Semantic Part Segmentation with Deep Learning

Stavros Tsogkas
CentraleSupelec and INRIA
stavros.tsogkas@centralesupelec.fr

Iasonas Kokkinos
CentraleSupelec and INRIA
iasonas.kokkinos@centralesupelec.fr

George Papandreou
Google Inc.
gpapan@google.com

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

Abstract

In this work we address the task of segmenting an object into its parts, or semantic part segmentation. We start by adapting a state-of-the-art semantic segmentation system to this task, and show that a combination of a fully-convolutional Deep CNN system coupled with Dense CRF labelling provides excellent results for a broad range of object categories. Still, this approach remains agnostic to high-level constraints between object parts. We introduce such prior information by means of the Restricted Boltzmann Machine, adapted to our task and train our model in an entirely discriminative fashion, as a hidden CRF, demonstrating that prior information can yield additional improvements.

We evaluate the performance of our approach on the Penn-Fudan and LFW datasets for the tasks of pedestrian parsing and face labelling respectively. We show superior performance with respect to competitive methods that have been extensively engineered on these benchmarks, as well as realistic qualitative results on part segmentation, even in challenging cases, such as occluded or deformable objects. We also provide quantitative and extensive qualitative results on three classes from the PASCAL Parts dataset.

1. Introduction

Recently Deep Convolutional Neural Networks (DCNNs) have delivered excellent results in a broad range of computer vision problems, including but not limited to image classification [15, 31, 33, 35, 26], semantic segmentation [2, 19], object detection [7] and fine-grained categorization [42]. Given this broad success, DCNNs seem to have become the method of choice for any image understanding task where the input-output mapping can be described as a classification problem and the output space is a set of discrete labels.

In this work we investigate the use of DCNNs to address the problem of semantic part segmentation, namely segmenting an object into its parts. This is illustrated in Fig. 1 given the bounding box of an object, the goal is to decompose it into its constituent parts. Part segmentation is an important subproblem for tasks such as recognition, pose estimation, tracking, or applications that require accurate segmentation of complex shapes, such as a host of medical applications. For example, all state-of-the-art methods for fine-grained categorization rely on the localization and/or segmentation of object parts [42].

Part segmentation is also interesting from the modeling perspective, as the configurations of parts are, for most objects, highly structured. Incorporating prior knowledge about the parts of an object lends itself naturally to structured prediction, which aims at training a map whose output space has a well defined structure. The key question addressed in this paper is how DCNN can be combined with structured output prediction to effectively parse object...
parts. In this manner, one can combine the discriminative power of CNNs to identify part positions and prior information about object layout, to recover from possible failures of the CNN. Integrating DCNNs with structured prediction is definitely not novel. For example, early work by [18] used Graph Transformer Networks for parsing 1D lines into digits. However, the combination of DCNN with models of shape, such as the Shape Boltzmann Machines, or of object parts, such as Deformable Part Models, are recent [30] [39].

This work makes several contributions. First, we show that, by adapting the semantic segmentation system of [2] (Sec. 3), it is possible to obtain excellent results in part segmentation. This system uses a dense Conditional Random Field (CRF) applied on top of the output of a DCNN. This simple and non-specialized combination often outperforms specialized approaches to part segmentation and localization by a substantial margin. In Sec. 4 we turn to the problem of augmenting this system with a statistical model of the shape of the object and its parts. A key challenge is that the shape of parts is subject to substantial geometric variations, including potentially a variable number of parts per instance, caused by variations in the object pose. We model this variability using Restricted Boltzmann Machines (RBMs). These implicitly incorporate rich distributed mixture models in a representation that is particularly effective at capturing complex localized variations in shape.

In order to use RBMs with DCNNs in a structured-output prediction formulation, we modify RBMs in several ways: first, we use hidden CRF training to estimate the RBM parameters in a discriminative manner, aiming at maximizing the posterior likelihood of the ground-truth part masks given the DCNN scores as input. We demonstrate that this can yield an improvement over the raw DCNN scores by injecting high-level knowledge about the desired object layout.

Extensive experimental results, reported in Sec. 5, confirm the merit of our approach on four different datasets. The paper concludes in Sec. 6 with a summary of our findings.

