# SPLeaP: Soft Pooling of Learned Parts for Image Classification

Praveen Kulkarni[1,2](✉), Frédéric Jurie[2], Joaquin Zepeda[1],
Patrick Pérez[1], and Louis Chevallier[1]

[1] Technicolor, Cesson-Sévigné, France
{praveen.kulkarni,joaquin.zepeda,patrick.pérez,
louis.chevallier}@technicolor.com
[2] Normandie Univ, UNICAEN, ENSICAEN, CNRS, Caen, France
{praveen.kulkarni,frédéric.jurie}@unicaen.fr

**Abstract.** The aggregation of image statistics – the so-called pooling step of image classification algorithms – as well as the construction of part-based models, are two distinct and well-studied topics in the literature. The former aims at leveraging a whole set of local descriptors that an image can contain (through spatial pyramids or Fisher vectors for instance) while the latter argues that only a few of the regions an image contains are actually useful for its classification. This paper bridges the two worlds by proposing a new pooling framework based on the discovery of useful parts involved in the pooling of local representations. The key contribution lies in a model integrating a boosted non-linear part classifier as well as a parametric soft-max pooling component, both trained jointly with the image classifier. The experimental validation shows that the proposed model not only consistently surpasses standard pooling approaches but also improves over state-of-the-art part-based models, on several different and challenging classification tasks.

## 1 Introduction

This paper addresses the problem of image classification with Part-Based Models (PBMs). Decomposing images into salient parts and aggregating them to form discriminative representations is a central topic in the computer vision literature. It is raising several important questions such as: How to find discriminative features? How to detect them? How to organize them into a coherent model? How to model the variation in the appearance and spatial organization? Even if works such as the pictorial structure [1], the constellation model [2], object fragments [3], the Deformable Part Model [4] or the Discriminative Modes Seeking approach of [5] brought interesting contributions, as well as those in [6–8], the automatic discovery and usage of discriminative parts for image classification remains a difficult and open question.

Recent PBMs for image classification *e.g.*, [5–9] rely on five key components: (i) The generation of a large pool of candidate regions per image from (annotated) training data; (ii) The mining of the most discriminative and representative regions from the pool of candidate parts; (iii) The learning of part classifiers

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 329–345, 2016.
DOI: 10.1007/978-3-319-46484-8_20

using the mined parts; (iv) The definition of a part-based image model aggregating (independently) the learnt parts across a pool of candidate parts per image; (v) The learning of final image classifiers over part-based representations of training images.

One key challenge in the 2nd and 3rd components of PBMs lies in the selection of discriminative regions and the learning of interdependent part classifiers. For instance, one cannot learn the part classifiers before knowing discriminative regions and vice-versa. Extensive work has been done to alleviate the problem of identifying discriminative regions in a huge pool of candidate regions, *e.g.*, [5,7,8].

Once the discriminative regions are discovered and subsequently part classifiers are trained, the 4th component in a PBM – *i.e.*, the construction of the image model based on the per image part presence – is basically obtained by average or sum pooling of part classifier responses across the pool of candidate regions in the image. The final classifiers are then learnt on top of this part-based image representation. Although the aforementioned methods address one of the key components of PBMs, *i.e.*, mining discriminative regions by using some heuristics to improve final classification, they fail to leverage the advantage of jointly learning all the components together.

The joint learning approach of all components of PBMs is indeed particularly appealing since the discriminative regions are explicitly optimized for the targeted task. But intertwining all components makes the problem highly non-convex and initialization critical. The recent works of Lobel *et al.* [10] and Parizi *et al.* [9] showed that the joint learning of a PBM is possible. However, these approaches suffer from several limitations. First, their intermediate part classifiers are simple linear classifiers and the expression power of these part classifiers is limited in capturing complex patterns in regions. Furthermore, they pool the part classifier responses over candidate regions per image using max pooling which is suboptimal [11]. Finally, as the objective function is non-convex they rely on a strong initialization of the parts.

In the present work, we propose a novel framework, coined "Soft Pooling of Learned Parts" (SPLeaP), to jointly optimize all the five components of the proposed PBM. A first contribution is that we describe each part classifier as a linear combination of weak non-linear classifiers, learned greedily and resulting in a strong classifier which is non-linear. This greedy approach is inspired by [12,13] wherein they use gradient descent for choosing linear combinations of weak classifiers. The complexity of the part detector is increased along with the construction of the image model. This classifier is eventually able to better capture the complex patterns in regions. A second contribution is that we softly aggregate the computed part classifier responses over all the candidate regions per image. We introduce a parameter, referred as the "pooling parameter", for each part classifier independently inside the optimization process. The value of this pooling parameter determines the softness level of the aggregation done over all candidate regions, with higher softness levels approaching sum pooling and lower softness levels resembling max pooling. This permits to leverage different pooling regimes for different part classifiers. It also offers an interesting way to relax the assignment between regions and parts and lessens the need for strong initialization of the parts. The outputs of all part classifiers are fed to the final classifiers driven by the classifier loss objective.

The proposed PBM can be applied to various visual recognition problems, such as the classification of objects, scenes or actions in still images. In addition, our approach is agnostic to the low-level description of image regions and can easily benefit from the powerful features delivered by modern Convolutional Neural Nets (CNNs). By relying on such representations, and outperforming [14,15], the proposed approach can also be seen as a low-cost adaptation mechanism: pre-trained CNNs features are fed to a mid-to-high level model that is trained for a new target task. To validate this adaptation scheme we use the pre-trained CNNs of [15]. Note that this network is not fine-tuned on target datasets.

