Generalized orderless pooling performs implicit salient matching

Marcel Simon[1], Yang Gao[2], Trevor Darrell[2], Joachim Denzler[1], Erik Rodner[3]
[1] Computer Vision Group, University of Jena, Germany [2] EECS, UC Berkeley, USA
[3] Corporate Research and Technology, Carl Zeiss AG
{marcel.simon, joachim.denzler}@uni-jena.de {yg, trevor}@eecs.berkeley.edu

Abstract

Most recent CNN architectures use average pooling as a final feature encoding step. In the field of fine-grained recognition, however, recent global representations like bilinear pooling offer improved performance. In this paper, we generalize average and bilinear pooling to "α-pooling", allowing for learning the pooling strategy during training. In addition, we present a novel way to visualize decisions made by these approaches. We identify parts of training images having the highest influence on the prediction of a given test image. This allows for justifying decisions to users and also for analyzing the influence of semantic parts. For example, we can show that the higher capacity VGG16 model focuses much more on the bird's head than, e.g., the lower-capacity VGG-M model when recognizing fine-grained bird categories. Both contributions allow us to analyze the difference when moving between average and bilinear pooling. In addition, experiments show that our generalized approach can outperform both across a variety of standard datasets.

1. Introduction

Deep architectures are characterized by interleaved convolution layers to compute intermediate features and pooling layers to aggregate information. Inspired by recent results in fine-grained recognition [19, 10] showing that certain pooling strategies offer equivalent performance as classic models involving explicit correspondence, we investigate here a new pooling layer generalization for deep neural networks suitable for both fine-grained and more generic recognition tasks.

Fine-grained recognition developed from a niche research field into a popular topic with numerous applications, ranging from automated monitoring of animal species [9] to fine-grained recognition of cloth types [8]. The defining property of fine-grained recognition is that all possible object categories share a similar object structure and hence similar object parts. Since the objects do not significantly differ in the overall shape, subtle differences in the appearance of an object part can likely make the difference between two classes. For example, one of the most popular fine-grained tasks is bird species recognition. All birds have the basic body structure with beak, head, throat, belly, wings as well as tail parts, and two species might dif-

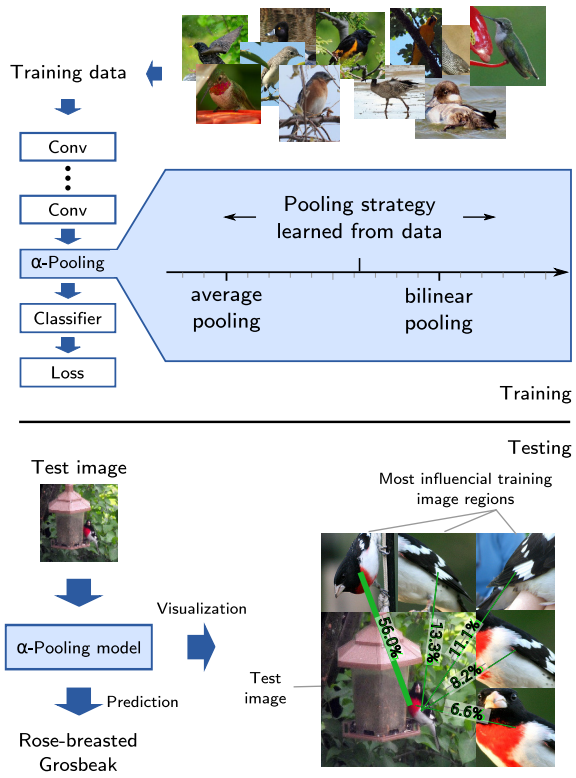


Figure 1. We present the novel pooling strategy α-pooling, which replaces the final average pooling or bilinear pooling layer in CNNs. It allows for a smooth combination of average and bilinear pooling techniques. The optimal pooling strategy can be learned during training to adapt to the properties of the task. In addition, we present a novel way to visualize predictions of α-pooling-based classification decisions. It allows in particular for analyzing incorrect classification decisions, which is an important addition to all widely used orderless pooling strategies.

fer only in the presence or absence of a yellow shade around the eyes.

Most approaches in the past five years concentrated on exploiting this extra knowledge about the shared general object structure. Usually, the objects are described by the appearance of different parts. This explicit modeling of the appearance of object parts is intuitive and natural. While explicit part modeling greatly outperformed off-the-shelf CNNs, the recently presented second-order or bilinear pooling [19] gives a similar performance boost at the expense of an understandable model.

Our paper presents a novel approach which has *both* state-of-the-art performance and allows for clear justification of the classification decision using visualization of influential training image regions. Classification accuracy on this task is reaching human level performance and hence we additionally focus on making classification decisions more understandable and explainable. We present an approach which can show for each evaluated image why the decision was made by referring to the most influential training image regions. Average pooling is mainly used in generic recognition. while bilinear pooling has its largest benefits in fine-grained recognition: our approach allows for understanding and generalizing the relationship between the two – a crucial step for further research.

Our *first contribution* is a novel generalization and parametric representation of the commonly used average and bilinear pooling. This representation allows for a smooth combination of these first-order and second-order operators. The framework provides both a novel conceptual understanding of the relationship of the methods and offers a new operating point with consistent improvement in terms of accuracy.