2. Related Work

The layout of object parts (shape, for short) obeys statistical constraints that can be both strict (e.g. head attached to torso) and diverse (e.g. for hair). As such, accounting for these constraints requires statistical models that can accommodate multi-modal distributions. Statistical shape models traditionally used in vision, such as Active Appearance Models [4] or Deformable Part Models [6] need to determine in advance a small, fixed number of mixtures (e.g. 3 or 6), which may not be sufficient to encompass the variability of shapes due to viewpoint, rotation, and object deformations.

A common approach in previous works has been combining appearance features with a shape model to enforce a valid spatial part structure. In [1], the authors compute appearance and shape features on oversegmentations of cropped pedestrian images from the Penn-Fudan pedestrian dataset [40] [1]. They use color and texture histograms to model appearance and spatial histograms of segment edge orientations as the shape features. The label of each superpixel is estimated by comparing appearance and shape features to a library of exemplar segments. Small segments are sequentially merged into larger ones and simple constraints, (such as that “head” appears above “upper body” and that “hair” appears above “head”) enforce a consistent layout of parts in resulting segmentations.

Multi-modal distributions can be naturally captured through distributed representations [10], which represent data through an assembly of complementary patterns that can be combined in all possible ways. Restricted Boltzmann Machines (RBMs) [34] [11] provide a probabilistic distributed representation that can be understood as a discrete counterpart to Factor Analysis (or PCA), while their restricted, bipartite graph topology makes sampling efficient. Stacking together multiple RBMs into a Deep Boltzmann Machine (DBM) architecture allows us to build increasingly powerful probabilistic models of data, as demonstrated for a host of diverse modalities e.g. in [28].

The early studies of using RBMs and their deeper variants for digit modelling, e.g. [11] can be understood as the construction of statistical models on shapes. In [5] RBMs are thoroughly studied and assessed as models of shape. The authors additionally introduce the Shape Boltzmann Machine (SBM), a two-layer network that combines ideas from part-based modelling and DBMs, and show that it is substantially more flexible and expressive than single-layer RBMs. The same approach was extended to deal with multiple region labels (parts) in [5] and coupled with a model for part appearances. The layered architecture of the model allows it to capture both local and global statistics of the part shapes and part-based object segmentations, while parameter sharing during training helps avoid overfitting despite the small size of the training datasets.

RBMs are typically trained in a generative manner: given a set of shapes, the goal is to learn a probabilistic model that models the given shapes accurately and can generalize to unseen instances of the same shape category. Discriminative algorithms for training RBMs have mostly focused on using RBMs for classification tasks [17] [23] where the hidden variables of an RBM are used to solve a K-ary classification problem. By contrast, we are interested in predicting the structured data presented to the RBM during training - and as such need to resort to variants of RBMs tuned to Structured Output Prediction tasks [23].

The discriminative training of RBMs has been pursued in shape modelling by [13] in a probabilistic setting and by
in a max-margin setting. We pursue a probabilistic setting and detail our approach below. Despite small theoretical differences, the major practical difference between our method and the aforementioned ones is that we do not use any superpixels, pooled features, or boundary signals, as do, but we rather entirely rely on the CNN scores.

Multi-layer Boltzmann machines can generate realistic multi-label segmentations but produce blurred boundaries among adjacent part regions. Conditional Random Fields (CRFs) are well-suited to modelling local interactions between neighboring regions (e.g. superpixels) and are particularly effective when there is a clear difference between those regions. However, CRFs may fail to draw boundaries between regions with similar appearances, or when a region mingles with the background. In [13], Kae et al. jointly train a CRF and a Restricted Boltzmann Machine (RBM) [29] to build a hybrid model that combines global consistency and clear region boundaries.

3. DCNNs for semantic part segmentation

Deep Convolutional Neural Networks have proven to be particularly successful in “Semantic Image Segmentation”, the task of pixel-wise labeling of images [32, 19, 2]. In this section we adapt the recently introduced, state-of-art DeepLab system [2] to our task of semantic part segmentation.

