is used. Learning the part model with 256 parts for each class of CUB200-2011 separately with five iterations per class took about 14.0 minutes on an Intel i7 processor with 3.4 GHz. The runtime is linear in the number of parts, views, and images, and hence does also scale well to models like ResNet-50, where we extracted 2048 parts. 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 analog to Eqs. (5) and (6) for fixed learned model parameters $\boldsymbol{b}$ and $\boldsymbol{d}$.

Part selection criteria. The learned part model is used to identify foreground object related parts. In the following, let $\boldsymbol{\nu}_p$, $1 \leq p \leq P$, be binary latent variables for the P parts denoting whether part p is related to the object. We formulate our selection as a maximum likelihood estimation

$$p^* = \operatorname*{argmax}_{1 \leq p \leq P} \mathbb{P}(\mathcal{X} \,|\, \boldsymbol{\nu}_p = 1) \tag{7}$$

$$= \operatorname*{argmax}_{1 \leq p \leq P} \prod_{n=1}^{N} \frac{\mathbb{P}(\boldsymbol{\nu}_p = 1 \,|\, \boldsymbol{x}^{(n)})\, \mathbb{P}(\boldsymbol{x}^{(n)})}{\mathbb{P}(\boldsymbol{\nu}_p = 1)}\,, \tag{8}$$

where $\mathcal{X} = \{\boldsymbol{x}^{(n)} \,|\, 1 \leq n \leq N\}$ denotes the training set and $\boldsymbol{x}^{(n)}$ are assumed to be independent samples. We assume a flat prior for $\mathbb{P}(\boldsymbol{x}^{(n)})$ and $\mathbb{P}(\boldsymbol{\nu}_p = 1)$ and apply $\log(\cdot)$:

$$p^* = \operatorname*{argmin}_{1 \leq p \leq P} \sum_{n=1}^{N} \log \mathbb{P}(\boldsymbol{\nu}_p = 1 \,|\, \boldsymbol{x}^{(n)})\,. \tag{9}$$

The model for $\mathbb{P}(\boldsymbol{\nu}_p = 1 \,|\, \boldsymbol{x}^{(n)})$ depends on the approach and the available annotation. In this work, we focus on the unsupervised case, where no location annotations are available at all. Details about the supervised and semi-supervised case can be found in [10].

We assume, that a part is relevant in a training image if it is visible and part of the view, which is selected for this image. This corresponds to the variable $\boldsymbol{t}^v_{n,p} = \boldsymbol{b}^p_v \cdot \boldsymbol{h}^p_n \cdot \boldsymbol{s}^v_n$ from above. If $\boldsymbol{t}^v_{n,p} = 1$, then the part is considered relevant. The probability of a part belonging to the foreground object is hence given by

$$\mathbb{P}(\boldsymbol{\nu}_p = 1 \,|\, \boldsymbol{x}^{(n)}) = \begin{cases} \epsilon & \text{if } \sum_{v=1}^{V} \boldsymbol{t}^v_{n,p} = 1 \\ 1 - \epsilon & \text{else} \end{cases}\,, \tag{10}$$

where $0.5 < \epsilon < 1$. The value of ϵ captures the confidence in the assumptions and is less than 1 as they might not hold in all cases. We use a constant value for all parts and obtain a simple selection scheme. The parts, which are used by the learned constellation model in most of the training images, are selected as most relevant. In contrast, parts occurring at random locations will be rarely selected by the part model and hence will also not be selected by Eq. (9).

5 Implicit Pose Normalization Approaches

Recent approaches based on a final global average [5] or bilinear pooling [2] before the classifier achieve a comparable accuracy as explicit pose normalization approaches. In this section, we show how these approaches achieve a pose normalization implicitly. We analyze the influence of the pooling strategy used before the classifier. The obtained insights will then lead to the development of α-pooling, which generalizes average and bilinear pooling and learns the optimal pooling strategy from data.

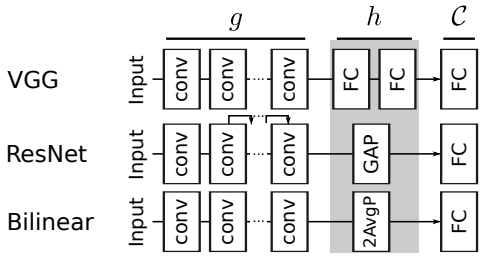

Fig. 6. Correspondence of functions defined in our model definition and computational blocks in common CNN architectures. Architectures are simplified for visualization purposes. Notation: `conv` - convolutional block, `FC` - linear transformation block, `GAP` - global average pooling, `2AvgP` - bilinear encoding.

Motivation. Recognition models based on a final global pooling operation can be defined by $(g, h, \mathcal{C})$ containing the local feature descriptor $g : \boldsymbol{x} \times l \mapsto \boldsymbol{y}_l \in \mathbb{R}^{\tilde{D}}$, the pooling function $h : \{\boldsymbol{y}_l\}_l \mapsto \boldsymbol{z} \in \mathbb{R}^D$ and a classifier $\mathcal{C} : \boldsymbol{z} \mapsto \boldsymbol{\nu} \in \mathbb{R}^C$. The function g takes an input image $\boldsymbol{x}$ and location index l to compute the appearance description $\boldsymbol{y}_l$ at this position. The function h aggregates all local descriptors into a single global representation $\boldsymbol{z}$ and the classifier $\mathcal{C}$ transforms that into the scores $\boldsymbol{\nu}$ over all known classes.

In the CNN architecture VGG-VD, g could be interpreted as the layers up to the last convolutional one, as they compute a grid of local features. The first two fully-connected layers correspond to h, because they transform the local features into a single global representation. Finally, the last linear layer can be interpreted as $\mathcal{C}$. Fig. 6 visualizes this assignment. In case of Residual Networks, the assignment is similar with the main difference that h is global average pooling, *i.e.*, $h^{\text{ave}}(\{\boldsymbol{y}_l\}_{l=1}^L) = \frac{1}{L}\sum_{l=1}^L \boldsymbol{y}_l \in \mathbb{R}^{\tilde{D}}$, $L \in \mathbb{N}$. The bilinear pooling model presented in [13] replaces h of these network architectures with the second-order operator presented in [3], *i.e.*, $h^{\text{bil}}(\{\boldsymbol{y}_l\}_{l=1}^L) = \frac{1}{L}\sum_{l=1}^L \boldsymbol{y}_l \boldsymbol{y}_l^\mathsf{T} \in \mathbb{R}^{\tilde{D} \times \tilde{D}}$.