The *second contribution* is an analysis of the learned optimal pooling strategy during training. Our parametrized pooling scheme is differentiable and hence can be integrated into an end-to-end-learnable pipeline. We show that the learned pooling scheme is related to the classification task it is trained on. In addition, a pooling scheme half-way between average and bilinear pooling seems to achieve the highest accuracy on several benchmark datasets.

Our *third contribution* is a novel way to visually justify a classification decision of a specific image to a user. It is complementary to our novel pooling scheme and hence also applicable to the previous pooling schemes average and bilinear pooling. Both classifier parameters and local feature matches are considered to identify training image regions of highest influence.

Finally, our *fourth contribution* is an approach for quantifying the influence of semantic parts in a classification decision. In contrast to previous work, we consider both the classifier parameters and the saliency. We show that the CNN's way of classifying objects increasingly diverges from the human way, *i.e*. CNNs base most of their decisions on one object part instead of using a broad set of object attributes. In particular, we show that more complex CNN models like VGG16 focus much more on the bird's head compared to less complex ones like VGG-M. We also show that a similar shift can be seen when moving from average pooled features to bilinear features encoding.

After reviewing related work in the following section, Sect. 3 will present our novel α-pooling formulation, which generalizes average and bilinear pooling into a single framework. Sect. 4 will then investigate the relationship between generalized orderless pooling and pairwise matching and present an approach for visualizing a classification decision. This is followed by the experiments and a discussion about the trade-offs between implicit and explicit pose normalization for fine-grained recognition in Sect. 5 and 6.

2. Related work

Our work is related to several topics in the area of computer vision. This includes pooling techniques, match kernels, bilinear encoding, and visualizations for CNNs.

Pooling techniques and match kernels The presented α-pooling is related to other pooling techniques, which aggregate a set of local features into a single feature vector. Besides the commonly used average pooling, fully-connected layers, and maximum pooling, several new approaches have been developed in the last years. Zeiler *et al*. [32] randomly pick in each channel an element according to a multinomial distribution, which is defined by the activations themselves. Motivated by their success with handcrafted features, Fisher vector [12, 19] and VLAD encoding [11] applied on top of the last convolutional layer have been evaluated as well. The idea of spatial pyramids was used by He *et al*. [14] in order to improve recognition performance. In contrast to these techniques, feature encoding based on α-pooling shows a significantly higher accuracy in fine-grained applications. Lin *et al*. [19, 18] presents bilinear pooling, which is a special case of α-pooling. It has its largest benefits in fine-grained tasks. As shown in the experiments, learning the right mix of average and bilinear pooling improves results especially in tasks besides fine-grained.

The relationship of average pooling and pairwise matching of local features was presented by Bo *et al*. [2] as an efficient encoding for matching a set of local features. This formulation was also briefly discussed in [10] and used for deriving an explicit feature transformation which approximates bilinear pooling. Bilinear encoding was first mentioned by Tenenbaum *et al*. [28] and used, for example, by Carreira *et al*. [5] and Lin *et al*. [19] for image recognition tasks. Furthermore, the recent work of Murray *et al*. [21] also analyzes orderless pooling approaches and proposes a technique to normalize the contribution of each local de-

scriptor to resulting kernel values. In contrast, we show how the individual contributions can be used either for visualizing the classification decisions and for understanding the differences between generic and fine-grained tasks.

Justifying classifier predictions for an image Especially Sect. 4 is related to visualization techniques for information processing in CNNs. Most of the previous works focused on the primal view of the feature representation. This means they analyze the feature representations by looking only at a single image. Zeiler *et al*. [33] identify image patterns which cause high activations of selected channels of a convolutional layer. Yosinksi *et al*. [31] try to generate input patterns which lead to a maximal activation of certain units. Bach *et al*. [1] visualize areas important to the classification decision with layer-wise relevance propagation. In contrast to the majority of these works, we focus on the dual (or kernel) view of image classification. While a visualization for a single image looks interesting at the first sight, it does not allow for understanding which parts of an image are compared with which parts of the training images. In other words, these visualizations look only at the image itself and are omitting the relationship to the training data. For example, while the bird's head might be an attentive region in the visualization techniques mentioned above, a system might still compare this head with some unrelated areas in other images. Our approach allows for a clearer understanding about which pairs of training and test image regions contribute to a classification decision.

Zhang *et al*. [35] tackle a related idea for the case of explicit part detectors. They use the prediction score of a SVM classifier for each part to identify the most important patches for a selected part detector. We extend this idea to orderless-pooled features which do not originate from explicit part detections.

3. From generic to fine-grained classification: generalized α-pooling

Fine-grained applications like bird recognition and more generic image classification tasks like ImageNet have traditionally been two related but clearly separate fields with their own specialized approaches. While the general CNN architecture is shared, its usage differs. In this work, we focus on two state-of-the-art feature encoding: global average and bilinear pooling. While bilinear pooling shows the largest benefits in fine-grained applications, average pooling is the most commonly chosen final pooling step in literally all state-of-the-art CNN architectures. In this section, we show the connection between these two worlds. We present the novel generalization α-pooling, which allows for a continuous transition between average and bilinear pooling. The right mixture is learned with back-propagation from data in training, which allows for adapting to the specific tasks. In addition, the results will allow us to investigate which mixture of pooling approaches is best suited for which application, and what makes fine-grained recognition different from generic image classification.