Following [2], we adopt the architecture of the state-of-art 16-layer classification network of [33] (VGG-16). We employ it in a fully-convolutional manner, turning it into a dense feature extractor for semantic image segmentation. In particular, as in [32, 25, 19], treating the last fully-connected layers of the DCNN as $1 \times 1$ spatial convolution kernels. This allows us to treat the DCNN as a feature extraction module that efficiently provides features, or class posteriors for a regularly sampled set of image rectangles. The spatial density at which these posteriors are computed is increased with only a moderate additional computation cost: the decimation factor of the original VGG network is 32 in every dimension, but this is efficiently reduced to 8 by employing the à trous (with holes) algorithm [22, 2] – while delivering the exact same results as would be obtained by a brute-force re-evaluation of the network at intermediate steps. Similarly to [2], we employ linear interpolation to upsample by a (factor of 8) the class scores of the final network layer to the original image resolution. We learn the DCNN network parameters using training images annotated with semantic object parts at the pixel-level, minimizing the cross-entropy loss averaged over all image positions with Stochastic Gradient Descent (SGD). We initialize the network parameters from the ImageNet-pretrained VGG-16 model of [33].

The model’s ability to capture low-level information related to region boundaries is enhanced by employing the fully-connected Conditional Random Field (CRF) of [14], exploiting its ability to combine fine edge details with long-range dependencies. This particularly simple combination is both efficient and effective: the DCNN evaluation runs at 8 frames per second for a $320 \times 320$ image on a GPU and CRF inference requires 0.5 seconds on a CPU. Similarly to [2], we set the dense CRF hyperparameters by cross-validation, performing grid search to find the values that perform best on a small held-out validation set for each task.

In order to simplify the evaluation of the learned networks we fine-tune one network per object category, even though improvements for all part segmentation tasks could be possible by using a joint training procedure. The system is thoroughly evaluated in Sec. 5, but a few interesting observations emerge immediately from Fig. 1: first, the method is surprisingly effective in segmenting parts even for objects such as horses that exhibit complicated, articulated deformations. Second, although the model is not trained with structured loss, it is apparent that it can implicitly capture some form of contextual information: in many cases, it would be extremely hard to successfully segment parts if it only used evidence local to each pixel. This DCNN excels in this task because the effective receptive field at the top DCNN layers is in practice quite large. To probe this point further, we have experimented with DCNNs of varying receptive field size, finding that the LargeFOV architecture of [2] gives qualitatively better results.

While this DCNN + CRF model is very powerful, it can still make gross errors in some cases. Such errors could be corrected by introducing knowledge of the layout of objects, allowing for a better, more principled use of global information. Integrating this information is the goal of Sec. 4.

4. Conditional Boltzmann Machines

The aim of this section is to construct a probabilistic model of image segmentations that can capture prior information on the layout of an object category. The goal of this model is to complement and correct information extracted bottom-up from an image by the DCNN as explained in the previous section.

In order to construct this model, we introduce three types of variables: (i) the output $v$ of the densely-computed DCNN that is visible during both training and testing; (ii) the binary latent variables $h$ that are hidden during both training and testing; and (iii) the ground-truth segmentation labels $y$ that are observed during training and inferred during testing. The latter is a one-hot vector for each pixel, with $y_{i,k} = 1$ indicating that pixel $i$ takes label $k$ out of a set of $K$ possible choices (the parts plus background).

The conditional probability $P(y, h|v; W)$ of the labels and hidden variables given the observed DCNN features is
the Boltzmann-Gibbs distribution

\[ P(y | v; W) = \frac{\exp(-E(y, h, v; W))}{\sum_{y, h} \exp(-E(y, h, v; W))} \]  

where \( E(y, h, v; W) \) is an energy function described below. The posterior probability of the labelling is obtained by marginalizing the latent variables:

\[ P(y | v; W) = \sum_h P(y, h | v; W). \]  

The goal is to estimate the parameters \( W \) of the energy function during training and to use \( P(y | v; W) \) during testing to drive inference towards more probable segmentations.

Before describing the energy function \( E(y, h | v; W) \) in detail note that (i) the DCNN-based quantities \( v \) are always observed and the model does not describe their distribution; in other words, we construct a conditional model of \( y \) \cite{16 8}: (ii) unlike common CRFs, there are also hidden variables \( h \), which results in a Hidden Conditional Random Fields (HCRFs) \cite{27 24}; (iii) however, unlike the loopy graphs used in generic HCRFs, the factor graph in this model is bipartite, which makes block Gibbs sampling possible (Sec. 4.1).