We validated our method on three challenging datasets: `Pascal-VOC-2007` (object), `MIT-Indoor-67` (scenes) and `Willow` (actions). We improve over state-of-the-art PBMs on the three of them.

The rest of the paper is organized as follows. The next section presents a review of the related works, followed by the presentation of the method in Sect. 3. Section 4 describes the algorithm proposed to jointly optimize the parameters, while Sect. 5 contains the experimental validation of our work.

## 2 Related Works

Most of the recent advances on image classification are concentrated on the development of novel Convolutional Neural Networks (CNNs), motivated by the excellent performance obtained by Krizhevsky *et al.* [16]. As CNNs require huge amount of training data (*e.g.*, `ImageNet`) and are expensive to train, some authors such as Razavian *et al.* [17] showed that the descriptors produced by CNNs *pre-trained* on a large dataset are generic enough to outperform many classification tasks on diverse small datasets, with reduced training cost. Oquaba *et al.* [14] and Chatfield *et al.* [15] were the first to leverage the benefit of fine-tuning the pre-trained CNNs to new datasets such as `Pascal-VOC-2007` [18]. Oquab *et al.* [14] reused the weights of initial layers of CNN pre-trained on `ImageNet`and added two new adaptation layers. They trained these two new layers using multi-scale overlapping regions from `Pascal-VOC-2007` training images, using the provided bounding box annotations. Chatfield *et al.* [15], on the other hand, fine-tuned the whole network to new datasets, which involved intensive computations due to the large number of network parameters to be estimated. They reported state-of-art performance on `Pascal-VOC-2007` till date by fine-tuning pre-trained CNN architecture.

In line with many other authors, [15,17] utilized the penultimate layer of CNNs to obtain global descriptors of images. However, it has been observed that computing and aggregating local descriptors on multiple regions described by pre-trained CNNs provides an even better image representation and improves classification performance. Methods such as Gong *et al.* [19], Kulkarni *et al.* [20] and Cimpoi *et al.* [21] relied on such aggregation using standard pooling techniques, *e.g.*, VLAD, Bag-of-Words and Fisher vectors respectively.

On the other hand, Part-Based Models (PBMs) proposed in the recent literature, *e.g.*, [5–8], can be seen as more powerful aggregators compared to [19,20,22]. PBMs attempt to select few relevant patterns or discriminative regions and focus on them in the aggregation, making the image representation more robust to occlusions or to frequent non-discriminative background regions.

PBMs differ in the way they discover discriminative parts and combine them into a unique description of the image. The Deformable Part Model proposed by Felzenszwalb *et al.* [4] solves the aforementioned problems by selecting discriminative regions that have significant overlap with the bounding box location. The association between regions and part is done through the estimation of latent variables, *i.e.*, the positions of the regions w.r.t. the position of the root part of the model. Differently, Singh *et al.* [6] aimed at discovering a set of relevant patches by considering the representative and frequent enough patches which are, in addition, discriminative w.r.t. the rest of the visual world. The problem is formulated as an unsupervised discriminative clustering problem on a huge dataset of image patches, optimized by an iterative procedure alternating between clustering and training discriminative classifiers. More recently, Juneja *et al.* [7] also aimed at discovering distinctive parts for an object or scene class by first identifying the likely discriminative regions by low-level segmentation cues, and then, in a second time, learning part classifiers on top of these regions. The two steps are alternated iteratively until a convergence criterion based on Entropy-Rank is satisfied. Doersch *et al.* [5] used density based mean-shift algorithms to discover discriminative regions. Starting from a weakly-labeled image collection, coherent patch clusters that are maximally discriminative with respect to the labels are produced, requiring a single pass through the data.

Contrasting with previous approaches, Li *et al.* [23] were among the first to rely on CNN activations as region descriptors. Their approach discovers the discriminative regions using association rule mining techniques, well-known in the data mining community. Sicre *et al.* [24] also build on CNN-encoded regions, introducing an algorithm that models image categories as collections of automatically discovered distinctive parts. Parts are matched across images while learning their visual model and are finally pooled to provide images signatures.

One common characteristic of the aforementioned approaches is that they discover the discriminative parts first and then combine them into a model of the classes to recognize. There is therefore no guaranty that the so-learned parts are optimal for the classification task. Lobel *et al.* [10] showed that the joint learning of part and category models was possible. More recently, Parizi *et al.* [9] build on the same idea, using max pooling and $l_1/l_2$ regularization.

Variour authors have likewise studied learned soft-pooling mechanisms. Gulcehre *et al.* [25] investigate the effect of using generalized soft pooling as a non-linear activation unit, bearing some similarity with the maxout non-linear unit of [26]. In contrast, our method uses a generalized soft pooling strategy as a down sampling layer. Our method is close to that of Lee *et al.* [27], who use linear interpolation of max and average pooling. Our approach, on the other hand, uses a non-linear interpolation of these two extrema.

## 3 Proposed Approach

Our goal is to represent each category involved in the visual recognition problem of interest as a collection of discriminative regions. These regions are automatically identified using learned part classifiers, that will operate on a pool of proposed fragments. A "part" classifier is meant to capture specific visual patterns. As such it does not necessarily capture a strong, human understandable semantic: it might respond highly on more than one region of the given image or, conversely, embrace at once several identifiable parts of the object. On images from "horse" class for instance, one part classifier might focus on the head of the animal when another one turns out to capture a large portion of the horse body.