Generalized α-pooling We propose a novel generalization of the common average and bilinear pooling as used in deep networks, which we call α-pooling. Let $(f, g, \mathcal{C})$ denote a classification model. $f : (\mathcal{I}, i) \mapsto \boldsymbol{y}_i \in \mathbb{R}^D$ denotes a local feature descriptor mapping from input image $\mathcal{I}$ and location i to a vector with length D, which describes this region. $g : \{\boldsymbol{y}_i \,|\, i = 1, \ldots, n\} \mapsto \boldsymbol{z} \in \mathbb{R}^M$ is a pooling scheme which aggregates n local features to a single global image description of length M. In our case, $M = D^2$ and is compressed using [10]. Finally, $\mathcal{C}$ is a classifier. In a common CNN like VGG16, f corresponds to the first part of a CNN up to the last convolutional layer, g are two fully connected layers and $\mathcal{C}$ is the final classifier.

An α-pooling-model is then defined by $(f, g^{\text{alpha}}, \mathcal{C})$, where

$$g^{\text{alpha}}(\{\boldsymbol{y}_i\}_{i=1}^n) = \text{v}\left(\frac{1}{n}\sum_{i=1}^{n} \text{alpha-prod}(\boldsymbol{y}_i, \alpha)\right) \qquad (1)$$

and

$$\text{alpha-prod}(\boldsymbol{y}_i, \alpha) = (\text{sgn}(\boldsymbol{y}_i) \circ |\boldsymbol{y}_i|^{\alpha-1})\boldsymbol{y}_i^T\,, \qquad (2)$$

where $\text{v}(\cdot)$ is the vectorization function, and $\text{sgn}(\cdot)$, $\cdot \circ \cdot$, $|\cdot|$, and $\cdot^\alpha$ denote the element-wise signum, product, absolute value and exponentiation function, and α is a model parameter. α has a significant influence on the pooling due to its role as an exponent. The optimal value is learned with back-propagation. For numerical stability, we add a small constant $\epsilon > 0$ to $|\boldsymbol{y}_i|$ when calculating the power and when calculating the logarithm. In our experiments, learning α was stable.

Special cases Average pooling is a common final feature encoding step in most state-of-the-art CNN architectures like ResNet [15] or Inception [27]. The combination [19] of CNN feature maps and bilinear pooling [28, 5] is one of the current state-of-the-art approaches in the fine-grained area. Both approaches are a special case of α-pooling.

For $\alpha = 1$ and $\boldsymbol{y}_i \geq 0$ we obtain $\text{alpha-prod}(\boldsymbol{y}_i, 1) = I \cdot \boldsymbol{y}_i^T$. Hence g^{alpha} calculates a matrix in which each row is the mean vector. This mean vector is identical to the one obtained in common average pooling. The vectorization $\text{v}(\cdot)$ turns the resulting matrix into a concatention of identical mean vectors.

In case of $\alpha = 2$ the mean outer product of $\boldsymbol{y}_i$ is calculated, which is equivalent to bilinear pooling: $\text{alpha-prod}(\boldsymbol{y}_i, 2) = \boldsymbol{y}_i\boldsymbol{y}_i^T$. Therefore, α-pooling allows for estimating the type of pooling necessary for a particular task by learning α directly from data.

α-pooling can continuously shift between average and bilinear pooling, which opens a great variety of opportunities. It shows a connection between both that was to the best

of our knowledge previously unknown. Furthermore, and even more important, all following contributions are also applicable to these two commonly used pooling techniques. They allow for analyzing and understanding differences between both special cases.

4. Understanding decisions of α-pooling

In this section, we give a "deep" insight into the class of α-pooled features, which includes average and bilinear pooling as well as shown in the last section. We make use of the formulation as pairwise matching of local features, which allows for visualizing both the gist of the representation and resulting classification decisions. We use the techniques presented in this section to analyze the effects of α-pooled features as we move between generic and fine-grained classification tasks. To simplify the notation, we will focus in this section on the case that all local features $\boldsymbol{y}$ are non-negative. This is the case for all features used in the experiments. All observations apply to the generic case in an analogous manner.

Interpreting decisions using most influential regions While an impressive classification accuracy can be achieved with orderless pooling, one of its main drawbacks is the difficulty of interpreting classification decisions. This applies especially to fine-grained tasks, since the difference between two categories might not be clear even for a human expert. Furthermore, there is a need to analyze false automatic predictions, to understand error cases and advance algorithms.

In this section, we use the formulation of α-pooling as pairwise matching to visualize classification decisions. It is based on finding locations with high influence on the decision. We show how to find the most relevant training image regions and show that even implicit part modeling approaches are well suited for visualizing decisions.