Consider now the relationship between the DCNN output \( v \) and the pixel label \( y \) and recall that \( v \) are obtained from the last layer of the DCNN. The DCNN is trained so that, for a given pixel \( i \), \( v_i \) contains the class posteriors up to the softmax operation:

\[ P(y_{i,k} = 1 | v) = \frac{\exp(v_{i,k})}{\sum_{k'} \exp(v_{i,k'})}. \]  

This suggests that \( v_{i,k} \) can be used as a bias term for \( y_{i,k} \) in the energy model, such that a larger value of \( v_{i,k} \) rewards the assignment \( y_{i,k} = 1 \). The raw values of \( v \) are rescaled using a set of learnable parameters which allows auto-calibration during training. The contribution to the energy term is then:

\[ E_{CNN}(y, v; W) = \sum_{i,k,k'} w^C_{k,k'} y_{i,k} v_{i,k'}, \]  

where the CNN calibration parameters, \( w^C \), are contained in the overall model parameters \( W \). Note also that this formulation allows to learn interactions between classes as class \( k' \) as predicted by the DCNN can vote through weight \( w^C_{k,k'} \) for class \( k \) in the energy.

We can now write the term linking output and hidden variables, which takes the form of an RBM:

\[ E_{RBM}(y, h; W) = \sum_{i,j,k} y_{i,k} w^R_{i,j,k} h_j + \sum_{i,k} y_{i,k} w^R_{i,k,k}. \]  

Note that this does not include any ‘lateral’ connection between the observed variables, or between the hidden variables (this would correspond to terms in which pairs of the same type of variables are multiplied). Instead, there are two types of terms. The first type has biases \( w^R_{i,j,k} \) for each pixel location \( i \) and label \( k \), favoring certain labels based on their spatial location only. The second type expresses the interaction between labels and latent variables through the interaction weights \( w^R_{i,j,k} \). These weights determine the effect that activating the hidden node \( h_j \) has on labelling position \( i \) as part \( k \). Activating \( h_j \) will favor or discourage simultaneously the activation of labels at different locations according to the pattern encoded by the weights \( w^R_{i,j,k} \). Intuitively latent variables can in this manner encode segmentation fragments.

The overall energy is obtained as the sum of these two terms:

\[ E(y, h, v; W) = E_{CNN}(y, v; W) + E_{RBM}(y, h; W) \]  

By aggregating the output variables \( y \), the hidden variables \( h \), and the observable variables \( v \) into a single vector \( z \), the energy above can be rewritten in the form:

\[ E(z; W) = z^T W z \]  

where \( W \) is a matrix of interactions. Importantly, this matrix is block-diagonal to reflect the bipartite structure of the RBM, where only certain pairwise interactions are allowed. All these parameters can be estimated using discriminative training of RBMs \cite{2413} as detailed below.

### 4.1. Parameter Estimation for Conditional RBMs

Given a set of \( M \) training examples \( \mathcal{X} = \{ (y^1, v^1), \ldots, (y^M, v^M) \} \), parameter estimation aims at maximizing the conditional log-likelihood of the ground-truth labels:

\[ S(W) = \sum_{m=1}^M \log P(y^m | v^m; W) \]  

\[ = \sum_{m=1}^M \log \sum_{h} \frac{\exp(-E(y^m, h, v; W))}{Z(v^m)}, \]  

where \( Z(v^m) = \sum_{y, h} \exp(-E(y, h, v; W)) \).

Using the notation of Eq. [7], a parameter \( W_{k,m} \) connects nodes \( z_k \) and \( z_m \) that can be either hidden or visible. The partial derivative of the conditional log-likelihood with respect to \( W_{k,m} \) is given by \cite{24}:

\[ \frac{\partial S}{\partial W_{k,m}} = \sum_{m=1}^M \langle z_k z_m \rangle p(h | y^m, v^m; W) - \langle z_k z_m \rangle p(h, y | v^m; W) \]  

where \( \langle \cdot, \cdot \rangle \) denotes expectation.

