Region-based semantic segmentation with end-to-end training

Holger Caesar, Jasper Uijlings, Vittorio Ferrari

University of Edinburgh

Abstract. We propose a novel method for semantic segmentation, the task of labeling each pixel in an image with a semantic class. Our method combines the advantages of the two main competing paradigms. Methods based on region classification offer proper spatial support for appearance measurements, but typically operate in two separate stages, none of which targets pixel labeling performance at the end of the pipeline. More recent fully convolutional methods are capable of end-to-end training for the final pixel labeling, but resort to fixed patches as spatial support. We show how to modify modern region-based approaches to enable end-to-end training for semantic segmentation. This is achieved via a differentiable region-to-pixel layer and a differentiable free-form Region-of-Interest pooling layer. Our method improves the state-of-the-art in terms of class-average accuracy with 64.0% on SIFT Flow and 49.9% on PASCAL Context, and is particularly accurate at object boundaries.

1 Introduction

We address the task of semantic segmentation, labeling each pixel in an image with a semantic class. Currently, there are two main paradigms: classical region-based approaches [1–17] and, inspired by the Convolutional Neural Network (CNN) revolution, fully convolutional approaches [18–26].

In the fully convolutional approach the idea is to directly learn a mapping from image pixels to class labels using a CNN. This results in a single model, directly optimized end-to-end for the task at hand, including the intermediate image representations (i.e. the hidden layers in the network). However, the spatial support on which predictions are based are fixed-size square patches of the input image. Intuitively, this is suboptimal since: (I) Objects are free-form rather than square, so ideally the intermediate representations should take this into account. (II) Objects do not have a fixed size, but occur at various scales. Hence many patches either cover pieces of multiple objects and mix their representations, or cover a piece of an object, which is sometimes difficult to recognize in isolation (e.g. a patch on the belly of a cow). An additional problem is that fully convolutional methods typically make predictions at a coarse resolution, which often results in inaccurate object boundaries [18, 20–22, 24, 26]. Fig. 1 illustrates this on example outputs of [20].

In the region-based approach, the image is first segmented into coherent regions, which are described by image features [1–16]. Typically many regions are

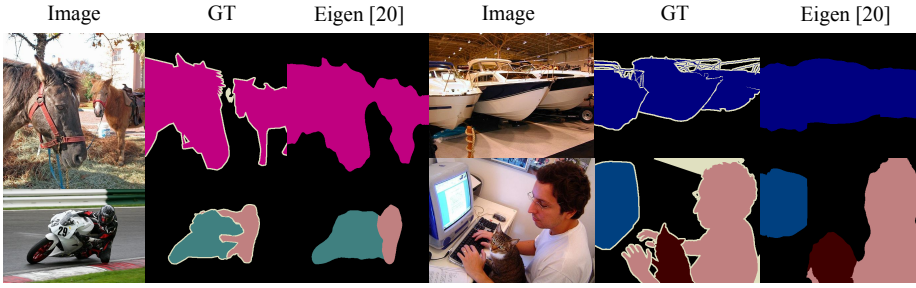

Fig. 1. *Fully convolutional methods typically produce fuzzy object boundaries, as illustrated here by examples from Eigen and Fergus [20].*

extracted at multiple scales [2–4, 6–8, 10–12], capturing complete objects and canonical object parts (e.g. faces) which in turn facilitates recognition. Furthermore, the segmentation process delivers regions which follow object boundaries quite well. However, these methods generally first extract region features and then train a classifier optimized for classifying regions rather than for the final semantic segmentation criterion (i.e. pixel-level labeling) [2–4, 6–8, 10]. Hence, while these methods benefit from the power of multi-scale, overlapping regions, they cannot be trained end-to-end for semantic segmentation.

In this paper we want the best of both worlds. We propose a region-based semantic segmentation model with an accompanying end-to-end training scheme based on a CNN architecture (Fig. 2c). To enable this we introduce a novel, differentiable region-to-pixel layer which maps from regions to image pixels. We insert this layer before the final classification layer, enabling the use of a pixel-level loss which allows us to directly optimize for semantic segmentation. Conceptually, our region-to-pixel layer ignores regions which have low activations for all classes and which therefore do not impact the final labeling. This is in contrast to all multi-scale region-based methods where such regions incorrectly affect training [2–4, 6–8, 10]. Additionally, we introduce a differentiable Region-of-Interest pooling layer which operates on the final convolutional layer in the spirit of Fast R-CNN [27], but which is adapted for free-form regions like [4, 11, 12]. Note how we use region proposals from a separate pre-processing stage. By end-to-end we mean training all parameters for the final pixel-level loss, rather than for region classification.