To show this, we calculate the linear kernel between the vectors $\boldsymbol{z}_k$ and $\tilde{\boldsymbol{z}}_\ell$, which induces a kernel between $\mathcal{Y}_k = \{\boldsymbol{y}_i\}_{i=1}^n$ and $\mathcal{Y}_\ell = \{\tilde{\boldsymbol{y}}_j\}_{j=1}^n$ as follows:

$$\begin{aligned}\langle \boldsymbol{z}_k, \tilde{\boldsymbol{z}}_\ell \rangle &\propto \langle \mathrm{v}\Big(\sum_{i=1}^n \boldsymbol{y}_i^{\alpha-1}\boldsymbol{y}_i^T\Big), \mathrm{v}\Big(\sum_{j=1}^n \tilde{\boldsymbol{y}}_j^{\alpha-1}\tilde{\boldsymbol{y}}_j^T\Big)\rangle \\ &= \mathrm{tr}\Big(\Big(\sum_{i=1}^n \boldsymbol{y}_i^{\alpha-1}\boldsymbol{y}_i^T\Big)^T\Big(\sum_{j=1}^n \tilde{\boldsymbol{y}}_j^{\alpha-1}\tilde{\boldsymbol{y}}_j^T\Big)\Big) \\ &= \sum_{i,j}\langle \boldsymbol{y}_i, \tilde{\boldsymbol{y}}_j\rangle \cdot \langle \boldsymbol{y}_i^{\alpha-1}, \tilde{\boldsymbol{y}}_j^{\alpha-1}\rangle, \end{aligned} \tag{3}$$

where we ignored normalizing with respect to n for brevity. Please note, that this derivation also reveals that the difference between bilinear and average pooling is only the quadratic transformation of the scalar product between two feature vectors $\boldsymbol{y}_i$ and $\tilde{\boldsymbol{y}}_j$.

If we use a single fully-connected layer after bilinear pooling and a suitable loss, the resulting score for a single class is given up to a constant as:

$$\sum_{k=1}^N \beta_k \langle \boldsymbol{z}_k, \tilde{\boldsymbol{z}}\rangle = \sum_{k=1}^N \sum_{i,j} \beta_k \cdot \langle \boldsymbol{y}_{i,k}, \tilde{\boldsymbol{y}}_j\rangle \langle \boldsymbol{y}_{i,k}^{\alpha-1}, \tilde{\boldsymbol{y}}_j^{\alpha-1}\rangle, \tag{4}$$

where β_k are the weights of each training image given by the dual representation of the last layer and N is the number of training samples. $\boldsymbol{z}_k$ is the α-pooled feature of the k-th training image and calculated using the local features $\boldsymbol{y}_{i,k}$.

A match between a region j in the test example and region i of a training example k is defined by the triplet (k, i, j). The influence of the triplet on the final score is given by the product

$$\gamma_{k,i,j} = \beta_k \cdot \langle \boldsymbol{y}_{i,k}, \tilde{\boldsymbol{y}}_j\rangle \langle \boldsymbol{y}_{i,k}^{\alpha-1}, \tilde{\boldsymbol{y}}_j^{\alpha-1}\rangle\,. \tag{5}$$

Therefore, we can visualize the regions with the highest influence on the classification decisions by showing the ones with the highest corresponding $\gamma_{k,i,j}$. This calculation can be done efficiently also on large datasets with the main limitation being the memory for storing the feature maps.

Figure 2 depicts a classification visualization for test images from four different datasets. In the bottom left of each block, the test image is shown. The test image is surrounded by the five most relevant training image regions. They are picked by first selecting the training images with the highest influence defined by the aggregated $\gamma_{k,i,j}$ over all locations i, j of the test and training image. In each training image, the highest $\gamma_{k,i,j}$ is shown using an arrow and a relative influence. The relative influence is defined by $\gamma_{k,i,j}$ normalized by the aggregated $\gamma_{k,i,j}$ over the test and all positive training image regions, *i.e.* images supporting the classification decision. Please note, that $\gamma_{k,i,j} >= 0$ for positive training samples as each element in $\boldsymbol{y}$ is greater or equal 0. Since multiple similar triplets occur, we use non-maximum suppression and group triplets with a small normalized distance of less than 0.15. As can be seen, this visualization of the classification decision is intuitive and reveals the high impact of a few small parts of the training images.

Measuring the contribution of semantic parts We are also interested whether human-defined semantic parts contribute significantly to decisions. Figure 3 shows the contribution of individual bird body parts for classification on CUB200-2011 [29]. For each test image, we obtain the ten most related training image similar to before. We divide the local feature into groups belonging to the bird's head, belly, and background and compute the sum of the squared inner products between these regions. As can be seen, on average, 25% of the VGG16 [26] prediction is caused by the comparison of the bird's heads. In contrast for VGG-M [7], the background plays the most significant role with a contribution of 31%. This shows that the deeper network VGG16 focuses much more on the bird instead of the background.