In order to compute the first term, the \( y \) and \( v \) components of the \( z \) vector are given and one has to average over
the posterior on $h$ to compute the expectation of $z_k z_m$. To do so, one starts with the CNN scores ($v$) and the ground-truth segmentation maps ($y$) and computes the posterior over the hidden variables $h$, which can be obtained analytically. Then he computes the expectation of the product of any pair of interacting nodes, also in closed form.

In order to compute the second term one needs to consider the joint expectation over segmentations $y$ and hidden variables $h$ when presented with the CNN scores $v$. The exact computation of this term is intractable, and is instead computed through Monte Carlo approximation using Contrastive Divergence [9]. Namely we initialize the state $y$ to $y^{init}$, perform $C = 10$ iterations of Block-Gibbs sampling over $y$ and $h$, and use the resulting state as a sample from $P(h, y^{inf}; W)$.

This training algorithm is identical to RBM training with the difference that the partition function is image-dependent, resulting in minor algorithmic modifications.

4.2. Implementation Details

In our implementation we did not use any momentum terms, while for every update we used the whole batch, so as to reduce the number of parameters that had to be cross-validated. We observed that due to the small number of images in most of our training datasets (100-200) we had to use dataset augmentation to avoid overfitting. For every training image we generated 81 replicates by translating the images within a regular grid.

We used 200 iterations for training, while the decay terms (amounting to regularization coefficients) were set with cross-validation.

Some indicative results are shown in Fig. 2: we observe that each component is tuned to a different style, while combining these allows us to synthesize a rich set of face segmentation masks.

5. Experiments

We evaluate our method on four datasets (LFW, Penn-Fudan, CUB and PASCAL-parts) and report qualitative and quantitative results. We compare the accuracy of our pipeline before and after refining part boundaries using the fully-connected CRF, and also report on the improvements delivered by the combination of RBMs with CNNs on three categories (faces, cows, horses). While using the exact same settings for the network and parameter values described in Sec. 3, we obtain state-of-the-art results when comparing to carefully engineered approaches for the individual problems.

Penn-Fudan pedestrian dataset

The Penn-Fudan dataset [1] provides manual segmentations of 170 pedestrians into head, hair, clothes, arms, legs and shoes/feet. This dataset does not come with a train/test split, so we had to train our networks on a different dataset. We finetune our network on the Pascal person category, using all images and corresponding part annotations from [3].

A complication is that in PASCAL-Parts clothing is not taken into account when segmenting people - the only regions are “torso”, “arms”, “legs” and “feet”; whereas in Penn-Fudan the semantic parts used are “hair”, “face”, “upper clothes”, “arms”, “lower clothes”, “legs” and “shoes/feet”. To facilitate comparison of the methods, we merge “torso” and “arms” from PASCAL and “upper clothes” and “arms” from Penn-Fudan into “upper body”; similarly we merge “legs” and “feet” from PASCAL and “lower clothes”, “legs” and “feet” from Penn-Fudan into “lower body”. Other methods also report results on these two superregions, making comparison possible. Detailed numbers for Intersection-over-Union (IOU) for each part are included in Tab. 1a.

Labeled Faces in the Wild

Labeled Faces in the Wild (LFW) is a dataset containing more than 13000 images of faces collected from the web. For our purposes, we used the “funneled” version of the dataset, in which images have been coarsely aligned using a congealing-style joint alignment approach [12]. This is the subset also used in [13] and consists of 1500 train, 500 validation and 927 testing images of faces, and their corresponding superpixel segmentations, with labels for background, hair (including facial hair) and face. We train our DCNN on the 2000 trainval images and evaluate on the 927 test images, using superpixel accuracy as in [13] for the purpose of comparison. Since our system returns pixelwise labels for each image, we employ a simple scheme to obtain superpixel labels: for each superpixel we compute a histogram of the pixel labels it contains and choose the most
### Table 1: Segmentation accuracies of our system on the Penn-Fudan and LFW datasets

| Method          | head | upper body | lower body | FG  | BG  | Average |
|-----------------|------|------------|------------|-----|-----|---------|
| SBP [1]         | 51.6 | 72.6       | 71.6       | 73.3| 81.0| 70.3    |
| DDN [21]        | 60.2 | 75.7       | 73.1       | 78.4| 85.0| 74.5    |
| DL [21]         | 60.0 | 76.3       | 75.6       | 78.7| 86.3| 75.4    |
| Ours (CNN)      | 67.8 | 77.0       | 76.0       | 83.0| 85.4| 77.8    |
| Ours (CNN+CRF)  | 64.2 | 81.5       | 80.9       | 84.4| 87.3| 79.7    |