Formally, we consider an image as a bag of $R$ regions, each one equipped with a descriptor $\mathbf{x}_r \in \mathbb{R}^D$. The image is thus represented at first by the descriptor collection $\mathcal{X} = \{\mathbf{x}_r\}_{r=1}^R$. The number of regions will be image-dependent in general even if we assume it is not for notational convenience.

Based on training images spanning $C$ images categories, $P$ "part" classifiers will be learned, each as a weighted sum of $K$ base classifiers applied to a region's descriptor ($K$ chosen by cross-validation). The score of the $p$-th part classifier for a given descriptor $\mathbf{x}$ is defined as:

$$H_p(\mathbf{x}; \boldsymbol{\theta}_p) = \sum_{k=1}^{K} a_k^p \sigma(\mathbf{x}^\top \mathbf{u}_k^p + b_k^p), \tag{1}$$

where $\sigma$ is the sigmoid function, $a_k^p$ is the weight of the $k$-th base classifier, $\mathbf{u}_k^p \in \mathbb{R}^D$ and $b_k^p \in \mathbb{R}$ are its parameters and $\boldsymbol{\theta}_p = \mathrm{vec}(a_{1:K}^p, \mathbf{u}_{1:K}^p, b_{1:K}^p) \in \mathbb{R}^{K(D+2)}$ is the vector of all the parameters that define the part classifier. This score is aggregated over the pool of $R$ regions a follows:

$$f_p(\mathcal{X}) = \sum_{r=1}^{R} \pi_r^p H_p(\mathbf{x}_r; \boldsymbol{\theta}_p), \tag{2}$$

where normalized weights are defined as

$$\pi_r^p \propto \exp\big(\beta_p H_p(\mathbf{x}_r; \boldsymbol{\theta}_p)\big), \ \sum_{r=1}^{R} \pi_r^p = 1, \tag{3}$$

with $\beta_p$ a part-dependent "pooling" parameter. For large values of this parameter the scores are max-pooled, while they are averaged for very small values.

Given a set of part classifiers with parameter $\varTheta = [\boldsymbol{\theta}_1|\cdots|\boldsymbol{\theta}_P]$ and associated pooling parameters $\boldsymbol{\beta} = [\beta_p]_{p=1}^{P}$, the bag of $R$ region descriptors $\mathcal{X} = \{\mathbf{x}_r\}_r$ attached to an input image is turned into a part-based description:

$$\mathbf{f}(\mathcal{X};\varTheta,\boldsymbol{\beta}) = [f_p(\mathcal{X})]_{p=1}^{P}. \tag{4}$$

The multiclass classification problem at hand is cast on this representation. Resorting to logistic regression, we aim at learning $P$-dimensional vectors, $\mathbf{w}_c = [w_1^c \cdots w_P^c]^\top \in \mathbb{R}^P$, one per class, so that the class label $y \in \{1 \cdots C\}$ of an input image $\mathcal{X}$ is predicted according to distribution

$$\Pr(y = c|\mathcal{X};\varTheta,\boldsymbol{\beta},W) = \frac{\exp\left(\mathbf{w}_c^\top \mathbf{f}(\mathcal{X};\varTheta,\beta)\right)}{\sum_{d=1}^{C}\exp\left(\mathbf{w}_d^\top \mathbf{f}(\mathcal{X};\varTheta,\beta)\right)}, \tag{5}$$

where $W = [\mathbf{w}_1|\cdots|\mathbf{w}_C]$. For simplicity in notation, we have omitted the bias term associated with each class. In practice, we append each of them to the corresponding vectors $\mathbf{w}_c$s and entry one is appended to descriptor $\mathbf{f}(\mathcal{X};\varTheta,\beta)$.

Discriminative learning is conducted on annotated training dataset $\mathcal{T} = \{(\mathcal{X}_n, y_n)\}_{n=1}^{N}$, with $\mathcal{X}_n = \{\mathbf{x}_r^n\}_{r=1}^{R}$ and $y_n \in \{1,\ldots,C\}$. Part-level and category-level classifiers are jointly learned by minimizing a regularized multiclass cross entropy loss:

$$\min_{\varTheta,\boldsymbol{\beta},W} -\sum_{n=1}^{N}\sum_{c=1}^{C}[y_n = c]\ln\Pr(c|\mathcal{X}_n;\varTheta,\boldsymbol{\beta},W) + \mu\|\varTheta\|_F^2 + \delta\|W\|_F^2, \tag{6}$$

where [.] is Iverson bracket. The two regularization weights $\mu$ and $\delta$, the number $P$ of part classifiers and the number $K$ of base learners in each part are set by cross-validation. Learning is done by block-wise stochastic gradient descent, as explained next into more details.

The multi-class loss in (6) being based on softmax (5), it requires that each image in the training set is assigned to a single class. If this is not the case, one can use instead a one-vs.-all binary classification approach, which can be easily combined as well with the proposed PBM.

## 4 Optimization Specific Details

In this section we provide details on how the joint optimization problem (6) is addressed. It aims at learning the final category-level classifiers (defined $W$), the part classifiers (defined by $\varTheta$) and the part-dependent pooling coefficients in $\boldsymbol{\beta}$. By conducting jointly these learnings, part classifiers are optimized for the target recognition task. Additionally, learning part-specific parameter $\beta_p$ enables to accommodate better the specifics of each part by adapting the softness of its region pooling.