To summarize, our contributions are: (1) We introduce a region-to-pixel layer which enables full end-to-end training of semantic segmentation models based on multi-scale overlapping regions. (2) We introduce a Region-of-Interest pooling layer specialized for free-form regions. (3) We obtain state-of-the-art results on the SIFT Flow and the PASCAL Context datasets, in terms of class-average accuracy. Our approach delivers crisp object boundaries, as demonstrated in Fig. 5 and Sect. 4.3. We release the source code of our method at https://github.com/nightrome/matconvnet-calvin

2 Related Work

2.1 Region-based semantic segmentation

Region-based semantic segmentation methods first extract free-form regions [28–31] from an image and describe them with features. Afterwards a region classifier is trained. At test time, region-based predictions are mapped to pixels, usually by labeling a pixel according to the highest scoring region that contains it. Region-based methods generally yield crisp object boundaries [1–17]. Fig. 2b shows a prototypical architecture for such an approach (which we modernized by basing it on Fast R-CNN [27]). We discuss several aspects below.

Multi-scale vs single-scale regions. Several region-based methods use an oversegmentation to create small, non-overlapping regions [1, 5, 9, 13–16]. Intuitively however, objects are more easily recognized as a whole than by looking at small object parts individually. The inherent multi-scale aspect of recognition is adequately captured in many recent works using multi-scale, overlapping regions [2–4, 6–8, 10–12].

Training criterion. The final criterion is pixel-level prediction of class labels. However, we use overlapping regions whose predictions are in competition with each other on the pixel level. Typically, many methods initially ignore this by simply training a classifier to predict region labels [2–4, 6–8, 10], which is *different* from semantic segmentation (Sec. 3.1). At test time one labels a pixel by simply taking the maximum over all regions containing it [2–4, 6, 7]. A few works partially addressed the mismatch between training and test time through a post-processing stage using graphical models [8, 10] or by joint calibration [2]. However, none of them does full end-to-end training.

Region representations. Most older works use hand-crafted region-based features [1, 3, 5, 8, 10, 13–16] often based on [3, 13]. More recent works instead use the top convolutional layers of a pre-trained CNN (e.g. [32, 33]) as feature representations [2, 4, 6, 7, 9, 11, 12]. These representations can be free-form respecting the shape of the region [4, 6, 7, 9, 11, 12] or simply represent the bounding box around the region [2, 6]. Furthermore regions can be cropped out from the image before being fed to the network [6, 7, 9] or one can create region representations from a convolutional layer [4, 11, 12], termed Region-of-Interest (ROI) pooling [27] or Convolutional Feature Masking [4]. CNN representations become more powerful when further trained for the task. In [2, 6, 7] they train CNNs, but for the task of region classification, not for semantic segmentation.

2.2 Fully convolutional semantic segmentation

Fully convolutional methods learn a direct mapping from pixels to pixels, which was pioneered by [34] in the pre-CNN era. Early CNN-based approaches train relatively shallow end-to-end networks [21, 25], whereas more recent works use much deeper networks whose weights are initialized by pre-training on the ILSVRC [35]

image classification task [18–20, 22–24, 26]. The main insight to adapt these networks for semantic segmentation was to re-interpret the classification layer as 1x1 convolutions [36, 23]. A prototypical model is illustrated in Fig. 2a.

Square receptive fields. All fully convolutional methods have receptive fields of fixed shape (square) [18–26]. However, since objects are free-form this may be suboptimal.

Multi-scale. Recognition is a multi-scale problem, which is addressed by using two strategies: *(I) Multi-scale representations.* Using skip-layer connections [37, 38], representations from different convolutional layers can be combined [20, 22, 23, 25]. This leads to multi-scale representations of a predetermined size. *(II) Multi-scale application.* In [22, 24] they train and apply their method on multi-scale, rectangular image crops. However, this results in a mismatch between training time, where each crop is considered separately, and test time, where predictions of multiple crops are combined before evaluation.

Fuzzy object boundaries. It is widely acknowledged that fully convolutional approaches yield rather fuzzy object boundaries [18, 20–22, 24, 26]. A variety of strategies address this. *(I) Multi-scale.* The multi-scale methods discussed above [20, 22–25] include a fine scale resulting in improved object boundaries. *(II) Conditional Random Fields (CRFs).* CRFs are a classical tool to refine pixel-wise labelings and are used as post-processing step by [18, 21, 24, 26]. Notably, [26] reformulate the CRF as a recurrent neural network enabling them to train the whole network including convolutional layers in an end-to-end fashion. *(III) Post-processing by region proposals.* Finally, [21] averages pixel-wise network outputs over regions from an oversegmentation.