(a) Segmentation accuracies on Penn-Fudan.

| Method          | Accuracy (SP) |
|-----------------|---------------|
| GLOC [13]       | 94.95%        |
| Ours (CNN)      | 96.54%        |
| Ours (CNN+SBM)  | 96.78%        |
| Ours (CNN+CRF)  | 96.76%        |
| Ours (CNN+SBM+CRF) | 96.97%     |

(b) Superpixel accuracies on LFW.

![Figure 3: Pedestrian parsing results on Penn-Fudan dataset. From top to bottom: a) Input image, b) SBP [1], c) Raw CNN scores, d) CNN+CRF, e) Groundtruth.](image)

CUB-200-2011 does not contain segmentation masks for parts, however Zhang et al. [42] provide bounding boxes for the whole bird, as well as for its head and body. In that work, the authors describe a system for detecting object parts under two different settings: 1) When the object bounding box is given, and 2) when the location of the ob-

**Caltech-UCSD Birds-200-2011**

CUB-200-2011 [38] is a dataset for fine-grained recognition that contains over 11000 images of various types of birds.
Figure 4: Face parsing results on LFW. From top to bottom: a) Input image, b) Masks from raw CNN scores, c) CNN+CRF, d) Groundtruth. CRF sharpens boundaries, especially in the case of long hair. In the 6th image we see a failure case: our system failed to distinguish the man’s very short hair from the similar-color head skin.

Figure 5: Bird part segmentations returned by our network and inferred bounding boxes for head and body. We use green color for the groundtruth and magenta for our output.

**Pascal Parts dataset**

In our last experiment, we evaluate our system on the PASCAL Parts dataset [3]. This dataset includes high quality part annotations for the 20 PASCAL object classes (train and val sets), but was released fairly recently, so there are not many works reporting part segmentation performance. The only work that we know of is by Lu et al. [20] on car parsing, but the authors do not provide quantitative results in the form of some accuracy percentage, making comparison challenging.

Nevertheless, we report our own results for horse, cow and car, which could serve as a first baseline. For each class we train a separate DCNN on the train set annotations (using horizontal flipping to augment the training dataset), and test on the validation set. Our quantitative results are compiled in tables 3, while in figure 6 we show qualitative results.

In Tables 1b and 4 we report on the relative performance of the CNN-based system compared to the CNN-SBM combination, as well as the results we obtain when combined with the CRF system. For the “cow” and “horse” categories we also consider a separate subset of images containing poses of only moderate variation, to focus on cases that should be tractable for an RBM-based shape prior.

We observe that while the RBM typically yields a moderate improvement in performance over the CNN, this does
not necessarily always carry over to the combination of these results with the CRF post-processing module. This suggests that also the CRF stage should be trained jointly, potentially along the lines of [13], which we leave for future work.

Table 3: IOU scores on PASCAL-Parts.

| Method     | head | torso | legs | tail | BG   | Average |
|------------|------|-------|------|------|------|---------|
| CNN        | 55.0 | 34.2  | 52.4 | 66.8 | 37.2 | 76.0    |
| CNN+CRF    | 55.4 | 31.9  | 53.6 | 43.4 | 37.7 | 77.9    |

(a) IOU scores on PASCAL-Parts horse class

| Method     | head | torso | legs | tail | BG   | Average |
|------------|------|-------|------|------|------|---------|
| CNN        | 57.6 | 62.7  | 38.5 | 11.8 | 69.7 | 48.03   |
| CNN+CRF    | 60.0 | 64.8  | 34.8 | 9.9  | 72.4 | 48.38   |

(b) IOU scores on PASCAL-Parts cow class

Table 4: Results due to joint RBM and CNN training.

| Method                   | Val set | Val subset |
|--------------------------|---------|------------|
| CNN                      | 77.3    | 83.7       |
| CNN+RBM                  | 77.7    | 84.4       |
| CNN+CRF                  | 79.1    | 83.5       |
| CNN+RBM+CRF              | 79.2    | 84.7       |