Algorithm 1 summarizes the different steps of the optimization. In Algorithm 1, we denote $\boldsymbol{\theta}_{(k)}$ the vector of parameters associated to $k$-th base classifiers in matrix $\Theta$, that is $\boldsymbol{\theta}_{(k)} = \mathrm{vec}(a_k^{1:P}, \mathbf{u}_k^{1:P}, b_k^{1:P})$ and $\mathcal{L}_n = \log \Pr(y_n|\mathcal{X}_n; \Theta, \boldsymbol{\beta}, W)$ the log-likelihood of $n$-th training pair $(\mathcal{X}_n, y_n)$.

We perform $E$ epochs of block-coordinate stochastic gradient descent. If part-related parameters $\Theta$ and $\boldsymbol{\beta}$ were known and fixed, the optimization of image classifiers $W$ alone in the proposed algorithm would amount to the classic learning of logistic regressors on image descriptors $\mathbf{f}(\mathcal{X})$ defined in (4). The interleaved learning of the $P$ part-classifiers defined by $\Theta$ is more involved. It relies on a stage-wise strategy whereby base classifiers are progressively incorporated. More precisely, we start with a single weak classifier per part, randomly initialized and optimized over the first $S$ epochs. Past this first stage with training a single weak classifier, each part-classifier is then allowed an additional weak classifier per epoch. With initialization to zero of the parameters of this new learner, non-zero gradients for these parameters is produced by training samples that were previously misclassified. Note that at each epoch, only the last weak classifier is updated for each part while previous ones are kept fixed.

We set all algorithm's parameters (number $P$ of parts, number $K$ of weak classifiers per part, number $S$ of epochs with part classifiers based only on a single weak learner, learning rates $\gamma_W$, $\gamma_\theta$ and $\gamma_\beta$) through careful cross-validation.

**Algorithm 1.** SPLeaP Training: joint part-category classifier learning

```
procedure LEARN($\mathcal{T}$)
    parameters: $P$, $K$, $\mu$, $\delta$, $S$, $\gamma_W$, $\gamma_\theta$, $\gamma_\beta$
    $W \leftarrow 0$
    $\boldsymbol{\theta}_{(1)} \leftarrow$ rand()
    $\boldsymbol{\theta}_{(2:K)} \leftarrow 0$
    $\boldsymbol{\beta} \leftarrow$ rand()
    $k \leftarrow 1$
    for $e = 1$ to $E = K + S - 1$ do
        $\mathcal{T} \leftarrow \mathit{RandomShuffle}(\mathcal{T})$
        for $n = 1$ to $N$ do
            $W \leftarrow (1 - \gamma_W)W + \gamma_W \sum_{(\mathcal{X}_n, y_n) \in \mathcal{T}} \nabla_W \mathcal{L}_n$
            $\boldsymbol{\theta}_{(k)} \leftarrow (1 - \gamma_\theta)\boldsymbol{\theta}_{(k)} + \gamma_\theta \sum_{(\mathcal{X}_n, y_n) \in \mathcal{T}} \nabla_{\boldsymbol{\theta}_{(k)}} \mathcal{L}_n$
            $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \gamma_\beta \sum_{(\mathcal{X}_n, y_n) \in \mathcal{T}} \nabla_{\boldsymbol{\beta}} \mathcal{L}_n$
        end for
        if $e > S$ then
            $k \leftarrow k + 1$
        end if
    end for
    Return $W, \Theta, \boldsymbol{\beta}$
end procedure
```

# 5 Results

## 5.1 Experimental Settings

**Datasets.** We evaluate our proposed method using three well-known datasets described below:

*Pascal-VOC-2007.* The Pascal-VOC-2007 dataset [18] is an image classification dataset consisting of 9963 images of 20 different visual object classes divided into 5011 training images (of which 2510 are validation images) and 4952 testing images. The images contain natural scenes and the visual object classes span a wide range, including animals (*e.g.*, dog, cat), vehicles (*e.g.*, aeroplane, car) and other manufactured objects (*e.g.*, tv monitor, chair).

*MIT Indoor 67.* As opposed to objects, scenes are non-localized visual concepts and might even be characterized by the presence or absence of several objects. The MIT-Indoor-67[28] dataset is a scene recognition dataset consisting of 67 indoor scenes (*e.g.*, nursery, movie theater, casino or meeting room) each represented by 80 training images and 20 test images. We use 20 randomly chosen training images from each class as a validation set.

*Willow Dataset.* Recognizing human action in photos is a challenging task due the absence of temporal information. Dedicated to this task, the Willow dataset [29] consists of 7 action categories such as “play instrument”, “walk” or “ride horse” spread across 490 training images (of which 210 are validation images) and 421 test images.

**Region Proposal Schemes.** We explored three different strategies to extract the pool of region proposals from each image:

*Selective Search (SS).* We use the selective search region proposal scheme of [30] to extract between 100 and 5000 region proposals per image, with an average of 800, using Matlab code provided by [31].

*Augmentation (aug).* Following the data augmentation technique of [15], we derive ten images from each input image by taking one center crop and four corner crops from the original image and further mirroring each crop vertically. The ten resulting modified image crops are used as region proposals.

*Selective search+augmentation (SS+aug).* We also explore merging the outputs of the two previous strategies into a single pool of region proposals.

