Generalized orderless pooling performs implicit salient matching

Marcel Simon\textsuperscript{1}, Yang Gao\textsuperscript{2}, Trevor Darrell\textsuperscript{2}, Joachim Denzler\textsuperscript{1}, Erik Rodner\textsuperscript{3}

\textsuperscript{1} Computer Vision Group, University of Jena, Germany \hspace{1em} \textsuperscript{2} EECS, UC Berkeley, USA \hspace{1em} \textsuperscript{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 \(\alpha\)-pooling\textsuperscript{,} 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. It 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 certain pooling strategies offered equivalent performance as classic models involving explicit correspondence, we investigate here a new pooling layer generalization for deep neural networks suitable both for 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-
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 pool-
ing [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 justifica-
tion of the classification prediction using visualization of in-
fuential 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 recog-
nition, while bilinear pooling has its largest benefits in fine-
grained recognition: our approach allows for understanding
the relationship between the two – a crucial
step for further research.

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

The second contribution is an analysis of the learned op-
timal 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 prediction 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 bi-
linear 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 quan-
tifying the influence of semantic parts in a classification de-
cision. In contrast to previous work, we consider both the
classifier parameters and the saliency. We show that the hu-
man way of classifying objects, i.e. using a broad set of ob-
ject attributes, increasingly diverges from the CNN’s way,
which bases most of its decision on one object part. 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 sim-
ilar shift can be seen when moving from average pooled
features to bilinear features encoding.

After reviewing related work in the following Section,
Section 3 will present our novel \( \alpha \)-pooling formulation,
which generalizes average and bilinear pooling into a single
framework. Section 4 will then investigate the relationship
between generalized orderless pooling and pairwise match-
"ing and present an approach for visualizing a classification
decision. This is followed by the experiments and a discus-
sion about the trade-offs between implicit and explicit pose
normalization for fine-grained recognition in Sec. 5 and 6.

2. Related work

Our work is related to several topics in the area of com-
puter vision. This includes pooling techniques, match ker-
nels, bilinear encoding, and visualizations for CNNs.

Pooling techniques and match kernels The presented
\( \alpha \)-pooling is related to other pooling techniques, which ag-
gregate a set of local features into a single feature vec-
tor. Besides the commonly used average pooling, fully-
connected layers, and maximum pooling, several new ap-
proaches have been developed in the last years. Zeiler et al.
[33] randomly pick in each channel an element accord-
ing to a multinomial distribution, which is defined by the ac-
tivations themselves. Motivated by their success with hand-
crafted features, Fisher vector [12, 19] and VLAD encod-
ing [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 per-
formance. In contrast to these techniques, feature encoding
based on \( \alpha \)-pooling show a significantly higher accuracy in
fine-grained applications. Lin et al. [19, 18] presents bilin-
ear pooling, which is a special case of average pooling. It
has its largest benefits in fine-grained tasks. As shown in
the experiments, learning the right mix of average and bi-
linear pooling improves results especially in tasks besides
fine-grained.

The relationship of average pooling and pairwise match-
"ing 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 approx-
imates bilinear pooling. Bilinear encoding was first men-
tioned by Tenenbaum et al. [29] 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-
Justifying classifier predictions for an image

Especially Section 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. [34] identify image patterns, which cause high activations of selected channels of a convolutional layer. Yosinski et al. [32] 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, we show the connection between these two worlds. 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 \( \alpha \)-pooling, which allows for a continuous transition between average and bilinear pooling.

### 3. From generic to fine-grained classification: \( \alpha \)-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 \( \alpha \)-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 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 \( \alpha \)-pooling

We propose a novel generalization of the common average and bilinear pooling as used in deep networks, which we call \( \alpha \)-pooling. Let \((f, g, C)\) denote a classification model. \(f : \mathcal{I} \times i \rightarrow y_i \in \mathbb{R}^D\) denotes a local feature descriptor mapping from input image \(\mathcal{I}\) and location \(i\) to a vector with length \(D\), which describes this region. \(g : \{y_i \mid i = 1, \ldots, n\} \rightarrow z \in \mathbb{R}^M\) 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, \(C\) is a classifier.