The global average or bilinear pooling before the classifier leads to an implicit matching of local features during classification, more specifically in the similarity function of the classifier $\mathcal{C}$. For example, suppose we use global average pooling for h and a linear classifier for $\mathcal{C}$, *i.e.*, $\mathcal{C}(\boldsymbol{z}) = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}$, $\boldsymbol{W} \in \mathbb{R}^{C \times D}$, $\boldsymbol{b} \in \mathbb{R}^C$. This is a common selection and used in ResNet, for example. The behavior of such a model can be analyzed using the distance function used by the classifier. In this case, we hence analyze the linear kernel between two average pooled vectors $\boldsymbol{z} = \sum_{l=1}^L \boldsymbol{y}_l$ and $\boldsymbol{z}' = \sum_{m=1}^L \boldsymbol{y}'_m$ as follows:

$$\begin{aligned} \langle \boldsymbol{z}, \boldsymbol{z}' \rangle &\propto \langle \operatorname{vec}\left(\sum_{l=1}^{L} \boldsymbol{y}_l\right), \operatorname{vec}\left(\sum_{m=1}^{L} \boldsymbol{y}'_m\right) \rangle \\ &= \operatorname{tr}\left(\left(\sum_{l=1}^{L} \boldsymbol{y}_l\right)^\mathsf{T} \left(\sum_{m=1}^{L} \boldsymbol{y}'_m\right)\right) \qquad = \sum_{l,m} \langle \boldsymbol{y}_l, \boldsymbol{y}'_m \rangle\,, \end{aligned} \tag{11}$$

where we omit the normalization with respect to L^2 for brevity. As can be seen, the similarity of $\boldsymbol{z}$ and $\boldsymbol{z}'$ is proportional to the aggregated pairwise similarity of the

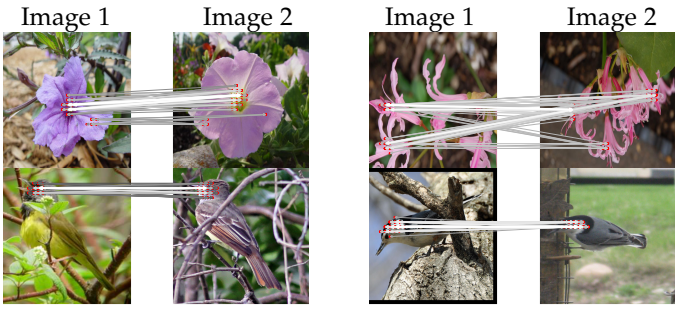

Fig. 7. Visualization of local regions which have the largest influence on the linear kernel. The brighter and thicker the line, the larger is the corresponding inner product between these local features.

local features. In this paper, we call this *pairwise matching* as presented in [6] before.

The $\langle \boldsymbol{z}, \boldsymbol{z}' \rangle$ in Eq. (11) is mainly influenced by the largest inner products between local features $\langle \boldsymbol{y}_l, \boldsymbol{y'}_m \rangle$. Fig. 7 shows these for VGG-VD and randomly chosen image pairs showing flowers and birds. It can be seen that the highest inner products occur at relevant semantic object parts in all fine-grained tasks under consideration. Consequently, the similarity of two images is mostly influenced by the similarity of corresponding semantic parts occurring in these images. As all possible location pairs are matched, the actual location of a part does not influence the overall similarity of two images. Hence these pooling strategies achieve implicitly robustness against pose variations.

In case of bilinear pooling, the similarity function is given by $\sum_{l,m} \langle \boldsymbol{y}_l, \boldsymbol{y'}_m \rangle^2$ and hence only differs in the square. It increases the relative influence of the highest matches considerably. This is useful for tasks like bird recognition, where only a few object parts are important. However, each recognition task has different properties. In other tasks like scene recognition, a wide range of elements in the scene might be important.

Hence it would be desirable to automatically learn the amount of focus on the highest matchings from data. We present α-pooling, which achieves that and also contains global average and bilinear pooling as special cases. We first present the model definition and afterward explain how α steers the pairwise matching.

Definition. Our α-pooling generalizes h^{ave} and h^{bil} to the learnable, parametric pooling operator h^{alpha}. While it could be used at multiple locations in a CNN architecture, we only consider it for the final encoding step before the classifier in this work. In order to achieve the desired learnable focus on the largest matches in the similarity function of the final classifier, we define it as

$$h^{\text{alpha}}(\{\boldsymbol{y}_l\}_{l=1}^L) = \text{vec}\left(\frac{1}{L}\sum_{l=1}^{L} \text{alpha-prod}(\boldsymbol{y}_l, \alpha)\right) \tag{12}$$

with vectorization function $\text{vec}(\cdot)$ and local feature encoding

$$\text{alpha-prod}(\boldsymbol{y}_l, \alpha) = (\text{sgn}(\boldsymbol{y}_l) \circ |\boldsymbol{y}_l|^{\alpha-1})\boldsymbol{y}_l^\mathsf{T} . \tag{13}$$

The operations $\cdot^\alpha$, sgn, $\circ$ and $|\cdot|$ denote element-wise functions for power, sign, multiply and absolute value. The local feature encoding $\text{alpha-prod}(\cdot, \alpha)$ computes the outer product of the local feature to the power of $\alpha - 1$ and the original feature $\boldsymbol{y}_l$. The power is calculated with the absolute value as the continuous real exponentiation requires non-negative bases and the $\text{sgn}(\cdot)$ function is used to preserve the sign. The result of the α-pooling operation has the shape $\mathbb{R}^{\tilde{D}\cdot\tilde{D}}$, which is identical to the shape of the vectorized bilinear pooling output.