2.3 This paper

We propose a model based on free-form, multi-scale, overlapping regions. We design a partially differentiable region-to-pixel layer enabling end-to-end training for semantic segmentation. Additionally we introduce a ROI pooling layer which is free-form [4, 11, 12] yet also differentiable [27].

3 Method

Section 3.1 presents a baseline model that is representative for modern region-based semantic segmentation [2, 4, 6, 7] (Fig. 2b), and explains its shortcomings. Sections 3.2-3.5 present our framework, which addresses these issues (Fig. 2c).

3.1 Region-based semantic segmentation

Model. Fig. 2b presents a typical region-based semantic segmentation architecture. It modernizes [2, 4, 6, 7] by using the Region-of-Interest pooling layer of [27]. We use this model as a baseline in our experiments (Sec. 4).

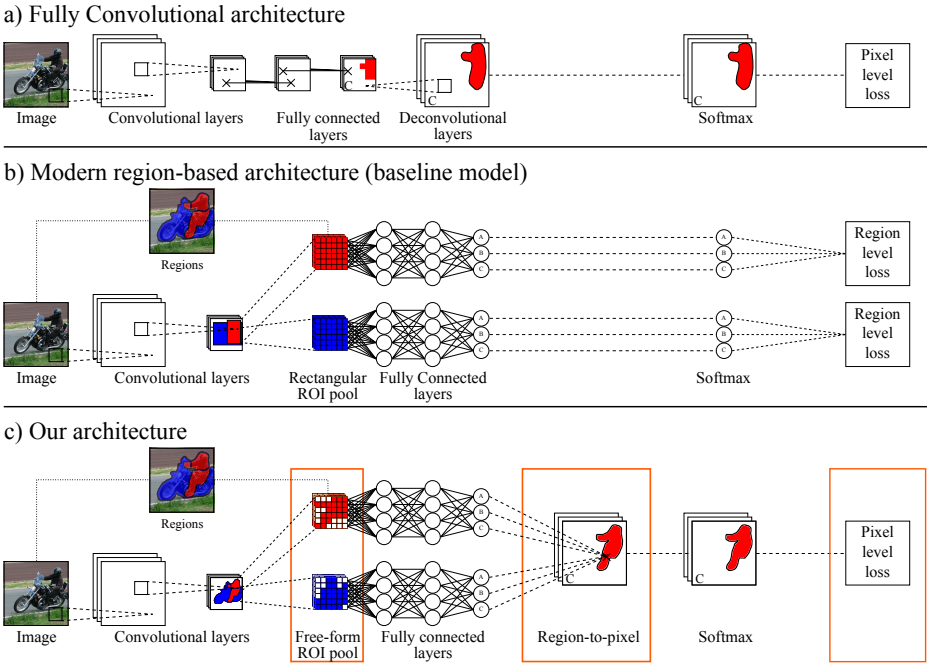

Fig. 2. *Overview of three semantic segmentation architectures. We show only layers with trainable parameters, softmax and loss layers. We omit all pre- and post-processing steps. a) shows the class of fully convolutional architectures that are end-to-end trainable, but do not have regions. b) shows the baseline model, representative for modern region-based architectures. It is not end-to-end trainable for the desired pixel labeling criterion. c) shows our suggested architecture, which pools activations of each region in a free-form manner, maps the region-level predictions to pixels and computes a loss at the pixel level. Hence our method combines regions and end-to-end training. Our main contributions are highlighted by orange boxes.*

The input to the network are images and free-form regions [29]. The image is fed through several convolutional layers. A Region-of-Interest pooling layer [27] creates a feature representation of the tight bounding boxes around each region. These region features are then fed through several fully connected layers and a classification layer, followed by a softmax, resulting in region-level predictions. At test time, these predictions are mapped from regions to pixels: each pixel p is assigned the label o_p with the highest probability over all classes and all regions containing p:

$$o_p = \operatorname*{argmax}_c \max_{r \ni p} \operatorname*{softmax}_c S_{r,c} \tag{1}$$

Here $S_{r,c}$ denotes the classifier scores for region r and class c (i.e. activations of the classification layer).

Training. The training procedure searches for the network parameters that minimize a cross-entropy log-loss $\mathcal{L}$ over regions:

$$\mathcal{L} = -\sum_{c} \frac{1}{R} \sum_{r=1}^{R} y_{r,c} \log \underset{c}{\text{softmax}}\, S_{r,c} \tag{2}$$

Here R indicates the number of regions in the training set and $y_{r,c} \in \{0, 1\}$ is a ground truth label indicating whether region r has label c. The network is trained with Stochastic Gradient Descent (SGD) with momentum. To update the network weights, one needs to compute the partial derivatives of the loss with respect to the weights. These derivatives depend on the partial derivatives of the loss with respect to the outputs of the respective layer.