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

\[
g^{\alpha}(\{y_i\}_{i=1}^n) = v \left( \frac{1}{n} \sum_{i=1}^n \alpha \cdot \text{alpha-prod}(y_i, \alpha) \right) \tag{1}
\]

and

\[
\alpha \cdot \text{alpha-prod}(y_i, \alpha) = (\text{sgn}(y_i) \circ |y_i|^{\alpha-1}) y_i^T. \tag{2}
\]

where \(v(\cdot)\) is the vectorization function, and \(\text{sgn}(\cdot), \cdot \circ \cdot, |\cdot|, \text{and } \cdot^\alpha\) denote the element-wise signum, product, absolute value and exponentiation function, and \(\alpha\) is a model parameter. \(\alpha\) 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 \(|y_i|\) when calculating the power and when calculating the logarithm. In our experiments, learning \(\alpha\) 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 [28]. The combination [19] of CNN feature maps and bilinear pooling [29, 5] is one of the current state-of-the-art approaches in the fine-grained area. Both approaches are a special case of \( \alpha \)-pooling.

For \( \alpha = 1 \) and \( y \geq 0 \), i.e. \( \alpha \cdot \text{alpha-prod}(y, 1) = I \cdot y_i^T \), \( g^{\alpha} \) 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 \( v(\cdot) \) turns the resulting matrix into a concatenation of identical mean vectors.

In case of \( \alpha = 2 \), i.e. \( \alpha \cdot \text{alpha-prod}(y, 2) = y_i y_i^T \), the mean outer product of \( y_i \) is calculated, which is equivalent to bilinear pooling. Therefore, \( \alpha \)-pooling allows for estimating the type of pooling necessary for a particular task by learning \( \alpha \) directly from data.
α-pooling can continuously shift between average and bilinear pooling, which opens a great variety of possibilities. 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 \( y \) 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 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.

Let \( \tilde{z} \) be the α-pooling representation of a new test image resulting from local activations \( \tilde{y}_i \) of a convolutional layer. 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 by the representer theorem as:

\[
\sum_{k=1}^{N} \beta_k \langle z_k, \tilde{z} \rangle = \sum_{k=1}^{N} \sum_{i,j} \beta_k \cdot \langle y_{i,k}, \tilde{y}_j \rangle \langle y_{i,k}^{\alpha-1}, \tilde{y}_j^{\alpha-1} \rangle,
\]

(3)

where \( \beta_k \) are the weights of each training image given by the dual representation of the last layer and \( N \) is the number of training samples. \( z_k \) is the α-pooled feature of the \( k \)-th training image and calculated using the local features \( y_{i,k} \).

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

\[
\gamma_{k,i,j} = \beta_k \cdot \langle y_{k,i}, \tilde{y}_j \rangle \langle y_{k,i}^{\alpha-1}, \tilde{y}_j^{\alpha-1} \rangle.
\]

(4)

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 \( \gamma_{k,i,j} \geq 0 \) for positive training samples as each element in \( y \) 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 [30]. 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 [27] 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 kernel [10] but also as implicitly performing salient matching. To show this, we calculate the linear kernel between the vectors \( z_k \) and \( \tilde{z}_i \), which induces a kernel between \( \mathcal{Y}_k = \{y_i\}_{i=1}^n \) and \( \mathcal{Y}_i = \{y_j\}_{j=1}^m \) as follows:

\[
\mathcal{Y}_k \cdot \mathcal{Y}_i = \sum_{k=1}^{N} \sum_{i=1}^{n} \beta_k \cdot \langle y_{k,i}, \tilde{y}_j \rangle \langle y_{k,i}^{\alpha-1}, \tilde{y}_j^{\alpha-1} \rangle.
\]
Figure 2. Visualization of the most influential image regions for the classification decision as defined in Eq. 4: 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.

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.

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 $y_i$ and $\tilde{y}_j$.