**Region Feature Extraction.** From each of the candidate regions obtained using one of the above described region proposal methods, we extract one feature vector consisting of the activation coefficients of the previous-to-last layer of several state-of-the-art CNN architectures. The CNN architectures we consider, available in CAFFE [32], are *(i)* the 128-dimensional feature extracted from the 13-layer architecture of [15] (VGG-128), *(ii)* the 16-layer architecture of [33] producing 4096 dimensional features (VD-16) and *(iii)* the architecture of [34] corresponding to Krizhevsky’s architecture [16] pre-trained using ImageNet (978 categories) and the Places database (HybridCNN).

**Cross-Validation of Hyper-parameters.** We use Stochastic Gradient Descent (SGD) to train our model. The performance of the model depends on the value of the various hyper-parameters: the number of parts $P$ and of weak learners in each part classifier $K$, the regularization weights $\mu$ and $\delta$ in (6), the number of epochs $E$ and the various learning rates (see Algorithm 1). For the `Pascal-VOC-2007` and `Willow` datasets, we use piecewise-constant learning rates decreased every ten epochs empirically, similarly to the appraoch of [15]. For the `MIT-Indoor-67`dataset, we use learning rates of the form $\gamma(i) = \frac{\gamma_0}{1.0+\lambda i}$, where $\gamma_0$ and $\lambda$ are hyper parameters that are cross-validated.

We select the values of these hyper-parameters using cross-validation. After the cross-validation phase, the hyper-parameters are set accordingly and the training and validation data are merged to re-train our model.

### 5.2 Experimental Validation of the Contributions

We now establish experimentally the benefits of our main contributions: weakly supervised parts learning, soft-max pooling with learned, per-part softness coefficients, and part detectors based on weak learners. To this end, we use the `Pascal-VOC-2007` dataset along with the mean Average Precision (mAP) performance measure specified by the dataset's authors, using VGG-128 features to represent all region proposals.

**Comparison with Unsupervised Aggregation.** In Table 1, we first verify that the improvements of our method are not due to simply the region proposal strategies we employ. We hence compare our supervised SPLeaPmethod to three analogous baseline features not employing supervised learning. The first baseline, denoted *VGG-128-G*, uses the global feature vector extracted from the whole image. The second baseline, denoted *VGG-128-sum*, aggregates VGG-128 features extracted from each candidate region using average pooling, similarly to an approach used in [35]. Both of these baselines result in 128-dimensional feature vectors. In a third baseline, denoted *VGG-128-K-means*, we perform $K$-means on all candidate regions from all images in the database to obtain $P = 40$ centroids. Computing an image feature then consists of selecting the image's $P \ll R$ candidate region whose features are closest to the $P$ centroids and concatenating them into a single vector of size $128P$.

For each of the aforementioned feature construction methods, the resulting image feature vectors are $\ell_2$-normalized and then used to learn linear SVMs using a one-vs.-rest strategy.

The results in Table 1 establish that large performance gains (more than 8 mAP points) are obtained by proposed SPLeaPmethod relative to the different baseline aggregation strategies, and hence the gain does not follow simply from using our region proposal strategies. Interestingly, contrary to the baseline strategies, our method succeeds in exploiting the merged *SS+aug* region proposal strategy (0.47 mAP improvement relative to *SS*).

**Table 1.** Comparison against unsupervised aggregation baselines

| VGG-128-G | VGG-128-sum | | | VGG-128-$K$-means | | SPLeaP | |
|---|---|---|---|---|---|---|---|
| | SS | aug | SS+aug | SS | SS+aug | SS | SS+aug |
| 75.32 | 77.31 | 78.21 | 77.36 | 76.28 | 76.8 | 84.21 | **84.68** |

**Table 2.** Importance of per-part softness coefficients

| Average pooling | Max pooling | Cross-valid. $\beta_p = \beta$ | Learned $\beta_p$ |
|---|---|---|---|
| 80.77 | 83.23 | 84.31 | **84.68** |

**Importance of Per-Part Softness Coefficient.** In Table 2, we evaluate our proposed soft-max pooling strategy in (3) that employs a learned, per-part softness coefficient $\beta_p$. We compare per-part softness coefficients to three alternatives: *(i)* average pooling, wherein $\forall p, \beta_p = 0$; *(ii)* max pooling, which is equivalent to $\forall p, \beta_p \rightarrow \infty$; and *(iii)* a cross-validated softness coefficient that is constant for all parts, $\forall p, \beta_p = \beta$. In all three of these alternatives, we run the complete SPLeaPoptimization process discussed in Sect. 4. As illustrated in the table, using our proposed learned, per-part softness coefficient yields the best performance, with an improvement of close to 4 mAP points over average pooling, 1.5 mAP points over max pooling, and 0.4 mAP points over a cross-validated but constant softness coefficient. Note that allowing the algorithm to choose $\beta_p$ during the optimization process eliminates the need for a costly cross validation of the $\beta_p$.

**Effect of Number of Weak Learners *K*.** In Fig. 1 we evaluate the effect of the number $K$ of weak learners per part by plotting mAP as a function of the number of training epochs. Note that, for a fixed number of learning iterations, adding more weak learners results in higher performance. We have tried the effect of other design choices such as averaging $K$ weak learners in contrast to greedily adding the weak learners. We obtain slight improvement *i.e.* we obtain 84.78 mAP for $K = 3$. We also compared adding weak learners to dropout, which is known to behave as averaging multiple thinned networks, and obtained a reduction in mAP of 0.5 % (83.98 mAP with 50 % dropout).