(a) PASCAL-cow

| Method                   | Val set | Val subset |
|--------------------------|---------|------------|
| CNN                      | 76.6    | 86.4       |
| CNN+RBM                  | 76.3    | 86.9       |
| CNN+CRF                  | 77.6    | 88.1       |
| CNN+RBM+CRF              | 76.7    | 87.6       |

(b) PASCAL-horse

6. Discussion

In this work we have demonstrated that a simple and generic system for semantic segmentation relying on Deep CNNs and Dense CRFs can provide state-of-the-art results in the task of semantic part segmentation, outperforming highly sophisticated techniques that have been carefully engineered in a category-specific manner. We have also explored methods of integrating high-level information through a joint discriminative training of the network with a statistical, category-specific shape prior, showing that these can act in a complementary manner to the bottom-up information provided by DCNNs.

In future work we aim at exploring the joint training of
DCNNs with high-level shape priors in an end-to-end manner, as well as to further explore the practical applications of semantic part segmentation in detection and fine-grained recognition.

References

[1] Y. Bo and C. C. Fowlkes. Shape-based pedestrian parsing. In Proc. CVPR, pages 2265–2272. IEEE, 2011.
[2] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. CoRR, abs/1412.7062, 2014.
[3] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In CVPR, 2014.
[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. IEEE Trans. PAMI, 23(6):681–685, 2001.
[5] S. A. Eslami, N. Heess, C. K. Williams, and J. Winn. The shape boltzmann machine: a strong model of object shape. IJCV, 107(2):155–176, 2014.
[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ranchan. Object detection with discriminatively trained part-based models. IEEE Trans. PAMI, 32(9):1627–1645, 2010.
[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
[8] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. pages 695–702, 2004.
[9] G. Hinton. A practical guide to training restricted boltzmann machines.
[10] G. Hinton, J. McClelland, and D. Rumelhart. Distributed representations. In PDP. MIT Press, 1986.
[11] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. 2006.
[12] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In ICCV, 2007.
[13] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller. Augmenting crfs with boltzmann machine shape priors for image labeling. In CVPR, pages 2019–2026. IEEE, 2013.
[14] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In NIPS, 2011.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2013.
[16] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML, pages 282–289. 2001.
[17] H. Larochelle and Y. Bengio. Classification using discriminative restricted boltzmann machines. In ICML, 2008.
[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE.
[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.
[20] W. Lu, X. Lian, and A. Yuille. Parsing semantic parts of cars using graphical models and segment appearance consistency. Proc. BMVC, 2014.
[21] P. Luo, X. Wang, and X. Tang. Pedestrian parsing via deep decompositional network. In ICCV, 2013.
[22] S. Mallat. A Wavelet Tour of Signal Processing. Acad. Press, 2 edition, 1999.
[23] V. Mnih, H. Larochelle, and G. E. Hinton. Conditional restricted boltzmann machines for structured output prediction. In UAI, 2011.
[24] K. Murphy. Machine Learning: a Probabilistic Perspective. MIT Press, 2012.
[25] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Weakly supervised object recognition with convolutional neural networks. Technical report, INRIA, 2014.
[26] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection. arXiv:1412.0296, 2014.
[27] A. Quattoni, S. B. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. IEEE PAMI, 2007.
[28] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In AISTATS, pages 448–455, 2009.
[29] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning, pages 791–798. ACM, 2007.
[30] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. CoRR, abs/1503.02351, 2015.
[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229, 2013.
[32] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. 2014.
[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.
[34] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In PDP. MIT Press, 1986.
[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv:1409.4842, 2014.
[36] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In NIPS, 2014.
[37] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In ICCV, 2011.
[38] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.
[39] L. Wan, D. Eigen, and R. Fergus. End-to-end integration of a convolutional network, deformable parts model and non-maximum suppression. arXiv, 2014.
[40] L. Wang, J. Shi, G. Song, and I.-f. Shen. Object detection combining recognition and segmentation. In *ACCV*. 2007.

[41] J. Yang, S. Safar, and M. Yang. Max-margin boltzmann machines for object segmentation. In *CVPR*. 2014.

[42] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *Proc. ECCV*, pages 834–849. Springer, 2014.