The $\alpha \in \mathbb{R}$ is a differentiable model parameter, which controls the pooling strategy. It is learned from data with back propagation as part of the main CNN training. The gradient of the pooling output with respect to α is given by

$$\frac{\partial h^{\text{alpha}}}{\partial \alpha} = \text{vec}\left(\frac{1}{L}\sum_{l=1}^{L} \boldsymbol{y}_l(\text{sgn}(\boldsymbol{y}_l) \circ |\boldsymbol{y}_l|^{\alpha-1} \circ \log|\boldsymbol{y}_l|)^\mathsf{T}\right). \tag{14}$$

A decay on α was not necessary in our experiments to achieve convergence. For numerical stability of h^{alpha} and its gradient wrt. α, we add a small constant $\delta > 0$ when calculating the power and logarithm.

The definition of α-pooling includes global average and bilinear pooling as special cases. Average pooling can be obtained with non-negative $\boldsymbol{y}_l$ and model parameter $\alpha = 1$, *i.e.*, $\text{alpha-prod}(\boldsymbol{y}_l, 1) = \boldsymbol{e}\boldsymbol{y}_l^\mathsf{T}$. The result is the vectorization of a matrix containing $\boldsymbol{y}_l$ in every row By increasing the value of α continuously, the pooling strategy shifts towards bilinear pooling, which is reached at $\alpha = 2$: $\text{alpha-prod}(\boldsymbol{y}_l, 2) = \boldsymbol{y}_l\boldsymbol{y}_l^\mathsf{T}$. Since α is learned from data, the right pooling strategy is automatically chosen. The smooth interpolation also offers the possibility to analyze differences when moving between global average and bilinear pooling. Furthermore, we show how α allows for controlling the degree of pose normalization.

Influence of $\boldsymbol{\alpha}$. We can analyze the similarity function between features obtained with α-pooling as in Eq. (11):

$$\langle \boldsymbol{z}, \boldsymbol{z}' \rangle \propto \sum_{l,m} \langle \boldsymbol{y}_l, \boldsymbol{y'}_m \rangle \langle \boldsymbol{y}_l^{\alpha-1}, \boldsymbol{y'}_m^{\alpha-1} \rangle . \tag{15}$$

Compared to average pooling, it includes the factor $\langle \boldsymbol{y}_l^{\alpha-1}, \boldsymbol{y'}_m^{\alpha-1} \rangle$. This factor vanishes for $\alpha = 1$ and hence is identical to average pooling in this case. For bilinear pooling with $\alpha = 2$, the factor is equal to the inner product between the local features and hence we obtain the squared inner product as before. This formulation allows for analyzing the influence of α and its meaning for fine-grained recognition. The first observation is that the larger the value of α, the bigger is the relative influence of large inner products on the similarity value. In other words, the similarity value between two global features is dominated by the similarity value of the pair of local features, which have the highest inner product. This is similar to p-norms, for which the influence of the largest element increases with increasing p until converging to the maximum norm. In α-pooling, a very large α has the comparable effect. In Fig. 11 of the experiments, we will support this theoretical analysis with an quantitative evaluation.

Analyzing decisions. A main drawback of implicit pose normalization approaches is interpretability. While we can easily visualize the parts used in the prediction of an explicit pose normalization model, the recognition process is rather unclear for implicit approaches. We present a visualization approach, which uses the pairwise matching formulation to relate a prediction to the most influential

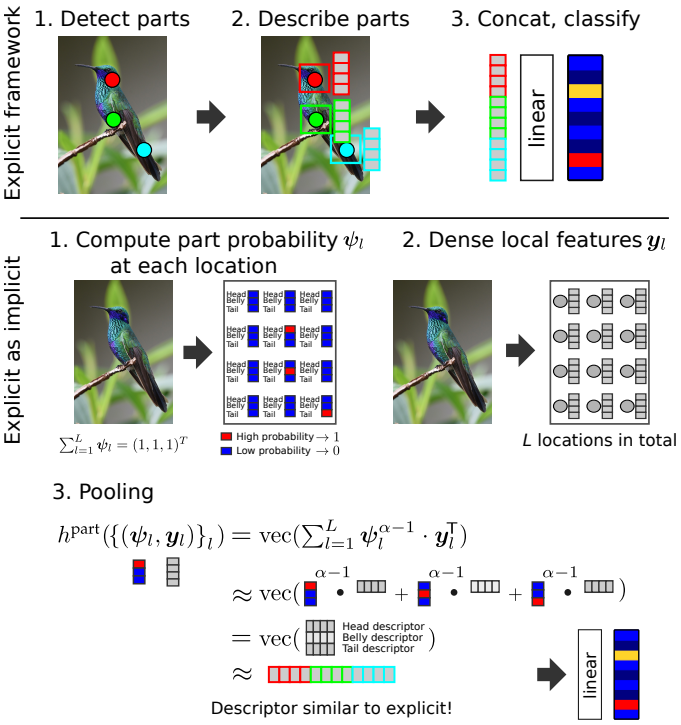

Fig. 8. Visualization on how to express explicit as implicit models.

pairs of regions in the test and training image regions. Please note that, in contrast to Section 3, we can only analyze the classifier and do not include the local feature extraction.

Let $\mathcal{C}$ be the learned classifier as before. The representer theorem [68], [69] states, that there exists a dual representation of $\mathcal{C}$ with $\beta_n \in \mathbb{R}$, $1 \leq n \leq N$, such that a prediction for an image can be computed by

$$\sum_{n=1}^{N} \beta_n \langle \boldsymbol{z}, \boldsymbol{z}_n \rangle = \sum_{n=1}^{N} \sum_{l,m} \beta_n \langle \boldsymbol{y}_l, \boldsymbol{y}_m^n \rangle \langle \boldsymbol{y}_l^{\alpha-1}, (\boldsymbol{y}_m^n)^{\alpha-1} \rangle \,, \quad (16)$$

where N is the number of training samples. The $\boldsymbol{z}$ is the α-pooled feature of the current test image computed from local features $\{\boldsymbol{y}_l\}_l$. Accordingly, $\boldsymbol{z}_n$ denotes the α-pooled feature of the n-th training image computed from local features $\{\boldsymbol{y}_m^n\}_m$. The β_n are the weights of each training image as also used in the dual formulation of an SVM, for example. However, the primal variant of the classifier as learned by CNNs can also be transformed into the dual formulation. Given this formulation, we can define the influence of a pair of training and test image region l and m by

$$\gamma_{l,m}^n = \beta_n \langle \boldsymbol{y}_l, \boldsymbol{y}_m^n \rangle \langle \boldsymbol{y}_l^{\alpha-1}, (\boldsymbol{y}_m^n)^{\alpha-1} \rangle \,, \quad (17)$$

which is a single summand of Eq. (16). In the experiments, we present how this influence can be used to visualize the most influential training image regions for a given image. In addition, we will use it to quantify the influence of semantic parts on the prediction and to analyze the influence of α.