### 5.3 Parameters/Design Related Choices

**Per-Category Parts and Number *P* of Parts.** When learning SPLeaPfor the `MIT-Indoor-67`dataset, we learn $P$ part classifiers that are common to all 67 categories using the multi-class objective described in Sect. 3. For `Willow` and `Pascal-VOC-2007`, on the other hand, we learn $P$ different part classifiers for each category, using a one-vs.-rest strategy to learn each SPLeaPmodel independently for each class.

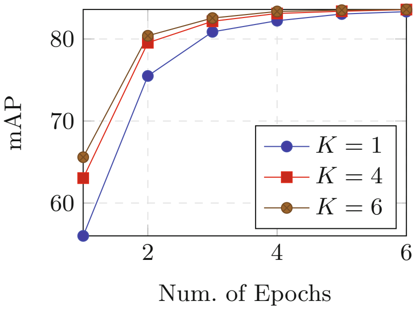

**Fig. 1.** Plot of test mAP versus number of training epochs.

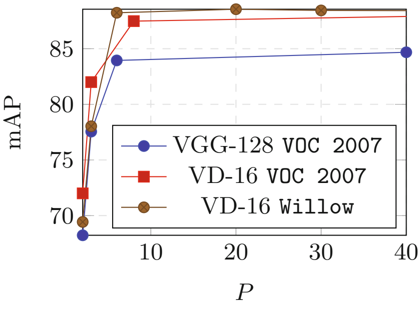

**Fig. 2.** Plot of test mAP versus the number of parts $P$.

In Fig. 2 we evaluate mAP on `Pascal-VOC-2007` as a function of the number of parts $P$. We show that even with a small number of parts $P = 6$ per class, we obtain a very good mAP of 83.94.

### 5.4 Comparisons with State-of-the-art

**Pascal-VOC-2007.** In Table 3 we compare SPLeaPto various existing state-of-the-art methods on the `Pascal-VOC-2007` dataset.

*Methods Employing Krizhevsky-Type Architectures.* On the left side of Table 3, we compare against Krizhevsky's original 13-layer architecture [16] and variants thereof such as VGG-128 [15]. In particular, the architectures of [14, 15] were first learned on `ImageNet` and subsequently fine-tuned specifically for `Pascal-VOC-2007`.

Note that, when using architectures derived from [16], including architectures fine-tuned specifically for `Pascal-VOC-2007`, our method outperforms all of these baselines by at least 3 absolute mAP points, despite using the 128-dimensional

**Table 3.** Comparison of results on `Pascal-VOC-2007` dataset ($P = 40$ parts per class, $K = 1$) using CNN features extracted from (*left*) Krizhevsky-like [16] and (*right*) very deep architectures [33]

| Methods | $mAP$ |
|---|---|
| VGG-G | 75.35 |
| Oquab *et al.* [14] | 77.31 |
| Li *et al.* [23] | 77.90 |
| Cimpoi *et al.* [21] | 79.50 |
| CNN-S fine tuned [15] | 82.42 |
| SPP [36] | 82.44 |
| SPLeaP-VGG-128 (*SS+ext. aug*) | **84.68** |

| Methods | $mAP$ |
|---|---|
| VD-16-G [33] | 81.73 |
| VD-16 (*dense_evaluation*) | 84.67 |
| VD-16-sum (*SS+ext. aug*) | 82.58 |
| Cimpoi *et al.* [21] | 85.10 |
| SPLeaP-VD-16 (*SS+ext. aug*) | **88.01** |

VGG-128 feature that is not fine-tuned for Pascal-VOC-2007. In particular, our method outperforms the recent, part-based representation of [23], which is a state-of-the-art part-based method employing association rule mining to discover discriminative patterns/regions. In Table 3 we present their results based on features from [16].

*Methods Employing Very Deep Architectures.* On the right side of Table 3, we compare against the deep pipelines of Simonyan *et al.* [33], using the pre-computed models provided by the authors in [32] to reproduce the baselines. We use the state-of-the-art VD-16 feature to reproduce three different baselines using our own implementations.

The first one (VD-16-G) uses a global VD-16 feature by feeding the entire image to the CNN architecture.

The second one, VD-16 *dense_evaluation*, follows [33] in employing their CNN architecture as a fully convolutional architecture by treating the weights of the last two fully connected layers as 7 × 7 and 1 × 1 convolutional kernels, respectively. This enables them to process images of arbitrary size. The approach further employs scaling, cropping and flipping to effectively produce a pool of close to 500 region proposals that are subsequently average-pooled. The resulting descriptor is $\ell_2$ normalized and used to compute linear SVMs, and achieves state-of-the-art results on Pascal-VOC-2007.[1]

For completeness, we further explore a third baseline that employs the extended augmentation (*ext. aug.*) strategy employed by [37], which effectively produces 144 crops per image, as opposed to the 10 crops of the *aug* strategy discussed above. We further extend this region proposals by the selective search region proposals and employ sum pooling.

The results, summarized in Table 3, show that proposed SPLeaPsystem outperforms all three baselines, and further outperforms a very recent baseline [21] relying on a hybrid bag-of-words/CNN scheme.

**Willow Action Dataset.** Our best results on Willow (Table 4 left) likewise outperforms VD-16-G by 3.35 mAP points and VD-16 *dense_evaluation* (Table 4 left) by 2.8 mAP points. For completeness, we have included several, previously-published results. To our knowledge, our approach outperforms the highest published results on this dataset.