We can now show a further direct relation to a simple matching of local features in two images by rewriting the scalar products as:

$$\langle y_i, \tilde{y}_j \rangle \propto \sum_{i,j} (\|y_i\|^2 + \|\tilde{y}_j\|^2 - \|y_i - \tilde{y}_j\|^2), \quad (6)$$

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. 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. (5) should focus on the relevant object itself and not on background elements.

**Focusing pairwise similarities by increasing $\alpha$** 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, summations with a high scalar product are emphasized dramatically for large values of $\alpha$ and in particular also for bilinear pooling. Increasing $\alpha$ 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 $\alpha$-pooling, which was presented in Section 3. First, we show that it achieves state-of-the-art results in both generic and fine-grained image recognition. In these experiments, $\alpha$ is learned from data and hence the pooling strategy is determined by the data. Second, based on deep neural nets learned on both kinds of datasets, we analyze distinguishing properties using the visualization techniques presented in Section 4. In detail, we discuss the relationship of the parameter $\alpha$ with dataset granularity, classification decisions, and implicit pose normalization. Hence we manually set the value of $\alpha$ in this second part.

**Accuracy of $\alpha$-pooling** We evaluated both training from scratch and fine-tuning using a network pre-trained on ImageNet Large Scale Visual Recognition 2012 challenge dataset [24]. For training from scratch, we used the VGG-M [7] architecture and replaced the last pooling and all fully-connected layers with $\alpha$-pooling. Batch normalization [16] was used after all convolutions as well as after the $\alpha$-pooling. We only show matchings larger than 50% of the maximum matching in this case. VGG16 with increased input size of $448 \times 448$ and the output of conv5-3 after activation and before pooling was used. The local features have a spatial resolution of $28 \times 28$.

![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. In the first row, feature similarity was defined by the lowest L2-distance between local features. The second row shows the magnitude of all local features normalized to the highest norm in both feature maps. The third and fourth row 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. VGG16 with increased input size of $448 \times 448$ and the output of conv5-3 after activation and before pooling was used. The local features have a spatial resolution of $28 \times 28$.](image)

| Approach          | Top-1 accuracy (Single crop) |
|-------------------|-------------------------------|
| VGG-M [7]         | 63.1%                         |
| Bilinear [18]     | 63.4%                         |
| **Our work**      | **64.2%**                     |

Figure 5. Accuracy on ImageNet using $\alpha$-pooling for VGG-M. The left plot shows validation accuracy over the first twenty trained epochs. VGG-M denotes the original architecture with batch normalization added and two fully-connected layers before the classifier. $\alpha$-pooling is the novel generalized pooling technique, which replaces these two fully-connected layers. With $\alpha$-pooling, the network converges faster and achieves a higher validation accuracy. $\alpha$ is learned from data.

For fine-tuning, we use a VGG-M [7] and VGG16 [27] pre-trained on ILSVRC 2012 [24] and replace all layers after the last convolutional layer with the novel $\alpha$-pooling encoding. We follow the authors of [19, 10] and add a signed square root as well as $L_2$-normalization layer before the last linear layer. Pooling is done across two scales with the smaller side of the input 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. $\alpha$ is learned from data.

The results for CUB200-2011 birds [30], FGVCAircraft [20] and Stanford 40 actions [31] can be seen in
Table 1. Accuracy on several datasets with $\alpha$-pooling using the multi-scale variant. No ground-truth part or bounding box annotations were used. $\alpha$ is learned from data.

| Dataset classes / images | CUB200-2011 | Aircraft 40 actions |
|--------------------------|-------------|---------------------|
| Previous                 | 81.0% [25]  | 72.5% [6] 72.0% [37] |
|                          | 82.0% [17]  | 78.0% [23] 80.9% [4]  |
|                          | 84.5% [35]  | 80.7% [15] 81.7% [23] |
| Special case: bilinear [19] | 84.1%      | 84.1%      |
| Learned strategy (Ours)  | 85.3%       | 85.5%       86.0%       |

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

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 $\alpha$ for this dataset. Our approach also shows a high accuracy on datasets besides traditional fine-grained tasks as shown by the Stanford 40 actions dataset, where we achieve 83.5% accuracy compared to the 81.7% reported in [23].