α-pooling as explicit pose normalization model. While α-pooling models implicitly perform pose normalization, it is possible to obtain an explicit pose normalization model by only slightly modifying the α-pooling framework. Fig. 8 shows a schematic explanation of the relationship.

It now consists of $(g, \Psi, h^{\text{part}}, \mathcal{C})$. In contrast to before, the formulation now contains a part detector function $\Psi : \boldsymbol{x}^{(n)} \times l \mapsto \boldsymbol{\psi}_l \in (0,1)^P$, $\sum_{l=1}^{L} \boldsymbol{\psi}_l = \boldsymbol{e}_P$, providing for all P parts and L locations in an image the probability that the part is present. The pooling function $h^{\text{part}} : \{(\boldsymbol{\psi}_l, \boldsymbol{y}_l)\}_l \mapsto \text{vec}(\sum_{l=1}^{L} \boldsymbol{\psi}_l^{\alpha-1} \cdot \boldsymbol{y}_l^T) \in \mathbb{R}^D$ now takes both the local features $\boldsymbol{y}_l$ and the detector scores $\boldsymbol{\psi}_l$ as input. It aggregates over all locations the outer product between the detector scores $\boldsymbol{\psi}_l$ to the power of α and the local feature $\boldsymbol{y}_l$. As $\boldsymbol{\psi}_l$ is always positive, we can omit the sign and absolute value function.

This classification model converges to explicit part modeling if there is only one single location with high detection score for each part, *i.e.*, $\boldsymbol{\psi}_l(p) \to 1$ for one location l_p for every part p. The resulting feature vector consists of the local features from only these locations. This special case also allows for investigating the influence of α. If $\alpha = 1$, the part detection scores are ignored and global average pooling is performed. The larger the value of α, the larger is the influence of the location with the highest detection score for a given part. Hence the larger the value of α, the more α-pooling converges towards explicit pose normalization. A major difference to our explicit pose normalization model presented in Section 4 is the part description. In Section 4, we crop image patches of parts and compute features independently. In contrast, the features from a convolutional layer are used in implicit pose normalization.

As detection scores in the real world are not as ideal as assumed here, implicit approaches are faced with higher requirements for the local features. They require local feature descriptors which allow for distinguishing a significantly higher number of patterns. While in explicit pose normalization, only features from corresponding semantic parts influence the similarity value, all features from all locations in the image are influencing in case of implicit pose normalization. Hence it is important that the local features allow for distinguishing between background and objects parts as well as between different semantic parts.

Another difference between both approaches is the use of prior knowledge. Our explicit pose normalization approach was able to exploit the prior knowledge that a constellation of object parts exists. As we will show in the experiments, this leads to an improved accuracy compared to randomly selected part proposals. The implicit approach currently cannot exploit such knowledge and it might be a fruitful research direction in future work.

6 Experiments

Our experimental evaluation aims at comparing the approaches in terms of recognition accuracy as well as the influence of different components. We present qualitative and quantitative results for our visualizations as well.

Setup. We evaluate on CUB200-2011 birds [1] (200 classes, 11788 images), Oxford Flowers 102 [71] (102 classes, 8189 images), Oxford-IIIT Pets [35] (37 classes, 7349 images), and Stanford 40 actions [9] (40 classes, 9532 images). We also present results using Stanford cars in the supplementary material. Provided splits are used and the mean class-wise accuracy is reported. We show results for AlexNet [7], VGG-VD [8] and ResNet-50 [5] pre-trained on ILSVRC 2012 [21]. We use the pre-trained model of [72] for ResNet-50. Deeper

models were not possible due to GPU memory limitations when using large input sizes. In case of Section 4, the last convolutional layer was used for part discovery and the second to last fully-connected layer was used for feature description. All models are fine-tuned using random cropping and flip augmentation. The part model learning of Section 4.3 is done for each class separately with 5 views, 10 parts per view, and is repeated 5 times. Given the part locations, we crop a patch of height and width $\sqrt{\lambda \cdot W \cdot H}$, $\lambda \in \{\frac{1}{5}, \frac{1}{16}\}$, where W and H are the width and height of the uncropped image, respectively. The features of missing detections are calculated on a mean image over the training set. Features of all parts are concatenated and a linear SVM is trained using flip augmentation. In contrast to [11], we do not estimate the bounding box for better comparison of the implicit and explicit normalization approaches, which decreased the accuracy especially in pet classification with AlexNet. We also ran the fine-tuning longer resulting in slightly increased results. Hyperparameters were optimized using cross-validation. In case of α-pooling, we initialized α with 2.0. Since α is learned from data, the initialization has no influence. All parameters including α and the parameters of g are fine-tuned using flip augmentation and random cropping. The learning rate multiplier for α was set to $\kappa = 10$. The learning rate is constant at 0.001 and a batch size of 8 or 16 was used. In the benchmark results, we use an input resolution of 448×448 pixels and follow the feature normalization of [13], *i.e.,* matrix root, element-wise signed square root, and L^2-normalization, but do not use the matrix root for ResNet-50 on flowers, pets, and actions as it yielded a significant decrease in accuracy. The code of our approaches is available on Github and links are provided in the supplementary material. We also discuss the run time of our approaches there.

Bird classification results. Table 2 compares the different approaches for bird species classification on CUB200-2011 and the benefit of human annotations like bounding box and semantic part locations. The baseline accuracy is obtained with a fine-tuned network and is 52.5%, 71.9%, and 80.4% for AlexNet, VGG-VG, and ResNet-50, respectively. Our approaches can consistently and significantly improve these results by up 22.9% absolute increase.