**MIT-Indoor-67.** In Table 4, we present results on the MIT-Indoor-67dataset. For this dataset, we represent candidate regions using the Hybrid CNN model of [34], which is learned on a training set obtained by merging ImageNet and the Places database [34] and is better suited for scene recognition. Given the large size (4096) of these features, we reduce them to size 160 using PCA, similarly to the approach of [9]. Note that our method outperforms all other methods in

[1] Our own implementation of this method achieves results below those reported in [33].

**Table 4.** Comparison of results on the `Willow` dataset ($P = 7$ parts per-class, $K = 1$) *(left)* and the `MIT-Indoor-67`dataset ($P = 500$ parts, common to all classes, $K = 2$) *(right)*

| Methods | $mAP$ |
|---|---|
| Khan *et al.* [38] | 70.10 |
| Sharma *et al.* [39] | 65.90 |
| Sharma *et al.* [40] | 67.60 |
| Sicre *et al.* [24] | 81.90 |
| VD-16-G | 85.12 |
| VD-16 (dense_evaluation) | 85.67 |
| SPLeaP (*SS*+*aug*) | **88.47** |

| Methods | $mAP$ |
|---|---|
| Orderless [19] | 68.80 |
| MLPM [23] | 69.69 |
| HybridCNN-G [34] | 72.54 |
| HybridCNN-sum [34] | 70.36 |
| Parizi *et al.* [9] | 73.30 |
| SPLeaP-PCA160 (*SS*) | **73.45** |

Table 4. Unlike our reported results, those of [9] use a spatial pyramid with two scales and five cells (1 × 1, 2 × 2), as well as a different number of parts and PCA-reduction factor, resulting in features that are 3.73 times bigger than ours.

## 6 Qualitative Analysis

We now present qualitative results to illustrate the response of our learned part classifiers on `Pascal-VOC-2007` test examples.

In Fig. 3 we demonstrate the selectivity of our part detectors by presenting image triplets consisting, in order, of *(i)* the image with candidate region bounding boxes superposed, *(ii)* the original image, and *(iii)* heatmaps for the part responses of each candidate region. Note in particular the selectivity of our part detectors: in all examples, the actual object occupies but a small fraction of the image area.

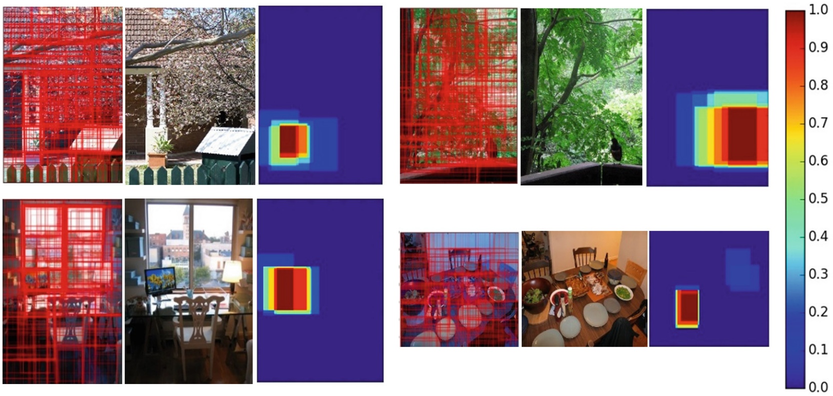

**Fig. 3.** Heatmaps for images `Pascal-VOC-2007` of classes *(clockwise from top-left)* "potted plant", "bird", "bottle" and "TV monitor".

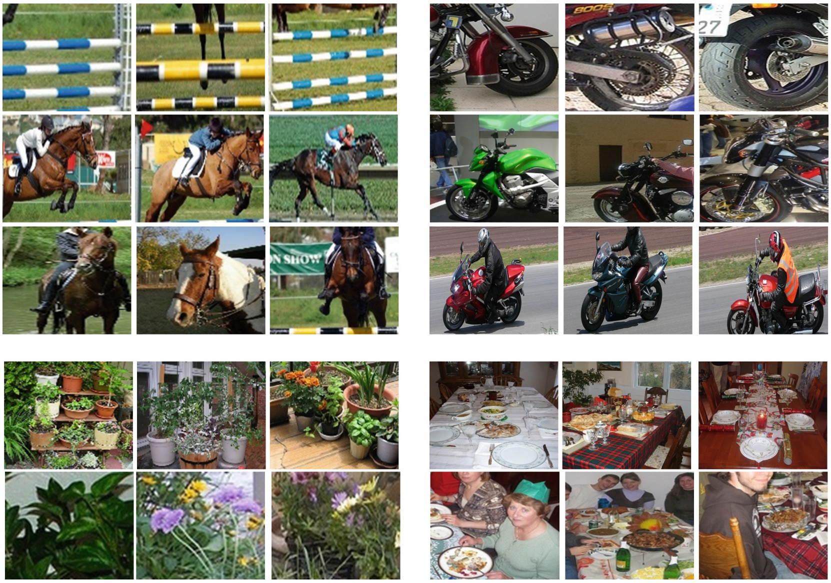

**Fig. 4.** Discriminative parts for the four `Pascal-VOC-2007` classes *(clockwise from top-left)* "horse", "motorbike", "dining table", and "potted plant".

In Fig. 4, we illustrate the highest ranking candidate regions from all images for the part classifiers associated to the largest entries in the corresponding weight vector $\mathbf{w}_c$, with each row of each group of images corresponding to a different part classifier. Note that the part classifiers all become specialized to different object parts or poses.