Relationship to salient regions We show that orderless pooling cannot just be rephrased as a correspondence ker-

Figure 2. Visualization of the most influential image regions for the classification decision as defined in Eq. 5: The large image in the bottom left corner is the test image and the surrounding images are crops of the training examples with highly relevant image regions. Percentages show the relative impact on the final decision. The lower four images show incorrect classifications.

nel [10] but also as implicitly performing salient matching. Eq. 3 calculates the linear kernel between the pooled features showing that it induces a kernel between each pair of local features. We can now show a further direct relation to a simple matching of local features in two images by rewriting the scalar products as:

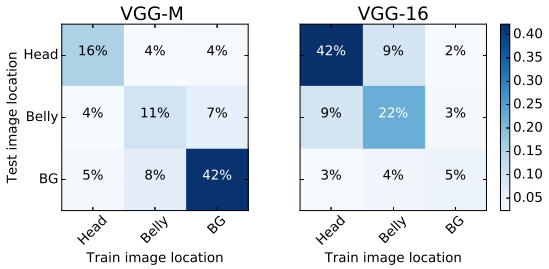


Figure 3. Contribution of different bird parts to the classification decision on CUB200-2011 comparing VGG-M and VGG16 without fine-tuning. For each semantic part in a given test image (rows), we compute the sum of inner products to another semantic part in a training image (columns). This statistic is normalized and averaged over all test images. The plots show that for VGG-M, 42% of the classification decision can be attributed to the comparison of background elements. In contrast, the comparison of the bird's head is most important for VGG16.

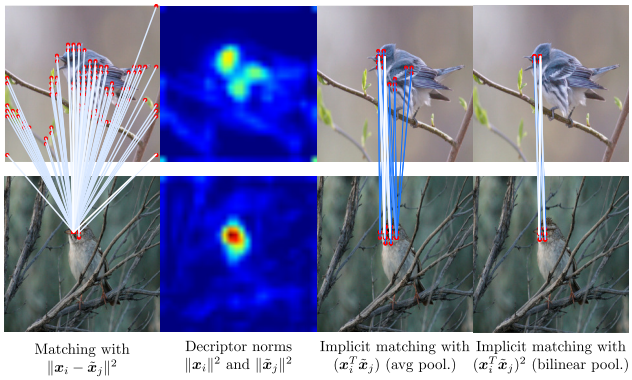


Figure 4. Visualization of the pairs of most similar local features using L2-distance. The thicker and whiter the line, the more similar are the features. Blue thin lines denote low similarity. We only show matchings larger than 50% of the maximum matching in this case.

$$\langle \boldsymbol{y}_i, \tilde{\boldsymbol{y}}_j \rangle \propto \sum_{i,j} \left(\|\boldsymbol{y}_i\|^2 + \|\boldsymbol{y}_j\|^2 - \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 \right) \quad , \qquad (6)$$

where the Euclidean distance between two features appears. The kernel output is therefore high if the feature vectors are highly similar (small Euclidean distance) especially for pairs (i, j) characterized by individual high Euclidean norms. In Figure 4, we visualize the Euclidean norms of the feature vectors in `conv5_3` extracted with VGG16. The input size was increased to 448×448 similar to [19, 10] and the output of `conv5_3` after activation and before pooling was used. Hence the local features have a spatial resolution of 28×28. In the first column, feature similarity was defined by the lowest L2-distance between local features. The second column shows the magnitude of all local features normalized to the highest norm in both feature maps. The third and fourth column show the implicit matchings using

the inner product and the squared inner product as similarity measure, as it is used in average and bilinear pooling. We only show matchings larger than 50% of the maximum matching in this case. As can be observed when comparing the plot for the norm and the matching, areas with a high magnitude of the features also correspond to salient regions. This is indeed reasonable since the "matching cost" in Eq. (3) should focus on the relevant object itself and not on background elements.

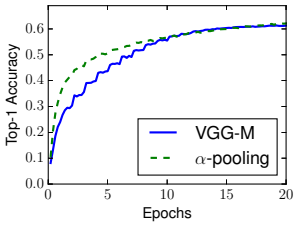
Focusing pairwise similarities by increasing α Similar to [19], we apply pooling directly after the ReLU activation following the last convolutional layer. Therefore, all scalar products between these features are positive. Hence, summands with a high scalar product are emphasized dramatically for large values of α and in particular also for bilinear pooling. Increasing α therefore leads to kernel values likely based on only a few relevant pairs (i, j). This fact is illustrated in the last two rows of Figure 4, where we only showed the pairs with an inner product larger 50% of the highest one for both average and bilinear pooling.

5. Experimental Analysis

In our experiments, we focus on analyzing the difference between average and bilinear pooling for image recognition. We make use of our novel α-pooling presented in Sect. 3. First, we show that it achieves state-of-the-art results in both generic and fine-grained image recognition. α and hence the pooling strategy is learned. Second, based on deep neural nets learned on both kinds of datasets, we analyze distinguishing properties using the visualization techniques of Sect. 4. We discuss the relationship of α with dataset granularity, classification decisions, and implicit pose normalization. Hence we manually set α in this second part.

Accuracy of α-pooling We evaluate both training from scratch and fine-tuning using a network pre-trained on the ILSVRC 2012 dataset [23]. For training from scratch, we use the VGG-M [7] architecture and replace the last pooling and the two fully-connected layers before the classifier with α-pooling. Batch normalization [16] is used after convolutions as well as after the α-pooling. In addition, we use dropout with a probability of $p = 0.5$ to reduce overfitting and improve generalization. Compact bilinear encoding [10] is used to reduce the dimensionality of the outer product to 8192. The learning rate starts at 0.025 and follows an exponential decay after every epoch. The batch size is 256. The results on ILSVRC 2012 (1000 classes, 1.2 million images) are shown in Figure 5. We plot both the validation accuracy during the first twenty epochs of the training as well as the final top-1 single crop accuracy. The network converges faster at only small additional computation cost and reaches a higher final accuracy compared to the original VGG-M with batch normalization.