In the category of explicit pose normalization, the ResNet architecture achieves the highest accuracy of all approaches with 83.4% compared to 81.4% of VGG-VD and 68.5% of AlexNet. We also compare to randomly selected parts, which is a reasonable approach if the part constellation assumption is not valid. These randomly selected parts already improve the recognition significantly. The proposed neural activation constellations model allows to select more discriminative parts and we can improve accuracy further.

We also compare our discovered parts to ground-truth part locations. As expected, they increase the accuracy, especially for AlexNet and ResNet. The part locations obtained by VGG-VD, however, achieve almost the same accuracy as the ground-truth parts. It seems that VGG parts are already very pure and discriminative. This observation is probably connected to the fact, that most works on fine-grained recognition use VGG-VD for part discovery and recognition. It seems that our assumption, that the channels of last convolutional layer can be interpreted as object part detectors, applies to the VGG architecture the best.

Moving on to implicit pose normalization with the presented α-pooling, ResNet-50 obtains the highest accuracy compared to the other architectures with 86.5%. VGG-VD with α-pooling achieves almost the same accuracy with 86.1%. The largest gain can be observed for AlexNet, with an absolute improvement of 22.9% compared to the baseline.

Flower classification results. The results for the Oxford 102 classification task are shown in Table 3. The experiments on this dataset are particularly interesting as flowers do not have obvious shared semantic parts. We still observe a moderate increase in accuracy with models. In case of AlexNet, randomly selected parts do not help to distinguish the flowers. However, the proposed constellation model-based selection improves the accuracy by 1.2% and α-pooling even by another 2%. In case of VGG, random parts are already helpful and achieve a small increase of 0.9%. The constellation model and α-pooling improve the accuracy further similar to AlexNet. For ResNet-50, only the implicit pose normalization approach achieves an improvement. The overall only moderate increase in accuracy might be caused by the already very high baseline accuracy. In addition, the lack of shared object parts most likely prevents larger gains.

Pets classification. Table 3 also presents the result on the Oxford-IIIT Pets dataset. Overall the observations of the flowers dataset also apply in this case. For example, the accuracy of AlexNet improves by 2.3% with explicit pose normalization and another 4.2% if α-pooling is used. The results of ResNet-50 are quite mixed showing a small decrease for the explicit pose normalization approach and a slight increase with α-pooling.

Action recognition. We are also interested in the transferability of fine-grained specific approaches to other tasks. In particular, we investigate the accuracy of action recognition from still images in Table 3. Our approaches consistently improve the recognition process for all models. There is even a significant improvement with ResNet-50 on the actions dataset ranging from 1.1% for the explicit to 3.6% for implicit pose normalization. The improvement is comparable for all models AlexNet, VGG, and ResNet-50.

Influence of the number of parts. A major difference between the explicit and implicit approach is the feature dimension. While the implicit approach uses all parts available, our part selection in the explicit approaches compresses the feature representation to relevant parts only. Fig. 9 presents a study on how the number of selected parts influences the accuracy using VGG-VD and only one extracted patch per part. The baseline accuracy is 71.9% as before. Adding only the most relevant part improves the accuracy by 5.8% and only two parts are required to achieve 78.6%. In contrast, randomly selected parts increases the accuracy much slower as the selection might also include noise detectors. Two selected parts only increase the accuracy a bit to 72.8% and ten parts are required for 76.6%. Hence our constellation-based part selection is well suited for feature selection in this case.

Influence of the pooling strategy. The main parameter of our implicit pose normalization approach is the learned parameter α, which controls the focus on the largest matches. We perform an ablation study about the recognition accuracy using VGG-VD in Fig. 10. We fix the value of α

TABLE 2
Comparison of our explicit pose normalization approaches on the CUB200-2011 birds dataset.

Method	Training annotation Bbox	Training annotation Parts	Test annotation Bbox	Test annotation Parts	AlexNet	VGG-VG	ResNet-50
Previous							
Donahue *et al.* [43], JMLR'14	✓		✓		58.8%	-	-
Zhang *et al.* [40], ECCV'14	✓	✓			73.9%	-	-
Branson *et al.* [39], BMVC'14	✓	✓			75.7%	-	-
Krause *et al.* [41], CVPR'15	✓				-	82.0%	-
Zhang *et al.* [47], CVPR'16					-	84.5%	-
Liu *et al.* [50], PAMI'16	✓		✓		-	77.0%	-
Zhang *et al.* [73], CVPR'16	✓	✓	✓		-	84.6%	-
Lin *et al.* [13], BMVC'17					-	85.8%	-
Zheng *et al.* [74], ICCV'17					-	86.5%	-
Li *et al.* [4], arXiv'17					-	-	86.0%
Ours							
No part modeling					52.2%	71.9%	80.4%
Explicit (Random part selection)					59.2% ± 0.7%	78.6% ± 0.2%	82.8% ± 0.3%
Explicit (Constellation model)					68.5%	81.4%	83.4%
Explicit (GT parts)		✓		✓	76.0%	82.0%	86.1%
Implicit (α-pooling)					75.1%	86.1%	86.5%

TABLE 3
Comparison of the presented pose normalization approaches on the Oxford flowers 102, Oxford-IIIT Pets, and Stanford 40 actions datasets.