Ranking dataset granularity wrt. $\alpha$ As already mentioned, the one main purpose of these experiments is to analyze the differences between generic and fine-grained image classification tasks. In particular, we are interested in why average pooling lacks accuracy in fine-grained tasks while bilinear can reach state-of-the-art.

The presented $\alpha$-pooling allows for a smooth transition between average and bilinear pooling. In this paragraph, we analyze the relationship of the parameter $\alpha$ and the granularity of the dataset. The results using VGG16 without fine-tuning can be seen in Figure 6. The input resolution of both networks was increased to 448 x 448 to achieve state-of-the-art results without fine-tuning. For each dataset and value of $\alpha$, we train with a multinomial logistic loss. The accuracy is normalized to 1 for each dataset for easier comparison of different datasets. As can be seen, each dataset seems to require a different type of pooling. If the datasets are ordered by the value of $\alpha$, which gives the highest validation accuracy, the order is as follows: MIT Scenes 67 [22], 40 actions [31], and CUB200-2011 [30] 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 $\alpha$. 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 $\alpha$ Section 4 presented a novel way to visualize classification decisions for feature representations based on $\alpha$-pooling. We are now interested in the change of classification decision reasoning with respect to $\alpha$. 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 few images. For example, the contribution of the most relevant training image region grows from 11.2% to 23.2% in case of the bird image. A statistical analysis for all test images and datasets can be found in the supplementary material.

Relevance of semantic parts versus $\alpha$ A second way to analyze the pairwise matching induced by $\alpha$-pooling is to quantify the matchings between ground-truth parts. Similar to the previous paragraph, we are especially interested in why average pooling is not well suited for fine-grained classification on CUB200-2011.

We evaluated on CUB200-2011 using the 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.
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 $\alpha$-pooling, this corresponds to a higher value of the parameter $\alpha$. It also explains why bilinear pooling showed such a large performance gain for fine-grained recognition tasks [19]: the corresponding $\alpha = 2$ causes an increase of influence for 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 $\alpha$ 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, $\alpha$ will be learned and automatically adapted to the best value.

Implicit matching vs. explicit pose normalization The majority of approaches for fine-grained recognition assumes objects being decomposed into a few number of parts [25, 3, 26]. It is common belief that part-based feature descriptors in contrast to global descriptors allow for a better representation for 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 $\alpha$-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 $\alpha$ focuses the classification decision on only a few most relevant pairs of local features. This understanding also allowed us to visualize classification decision by using matchings to the most relevant training images. However, there are even more possible ways to exploit this formulation in future work. For example, we can derive a feature matching over multiple scales in a theoretically sound way. In previous work, multiple scales were often handled by extracting crops at different scales and averaging the decision values of the last layer across all crops [15, 28]. While this gives an improvement, a theo-
retical justification is missing. In contrast, if we perform \( \alpha \)-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 \( \alpha \)-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 predictions 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 hardware 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. 7
[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. Lemnitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In ICCV, pages 321–328, 2013. 7
[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. 7
[10] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. CoRR, abs/1511.06062, 2015. 1, 2, 3, 4, 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. 7
[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, 9
[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. 7
[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. Jégou, F. Perronnin, and A. Zisserman. Interferences in match kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016. 2
[22] A. Quattoni and A. Torralba. Recognizing indoor scenes. In CVPR, pages 413–420. IEEE Computer Society, 2009. 7
[23] A. Rosenfeld and S. Ullman. Visual concept recognition and localization via iterative introspection. In ACCV (5), pages 264–279, 2016. 7
[24] 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
[25] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In ICCV, pages 1143–1151. IEEE Computer Society, 2015. 7, 8
[26] 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
[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. 4, 6
[28] 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, 9
[29] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. Neural Computation, 12(6):1247–1283, 2000. 2, 3
[30] 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, 7
[31] 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, 7
[32] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In Deep Learning Workshop, ICML, 2015. 3
[33] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. CoRR, abs/1301.3557, 2013. 2
[34] 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
[35] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian. Picking deep filter responses for fine-grained image recognition. In CVPR, June 2016. 7
[36] 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
[37] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In CVPR, pages 2921–2929, 2016. 7