For fine-tuning, we use VGG-M [7] and VGG16 [26] pre-trained on ILSVRC 2012. We replace the last pooling and the two fully connected layers before the classifier with the novel α-pooling encoding. We follow [19, 10] and add a signed square root as well as L_2-normalization layer before the classifier. Pooling is done across two scales with the smaller side of the image being 224 and 560 pixels long. Two-step fine-tuning [3] is used, where the last linear layer is trained first with 0.01 times the usual weight decay and the whole network is trained afterwards with the usual weight decay of 0.0005. The learning rate is fixed at 0.001 with a batch size of eight. α is learned from data.



Approach	Top-1 accuracy (Single crop)
VGG-M [7]	63.1%
Bilinear [18]	63.4%
Our work:	
VGG-M w/ α-pool	**64.2%**

Figure 5. Accuracy on ILSVRC 2012 for VGG-M. The left plot shows validation accuracy over the first twenty trained epochs. VGG-M denotes the original architecture and α-pooling the novel generalized pooling technique. α is learned from data.

Table 1. Accuracy on several datasets with α-pooling using the multi-scale variant. No ground-truth part or bounding box annotations were used. α is learned from data.

Dataset classes / images	CUB200-2011 200 / 12k	Aircraft 89 / 10k	40 actions 40 / 9.5k
Previous	81.0% [24] 82.0% [17] 84.5% [34]	72.5% [6] 78.0% [22] 80.7% [13]	72.0% [36] 80.9% [4] 81.7% [22]
Special case: bilinear [19]	84.1%	84.1%	-
Learned strategy (Ours)	**85.3%**	**85.5%**	**86.0%**

The results for CUB200-2011 birds [29], FGVC-Aircraft [20] and Stanford 40 actions [30] can be seen in Table 1. We achieve higher top-1 accuracy for all datasets compared to previous work. For fine-grained datasets like birds and aircraft, we slightly improve the results of [19, 10], which is due to $\alpha = 2$ being close to the learned α for this dataset. Our approach also shows a high accuracy on datasets besides traditional fine-grained tasks as shown by the actions dataset, where we achieve 86.0% accuracy compared to 81.7% reported in [22].

Ranking dataset granularity wrt. α As mentioned before, the main purpose of the experiments is to analyze the differences between average and bilinear pooling. In particular, we are interested in why average pooling lacks accuracy in fine-grained while bilinear reaches state-of-the-art.

The presented α-pooling allows for a smooth transition between average and bilinear pooling. In this paragraph, we analyze the relationship of the parameter α and the granularity of the dataset. The results using VGG16 can be seen

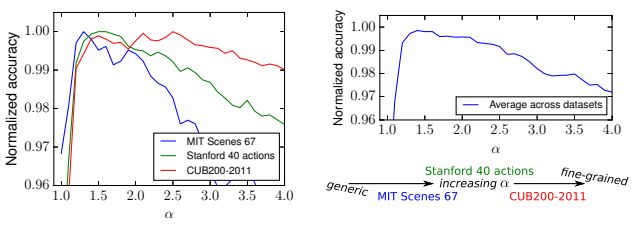


Figure 6. Influence of α using VGG16 without fine-tuning. $\alpha = 1$ corresponds to average pooling and $\alpha = 2$ to bilinear pooling. α is manually set in this experiment.

in Figure 6. α has been manually set to values in $[1.0, 4.0]$ and the accuracy without fine-tuning is plotted. The input resolution was increased to 448×448 as done in previous work [19, 10]. The accuracy is normalized to 1 for easier comparison of different datasets. It seems each dataset requires a different type of pooling. If the datasets are ordered by the value of α, which gives the highest validation accuracy, the order is as follows: MIT Scenes 67, 40 actions, and CUB200-2011 with $\alpha = 1.3$, 1.5, and 2.5, respectively. This seems to suggest that the more we move from generic to fine-grained classification, the higher is the value of α. In addition, larger values of alpha are still good for fine-grained while accuracy drops quickly for generic tasks. Hence focusing the classification on few correspondences seems a good strategy for fine-grained while it lowers accuracy on generic tasks. VGG-M shows a similar trend.

Classification visualization versus α Sect. 4 presents a novel way to visualize classification decisions for feature representations based on α-pooling. We are now interested in the change of classification decision reasoning with respect to α. Figure 7 shows the classification visualization for two sample test images from CUB200-2011 and MIT scenes 67. For each test image, we show the visualization for $\alpha = 1$ and $\alpha = 3$. While $\alpha = 1$ causes a fairly equal contribution of multiple training image regions to the decision, $\alpha = 3$ pushes the importance of the first images. For example, the contribution of the most relevant training image region grows from 11.2% to 23.2% in the bird image. A statistical analysis is shown in the supplementary material.