Method		Oxford Flowers 102	Oxford-IIIT Pets	Stanford 40 actions
Previous		84.6% [75], 86.8% [44], 91.3% [76], 94.8% [79], 96.1% [83], 96.6% [81]	88.1% [76], 88.2% [77], 91.4% [80], 92.2% [81],	72.0% [78], 80.9% [79], 81.7% [82]
Ours				
AlexNet	No part modeling	90.9%	80.5%	63.8%
	Explicit (Random part selection)	90.4%±0.7%	80.5%±0.7%	63.3%±0.5%
	Explicit (Constellation model)	92.1%	82.8%	65.6%
	Implicit (α-pooling)	94.1%	87.0%	68.8%
VGG-VD	No part modeling	93.7%	91.1%	80.5%
	Explicit (Random part selection)	94.6%±0.4%	91.3%±0.3%	82.4%±0.4%
	Explicit (Constellation model)	95.5%	91.8%	83.3%
	Implicit (α-pooling)	97.1%	93.2%	86.1%
ResNet-50	No part modeling	95.7%	93.6%	84.1%
	Explicit (Random part selection)	95.6%±0.3%	91.1%±0.2%	84.2%±0.4%
	Explicit (Constellation model)	95.7%	91.6%	85.2%
	Implicit (α-pooling)	**96.7%**	**94.2%**	**87.7%**

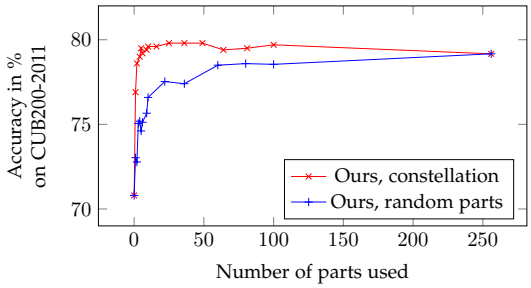

Fig. 9. Influence of the number of selected parts on the bird classification accuracy on CUB200-2011 with VGG-VD. Compared to the other experiments, only one patch was extracted per part proposal.

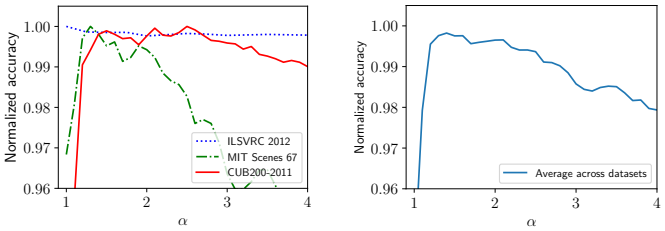

Fig. 10. Influence of α using VGG16 without fine-tuning. $\alpha = 1$ corresponds to average pooling and $\alpha = 2$ to bilinear pooling. α is manually set in this experiment.

and learn a logistic regression classifier without fine-tuning on ILSVRC 2012, MIT scenes 67, and CUB200-2011 birds to compare the behavior on different types of classification tasks. In case of ILSVRC 2012, we only use 130 training images per category for efficiency reasons. The dataset contains a wide range of categories including numerous fine-grained categories. This seems to result in no particular preference for specific values of alpha. However, when moving towards specialized datasets, the properties of the task change and adapting the pooling strategy is beneficial. The accuracy improves when moving from average pooling ($\alpha = 1$) to a strategy between average and bilinear pooling

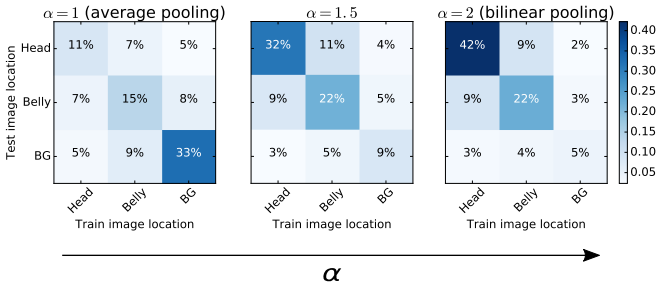

Fig. 11. Influence of α on the contribution of different bird body parts to the classification decision on CUB200-2011. The higher the value of α, the higher is the influence of the actual bird body parts to the classification decision. α is manually set to $\{1, 1.5, 2\}$.

until around $\alpha = 1.5$. The accuracy drops quickly after that for scene recognition. For CUB200-2011, it increases a bit further with a peak at $\alpha = 2.5$ for CUB and only slowly decreases afterward. This shows that the learned pooling strategy allows for better adapting to the properties of a classification task. The right figure shows the average curve across all datasets. We conclude that $\alpha = 1.5$ is a good value if no fine-tuning is used. Please note that we learned α in the benchmark results and results here only apply to the case of no fine-tuning. In the supplementary material, we evaluate how learning α can increase the accuracy and also compare the learned value of α for different input sizes.

We also investigate the influence of α on the contribution of body parts on the classification on CUB200-2011 in Fig. 11. Section 5 presented an approach for measuring the influence of local matches on the classification score. We relate this influence to semantic body parts of birds using ground-truth annotations. The influence is plotted for the values $\alpha \in \{1, 1.5, 2\}$. As can be seen, the focus on the bird itself increases significantly with larger α. In particular, the focus on the head increases from 11% for $\alpha = 1$ to 42% for $\alpha = 2$. At the same time, the role of background decreases. This supports the theoretical analysis, that larger values of α focus the decision on a few relevant object parts.

Classification visualization for α-pooling. The implicit approach α-pooling often outperforms our explicit model, but lack its interpretability. We explained in Section 5 how to address this by computing the influence of training image regions on the decision of a given test image. In Fig. 12, we show the most contributing matches with the highest $\gamma^n_{l,m}$ for our implicit pose normalization approach. We show two correct and two false predictions taken from CUB200-2011 and Oxford flowers 102. In each prediction, we visualize the test image in the bottom left and the most relevant training image regions of the predicted class around it. In case of the correct predictions, the focus is as expected on important object parts. However, analyzing false prediction delivers more enlightening insights. For example, in the bottom right, the model confused the dark red young flowers in the top right of the image with the similarly looking stamen of the predicted class. These insights allow for identifying weaknesses and hence for successfully improving approaches. In the supplementary, we provide more examples and also show the activation flow of α-pooling-based models.