## 7 Conclusions

We introduce SPLeapP, a novel part-based model for image classification. Based on non-linear part classifiers combined with part-dependent soft pooling – both being trained jointly with the image classifier – this new image model consistently surpasses standard pooling approaches and part-based models on several challenging classification tasks. In addition, we have experimentally observed that the proposed method does not need any particular initialization of the parts, contrarily to most of the recent part-based models which require a first step for selecting a few regions candidates from the training images before they actually start learning the parts.

## References

1. Fischler, M.A., Elschlager, R.A.: The representation and matching of pictorial structures. IEEE Trans. Comput. **22**(1), 67–92 (1973)

2. Weber, M., Welling, M., Perona, P.: Towards automatic discovery of object categories. In: IEEE International Conference on Computer Vision and Pattern Recognition (2000)
3. Ullman, S., Sali, E., Vidal-Naquet, M.: A fragment-based approach to object representation and classification. In: Arcelli, C., Cordella, L.P., Sanniti di Baja, G. (eds.) IWVF 2001. LNCS, vol. 2059, pp. 85–100. Springer, Heidelberg (2001). doi:10.1007/3-540-45129-3_7
4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
5. Doersch, C., Gupta, A., Efros, A.A.: Mid-level visual element discovery as discriminative mode seeking. In: Proceedings on Neural Information Processing Systems (2013)
6. Singh, S., Gupta, A., Efros, A.: Unsupervised discovery of mid-level discriminative patches. In: European Conference on Computer Vision, pp. 73–86 (2012)
7. Juneja, M., Vedaldi, A., Jawahar, C., Zisserman, A.: Blocks that shout: distinctive parts for scene classification. In: IEEE International Conference on Computer Vision and Pattern Recognition (2013)
8. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.: What makes Paris look like paris? ACM Trans. Graph. **31**(4) (2012)
9. Parizi, S.N., Vedaldi, A., Zisserman, A., Felzenszwalb, P.: Automatic discovery and optimization of parts for image classification. In: International Conference on Learning Representations (2015)
10. Lobel, H., Vidal, R., Soto, A.: Hierarchical joint max-margin learning of mid and top level representations for visual recognition. In: IEEE International Conference on Computer Vision (2013)
11. Hoai, M., Zisserman, A.: Improving human action recognition using score distribution and ranking. In: Asian Conference on Computer Vision (2014)
12. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent in function space. In: NIPS (1999)
13. Friedman, J., Hastie, T., Tibshirani, R., et al.: Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). Ann. Stat. **28**(2), 337–407 (2000)
14. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: IEEE International Conference on Computer Vision and Pattern Recognition (2014)
15. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: delving deep into convolutional nets. In: British Machine Vision Conference (2014)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings on Neural Information Processing Systems (2012)
17. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Computer Vision and Pattern Recognition Workshops (2014)
18. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
19. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: European Conference on Computer Vision (2014)

20. Kulkarni, P., Zepeda, J., Jurie, F., Perez, P., Chevallier, L.: Hybrid multi-layer deep cnn/aggregator feature for image classification. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (2015)
21. Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: IEEE International Conference on Computer Vision and Pattern Recognition (2015)
22. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: IEEE International Conference on Computer Vision and Pattern Recognition (2010)
23. Li, Y., Liu, L., Shen, C., van den Hengel, A.: Mid-level deep pattern mining. In: IEEE International Conference on Computer Vision and Pattern Recognition (2015)
24. Sicre, R., Jurie, F.: Discovering and aligning discriminative mid-level features for image classification. In: International Conference on Pattern Recognition (2014)
25. Gulcehre, C., Cho, K., Pascanu, R., Bengio, Y.: Learned-norm pooling for deep feedforward and recurrent neural networks. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8724, pp. 530–546. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44848-9_34
26. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A.C., Bengio, Y.: Maxout networks. ICML **28**(3), 1319–1327 (2013)
27. Lee, C.Y., Gallagher, P.W., Tu, Z.: Generalizing pooling functions in convolutional neural networks: mixed, gated, and tree. In: International Conference on Artificial Intelligence and Statistics (2016)
28. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: IEEE International Conference on Computer Vision and Pattern Recognition (2009)
29. Delaitre, V., Laptev, I., Sivic, J.: Recognizing human actions in still images: a study of bag-of-features and part-based representations. In: British Machine Vision Conference (2010)
30. van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M.: Segmentation as selective search for object recognition. In: IEEE International Conference on Computer Vision (2011)
31. Chavali, N., Agrawal, H., Mahendru, A., Batra, D.: Object-proposal evaluation protocol is ‘gameable’. arXiv:1505.05836 (2015)
32. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia (2014)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
34. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Proceedings on Neural Information Processing Systems (2014)
35. Kulkarni, P., Zepeda, J., Jurie, F., Perez, P., Chevallier, L.: Max-margin, single-layer adaptation of transferred image features. In: BigVision Workshop, Computer Vision and Pattern Recognition (2015)
36. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
37. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE International Conference on Computer Vision and Pattern Recognition (2015)

38. Khan, F.S., Anwer, R.M., van de Weijer, J., Bagdanov, A.D., Lopez, A.M., Felsberg, M.: Coloring action recognition in still images. Int. J. Comput. Vis. **105**(3), 205–221 (2013)
39. Sharma, G., Jurie, F., Schmid, C.: Discriminative spatial saliency for image classification. In: IEEE International Conference on Computer Vision and Pattern Recognition (2012)
40. Sharma, G., Jurie, F., Schmid, C.: Expanded parts model for human attribute and action recognition in still images. In: IEEE International Conference on Computer Vision and Pattern Recognition (2013)