Relevance of semantic parts versus α A second way to analyze the pairwise matching induced by α-pooling is to quantify the matchings between semantic parts. We evaluated on CUB200-2011 using ground-truth part annotations. A ground-truth segmentation of the bird's head and belly was generated based on these annotations and used to assign feature locations in `conv5_3` of VGG16 to bird head, body, and background. Afterwards, for each test image, the kernel between all features of the bird's head in the test and a training image is aggregated. This is done for all pairs of regions and for the 10 most relevant training images. The obtained statistics are averaged over all test images.

Figure 7. Influence of α on the classification decisions. As in Figure 2, we visualize the most relevant image regions for the classification decision. A larger α increases the importance of the most influencial regions in the training images. Hence the decision is based on few important regions. α is manually set.

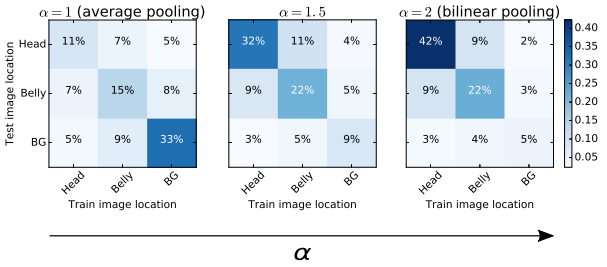


Figure 8. Influence of α on the contribution of different bird body parts to the classification decision on CUB200-2011. The higher the value of α, the higher is the influence of the actual bird body parts to the classification decision. α is manually set to $\{1, 1.5, 2\}$.

First, we analyze the influence of α on the contribution of different body parts. Figure 8 shows the results for VGG16 without fine-tuning when α is manually set to $\{1, 1.5, 2\}$. It seems that larger values of α focus the classification decision on actual body parts. The contribution of the bird's head to the classification decision shifts from 9% ($\alpha = 1$) to 42% ($\alpha = 2$). This observation matches our previous interpretation that larger values for α focus the classification decision on fewer discriminative pairs of local features.

Second, we are also interested in the effect of fine-tuning on classification decisions. Figure 9 depicts the results for VGG16 and α set to 2. Fine-tuning shifts the focus towards the bird's head, while especially the influence of background decreases. α-pooling is one of the few approaches which allow quantifying the influence of semantic parts.

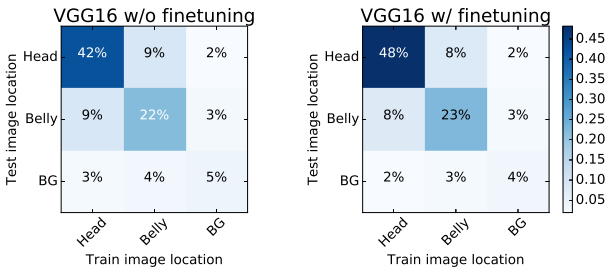


Figure 9. Influence of fine-tuning on the contribution of different bird body parts to the classification decision on CUB200-2011. As can be seen, the bird's head gains influence at the cost of background areas. α is set to 2.0.

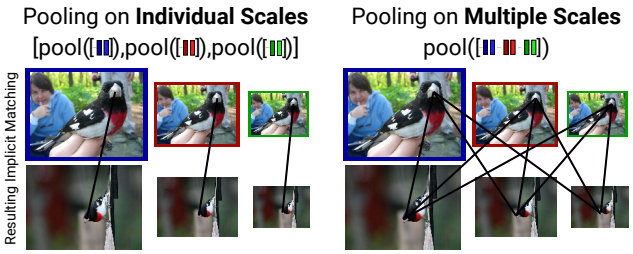


Figure 10. Illustration of different techniques to deal with multiple scales and their resulting implicit matching. Directly pooling over multiple scales allows for implicit matching across scales.

6. Discussion

Fine-grained tasks are about focusing on a few relevant areas Our in-depth analysis revealed that a high accuracy for fine-grained recognition can be achieved when only a few relevant areas are compared with each other by implicit salient matching. In terms of α-pooling, this corresponds to a higher value of the parameter α. It also explains why bilinear pooling showed a large performance gain for fine-grained tasks [19]: the corresponding $\alpha = 2$ increases the influence of highly related features. On the other hand, in generic image classification tasks like scene recognition, the general appearance seems more important and hence a lower value of α is better suited. Our experiments showed that $\alpha = 1.5$ is a good trade-off for a wide range of classification datasets and hence is a good starting point. If fine-tuning is used, α is learned and adapts to the best value.

Implicit matching vs. explicit pose normalization Most approaches for fine-grained recognition assumes objects consists of a few parts [24, 3, 25]. It is common belief that part-based in contrast to global descriptors allow for better representations of objects appearing in diverse poses.

In contrast, our analysis reveals that state-of-the-art global representations perform an implicit matching of several different image regions. Compared to explicit part-based models, they are not limited by a fixed number of parts learned from the data or utilized during classification. Our α-pooling strategy can even learn how much a classification decision should rely on a few rather than a large number of matchings. As argued in the last paragraph, the intuition that fine-grained recognition tasks are about "detecting a small set of image regions that matter" is right. However, the consequence that explicit part-based models are the solution is questionable. Rather than designing yet another part-based model, representations should be developed that lead to an even better implicit matching.