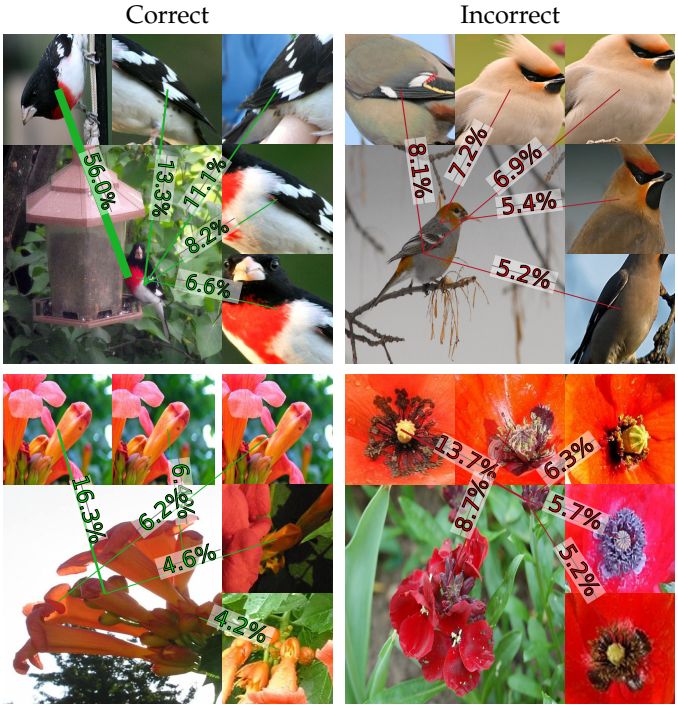

Fig. 12. Visualization of the most influential training image regions as presented in Section 5. We show two correct on the left and two false predictions on the right. Each test image is surrounded by five training regions, whose annotated influence is given by Eq. (17).

7 Conclusions

This paper compares recent concepts for fine-grained recognition and presents improvements based on the obtained observations. We include common CNN architectures using several fully-connected layers, the explicit pose normalization approach called neural activation constellations, and the global pooling approach α-pooling. Our visualization technique activation flow showed that common CNN architectures like AlexNet and VGG-VD lack the ability to handle large pose variations due to scarce training data and hence are less suited for fine-grained recognition. The comparison of neural activation constellations and α-pooling revealed that both can significantly improve the accuracy of common CNN models like AlexNet, VGG-VG, and ResNet by up to 22.9%. While α-pooling is often slightly leading, it lacks the clear interpretability and the incorporation of prior knowledge used in the explicit approach. The former is addressed with a visualization of classification decisions, which relates test to training images regions. The latter aspect remains open for future research. The comparison also revealed possible directions for explicit approaches such as using the whole part detection map instead of peak detections only.

Acknowledgments

Part of this research was supported by grant RO 5093/1-1 of the German Research Foundation (DFG). The authors thank Nvidia for GPU donations.

REFERENCES

[1] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[2] T. Lin, A. Roy Chowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *ICCV*, 2015, pp. 1449–1457.

[3] J. Carreira, R. Caseiro, J. P. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *ECCV*, 2012, pp. 430–443.

[4] Z. Li, Y. Yang, X. Liu, S. Wen, and W. Xu, "Dynamic computational time for visual attention," *CoRR*, vol. abs/1703.10332, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[6] L. Bo and C. Sminchisescu, "Efficient match kernel between sets of features for visual recognition," in *NIPS*, 2009, pp. 135–143.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[9] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *ICCV*, 2011, pp. 1331–1338.

[10] M. Simon, E. Rodner, and J. Denzler, "Part detector discovery in deep convolutional neural networks," in *ACCV*, 2014, pp. 162–177.

[11] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *ICCV*, 2015, pp. 1143–1151.

[12] M. Simon, Y. Gao, T. Darrell, J. Denzler, and E. Rodner, "Generalized orderless pooling performs implicit salient matching," in *ICCV*, 2017, pp. 4970–4979.

[13] T.-Y. Lin and S. Maji, "Improved bilinear pooling with cnns," in *BMVC*, 2017.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.

[16] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV Workshops*, no. 1-22, 2004, pp. 1–2.

[17] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *CVPR*, 2007, pp. 1–8.

[18] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010, pp. 143–156.

[19] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010, pp. 3304–3311.

[20] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006, pp. 2169–2178.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[22] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *CoRR*, vol. abs/1603.05027, 2016.

[24] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.

[25] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *NIPS*, 2015, pp. 2377–2385.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

[27] ——, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *PAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.

[28] M. Zobel, A. Gebhard, D. Paulus, J. Denzler, and H. Niemann, "Robust facial feature localization by coupled features," in *International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 2–7.

[29] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *CVPR*, 2003, pp. 264–271.

[30] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *ECCV*, 2010, pp. 438–451.

[31] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *CVPR*, 2011, pp. 1577–1584.

[32] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *ECCV*, 2012, pp. 172–185.

[33] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Computer Vision, Graphics and Image Processing*, 2008, pp. 722–729.

[34] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *CVPR Workshops*, 2011.

[35] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *CVPR*, 2012, pp. 3498–3505.

[36] N. Zhang, R. Farrell, and T. Darrell, "Pose pooling kernels for subcategory recognition," in *CVPR*, 2012, pp. 3665–3672.

[37] B. Yao, G. Bradski, and L. Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in *CVPR*, 2012, pp. 3466–3473.

[38] C. Göring, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *CVPR*, 2014, pp. 2489–2496.

[39] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Improved bird species categorization using pose normalized deep convolutional nets," in *BMVC*, 2014.

[40] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *ECCV*, 2014, pp. 834–849.

[41] J. Krause, H. Jin, J. Yang, and F. Li, "Fine-grained recognition without part annotations," in *CVPR*, 2015, pp. 5546–5555.

[42] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *ICCV*, 2013, pp. 1713–1720.

[43] 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, pp. 647–655.

[44] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR*, 2014, pp. 806–813.

[45] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, and A. Zisserman, "Tricos: A tri-level class-discriminative co-segmentation method for image classification," in *ECCV*, 2012, pp. 794–807.

[46] 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, pp. 842–850.

[47] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," in *CVPR*, 2016, pp. 1134–1142.

[48] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, "Multiple granularity descriptors for fine-grained categorization," in *ICCV*, 2015, pp. 2399–2406.

[49] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.

[50] L. Liu, C. Shen, and A. van den Hengel, "Cross-convolutional-layer pooling for image recognition," *PAMI*, vol. 39, no. 11, pp. 2305–2313, 2017.

[51] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *ECCV*, 2016, pp. 301–320.

[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[53] F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *PAMI*, vol. 28, no. 4, pp. 594–611, 2006.