Kernel view of classification decisions We argue that the kernel view of classification decisions is a valuable tool for understanding and analyzing different feature encoding. We used the kernel view in the previous sections to show that a larger value of α focuses the classification decision on the most relevant pairs of local features. This understanding also allowed us to visualize classification decisions by using matchings to the most relevant training images. However, there are even more possible ways to exploit this in future work. For example, we can derive a feature matching over multiple scales in a theoretically sound way. Previous work often handled multiple scales by extracting crops at different scales and averaging the decision values across all crops [15, 27]. While this gives an improvement, a theoretical justification is missing. In contrast, if we perform α-pooling across all local features extracted from all scales of the input image, the kernel view reveals that this relates to a matching of local features across all possible combinations of locations and scales of two images, see Figure 10. To summarize, while kernel functions are rarely explicitly used in state-of-the-art approaches, they can be useful for both understanding and designing new approaches.

7. Conclusions

In this paper, we propose a novel generalization of average and bilinear pooling called α-pooling. Our approach has *both* state-of-the-art performance and a clear justification of predictions. It allows for a smooth transition between average and bilinear pooling, and to higher-order pooling, allowing for understanding the connection between these operating points. We find that in practice our method learns that an intermediate strategy between average and bilinear pooling offers the best performance on several fine-grained classification tasks. In addition, a novel way for visualizing classification decision is presented showing the most influential training image regions for a decision. Furthermore, we quantify the contributions of semantic parts in a classification decision based on these influential regions.

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] 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*, 10(7), 2015. 3

[2] L. Bo and C. Sminchisescu. Efficient match kernel between sets of features for visual recognition. In *NIPS*, pages 135–143. Curran Associates, Inc., 2009. 2

[3] S. Branson, G. V. Horn, S. J. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. *CoRR*, abs/1406.2952, 2014. 6, 8

[4] S. Cai, L. Zhang, W. Zuo, and X. Feng. A probabilistic collaborative representation based approach for pattern classification. In *CVPR*, pages 2950–2959, 2016. 6

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV (7)*, volume 7578 of *Lecture Notes in Computer Science*, pages 430–443. Springer, 2012. 2, 3

[6] Y. Chai, V. S. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, pages 321–328, 2013. 6

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*. BMVA Press, 2014. 4, 6

[8] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *ICCV Workshops*, pages 8–13, 2013. 1

[9] A. Freytag, E. Rodner, M. Simon, A. Loos, H. Khl, and J. Denzler. Chimpanzee faces in the wild: Log-euclidean cnns for predicting identities and attributes of primates. In *GCPR*, 2016. 1

[10] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. *CoRR*, abs/1511.06062, 2015. 1, 2, 3, 5, 6, 7

[11] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV (7)*, volume 8695 of *Lecture Notes in Computer Science*, pages 392–407. Springer, 2014. 2

[12] P. H. Gosselin, N. Murray, H. Jégou, and F. Perronnin. Revisiting the fisher vector for fine-grained classification. *Pattern Recognition Letters*, 49:92–98, 2014. 2

[13] P. H. Gosselin, N. Murray, H. Jégou, and F. Perronnin. Revisiting the fisher vector for fine-grained classification. *Pattern Recognition Letters*, 49:92–98, 2014. 6

[14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV (3)*, volume 8691 of *Lecture Notes in Computer Science*, pages 346–361. Springer, 2014. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3, 8

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. 6

[17] J. Krause, H. Jin, J. Yang, and F. Li. Fine-grained recognition without part annotations. In *CVPR*, pages 5546–5555, 2015. 6

[18] T. Lin and S. Maji. Visualizing and understanding deep texture representations. In *CVPR*, pages 2791–2799, 2016. 2, 6

[19] T. Lin, A. Roy Chowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, pages 1449–1457. IEEE Computer Society, 2015. 1, 2, 3, 5, 6, 7, 8

[20] S. Maji, E. Rahtu, J. Kannala, M. B. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. 6

[21] N. Murray, H. Jegou, F. Perronnin, and A. Zisserman. Interferences in match kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 2

[22] A. Rosenfeld and S. Ullman. Visual concept recognition and localization via iterative introspection. In *ACCV (5)*, pages 264–279, 2016. 6

[23] 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*, 115(3):211–252, 2015. 6

[24] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *ICCV*, pages 1143–1151. IEEE Computer Society, 2015. 6, 8

[25] M. Simon, E. Rodner, and J. Denzler. Part detector discovery in deep convolutional neural networks. In *ACCV (2)*, volume 9004 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 2014. 8

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4, 6

[27] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. 3, 8

[28] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000. 2, 3

[29] 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. 4, 6

[30] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and F. Li. Human action recognition by learning bases of action attributes and parts. In *ICCV*, pages 1331–1338, 2011. 6

[31] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, ICML*, 2015. 3

[32] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *CoRR*, abs/1301.3557, 2013. 2

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV (1)*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014. 3

[34] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian. Picking deep filter responses for fine-grained image recognition. In *CVPR*, June 2016. 6

[35] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M. N. Do. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Transactions on Image Processing*, 25(4):1713–1725, 2016. 3

[36] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 6