[54] E. Riabchenko, J.-K. Kämäräinen, and K. Chen, "Density-aware part-based object detection with positive examples," in *CVPR*, 2014, pp. 2814–2819.

[55] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *NIPS*, 2012, pp. 3122–3130.

[56] D. J. Crandall and D. P. Huttenlocher, "Composite models of objects and scenes for category recognition," in *CVPR*, 2007, pp. 1–8.

[57] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.
[58] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *ICCV*, 2013, pp. 729–736.
[59] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014, pp. 818–833.
[60] A. Mordvintsev, C. Olah, and M. Tyka, "Deepdream - a code example for visualizing neural networks," 2015, https://research.googleblog.com/2015/07/deepdream-code-example-for-visualizing.html. [Online; accessed 23. August 2017].
[61] ——, "Inceptionism: Going deeper into neural networks," 2015, https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html. [Online; accessed 23. August 2017].
[62] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *Technical report*, vol. 1341, 2009.
[63] T.-Y. Lin and S. Maji, "Visualizing and understanding deep texture representations," in *CVPR*, 2016, pp. 2791–2799.
[64] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *ICML Workshops*, 2015.
[65] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, 2015.
[66] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013.
[67] T. Jost, N. Ouerhani, R. Von Wartburg, R. Müri, and H. Hügli, "Assessing the contribution of color in visual attention," *CVIU*, vol. 100, no. 1, pp. 107–123, 2005.
[68] B. Schölkopf, R. Herbrich, and A. Smola, "A generalized representer theorem," in *Computational Learning Theory*, 2001, pp. 416–426.
[69] A. Argyriou, C. A. Micchelli, and M. Pontil, "When is there a representer theorem? vector versus matrix regularizers," *JMLR*, vol. 10, no. Nov, pp. 2507–2529, 2009.
[70] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *CVPR*, 2009, pp. 1014–1021.
[71] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *CVPR*, 2006, pp. 1447–1454.
[72] M. Simon, E. Rodner, and J. Denzler, "Imagenet pre-trained models with batch normalization," *CoRR*, vol. abs/1612.01452, 2016.
[73] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas, "SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition," in *CVPR*, 2016, pp. 1143–1152.
[74] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *ICCV*, 2017, pp. 5209–5217.
[75] N. Murray and F. Perronnin, "Generalized max pooling," in *CVPR*, 2014, pp. 2473–2480.
[76] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *CVPR*, 2015, pp. 36–45.
[77] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *TIP*, vol. 26, no. 6, pp. 2868–2881, 2017.
[78] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016, pp. 2921–2929.
[79] S. Cai, L. Zhang, W. Zuo, and X. Feng, "A probabilistic collaborative representation based approach for pattern classification," in *CVPR*, 2016, pp. 2950–2959.
[80] X. Zhang, H. Xiong, W. Zhou, and Q. Tian, "Fused one-vs-all features with semantic alignments for fine-grained visual categorization," *TIP*, vol. 25, no. 2, pp. 878–892, 2016.
[81] G. Xie, X. Zhang, W. Yang, M. Xu, S. Yan, and C. Liu, "LG-CNN: from local parts to global discrimination for fine-grained recognition," *Pattern Recognition*, vol. 71, pp. 118–131, 2017.
[82] A. Rosenfeld and S. Ullman, "Visual concept recognition and localization via iterative introspection," in *ACCV*, 2016, pp. 264–279.
[83] C. Huang, H. Li, Y. Xie, Q. Wu, and B. Luo, "Pbc: Polygon-based classifier for fine-grained categorization," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 673–684, 2017.

Marcel Simon received the B.Sc. and M.Sc. degrees in computer science from Friedrich-Schiller-Universität Jena, Germany, in 2011 and 2014, respectively. He is currently working at the same university towards the PhD degree at the Computer Vision Group under the supervision of Joachim Denzler. His research interests are in the field of image classification, fine-grained recognition, and part-based models.

Erik Rodner earned the Diploma degree in Computer Science with honours in 2007 from the Friedrich Schiller University Jena, Germany. He received his PhD in 2011 with summa cum laude for his work on learning with few examples, which was done under supervision of Joachim Denzler at the computer vision group of the University of Jena. From 2012 to 2013, Erik joined UC Berkeley and the International Computer Science Institute as a postdoctoral researcher. He was now senior researcher and lecturer in the computer vision group at the University of Jena from 2013 to 2016 and is now researcher at Carl Zeiss AG. His research interests include domain adaptation, deep learning, visual object discovery, active and continuous learning, and scene understanding.

Trevor Darrell received the BSE degree from the University of Pennsylvania in 1988, having started his career in computer vision as an undergraduate researcher in Ruzena Bajcsys GRASP lab. He received the SM and PhD degrees from MIT in 1992 and 1996, respectively. His group is located at the University of California, Berkeley, where he is on the faculty of the CS and EE Divisions of the EECS Department. His group develops algorithms for large-scale perceptual learning, including object and activity recognition and detection, for a variety of applications including multimodal interaction with robots and mobile devices. His interests include computer vision, machine learning, computer graphics, and perception-based human computer interfaces. He was previously on the faculty of the MIT EECS department from 1999-2008, where he directed the Vision Interface Group. He was a member of the research staff at Interval Research Corporation from 1996-1999. He is a member of the IEEE.

Joachim Denzler earned the Diploma degrees "Diplom-Informatiker", "Dr.-Ing." and "Habilitation" from the University of Erlangen, Germany, in years 1992, 1997, and 2003, respectively. Currently, he holds a position as full professor for computer science and is head of the Computer Vision Group, Department of Mathematics and Computer Science, Friedrich Schiller University Jena, Germany. He is also Director of the Michael Stifel Center for Data-Driven and Simulation Science, Jena. His research interests comprise the automatic analysis, fusion, and understanding of sensor data, especially development of methods for visual recognition tasks and dynamic scene analysis. He contributed in the area of active vision, 3D reconstruction, as well as object recognition and tracking. He is author and co-author of over 300 journal and conference papers as well as technical articles. He is a member of IEEE, IEEE computer society, DAGM, and GI.