Problems. A first problem arises because the softmax is applied before pixel assignment in Eq. (1): (I) regions with low but highly varying activation scores are unsure about the class, but can still yield high probabilities due to the softmax. Intuitively, this means that such non-discriminative regions can wrongly affect the final prediction.

More importantly, since $\max_{r \ni p}$ occurs at test time (Eq. (1)), but not at training time (Eq. (2)), the pixel-wise evaluation criterion at test time is *different* from the region-level optimization criterion at training time. This has several consequences: (II) While during training *all* regions affect the network, at test time most regions are ignored. (III) It is unclear what are good region training examples for achieving good performance at test time: Are positive examples only ground truth regions? Or should we use also region proposals which partially overlap with the ground truth? And with what threshold? What overlap are negative proposals allowed to have to count as negative examples? Hence one has to select overlap thresholds for positive and negative examples empirically using test time evaluations. (IV) Regions with different size have the same weight. (V) The network is not trained end-to-end for semantic segmentation, but for the intermediate task of region classification instead. Hence both the classification layer and the representation layers will be suboptimal for the actual semantic segmentation task.

3.2 End-to-end training for region-based semantic segmentation

Model. To combine the paradigms of region-based semantic segmentation and end-to-end training, we map from regions to pixels as in Eq. (1), but *before* the softmax and loss computation on a pixel-level:

$$o_p = \underset{c}{\text{argmax}}\, \underset{c}{\text{softmax}} \max_{r \ni p} S_{r,c} \tag{3}$$

This region-to-pixel layer is shown in Fig. 2c. It brings two benefits. At training time, having the region-to-pixel layer before the loss enables optimizing a pixel-level loss. Furthermore, having the region-to-pixel layer before the softmax ensures that the class score for each pixel is taken from the region with the highest activation score, hence each class can be recognized at its appropriate scale.

Training. In Eq. (2) the baseline model computes a cross-entropy log-loss on the region-level. Here instead we compute a log-loss on the pixel-level:

$$\mathcal{L} = -\sum_{c} \frac{1}{P} \sum_{p=1}^{P} y_{p,c} \ \log \underset{c}{\text{softmax}} \, S_{p,c} \tag{4}$$

Here P indicates the number of pixels in the training set, $y_{p,c} \in \{0, 1\}$ indicates whether pixel p has ground truth label c, and $S_{p,c} = \max_{r \ni p} S_{r,c}$ is the pixel-level score for class c. As in Section 3.1 we train the network using SGD. To determine the partial derivatives of our region-to-pixel layer, we observe that it does not have any weights and we only need to compute the subgradients of the loss with respect to the region-level scores $S_{r,c}$:

$$\frac{\partial \mathcal{L}}{\partial S_{r,c}} = \sum_{p \in r \ | \ r = \text{argmax}_{r' \ni p'} \ S_{r',c}} \frac{\partial \mathcal{L}}{\partial S_{p,c}} \tag{5}$$

This means that for each class we map each pixel-level gradient to the region with the highest score among all regions that include the pixel. If multiple pixels per class map to the same region, their gradient contributions are summed.

Advantages. Our model addresses all problems raised in Sec. 3.1: (I) Pixels are always labeled according to the relevant region with the highest activation score for that class. (II) Regions which do not affect the pixel-level prediction are ignored during training. (III) Since we evaluate pixels there is no need to assign class labels to regions for training. (IV) The pixel-level loss is agnostic to different sizes of region proposals. (V) We train our method end-to-end for the actual semantic segmentation criterion, resulting in properly optimized classifiers and region representations.

3.3 Pooling on free-form regions

Model. While the baseline model classifies free-form regions, their feature representations are computed on the bounding box. This is suboptimal as the regions can take highly irregular shapes. We propose here a free-form Region-of-Interest (ROI) pooling layer which computes representations taking into account only pixels actually in the region (Fig. 2c):

$$S^R_{i,d,r} = \max_{j \ | \ \phi(j) = i, \ \delta_{j,r} = 1} S^C_{j,d} \tag{6}$$

Here $S^R_{i,d,r}$ is the ROI pooling activation for ROI coordinate i, channel d and region r. For each ROI coordinate and channel we maximize over the corresponding coordinate j in the convolutional map $S^C_{j,d}$, considering only points inside the region, i.e. $\delta_{j,r} = 1$. The mapping ϕ from convolutional map coordinates to ROI ones is done as in [27, 39], but operates on a free-form region rather than a bounding box.

Training. During the forward pass the highest scoring convolutional map coordinate $\pi(i, d, r)$ for each ROI coordinate and channel is computed as:

$$\pi(i, d, r) = \underset{j \mid \phi(j) = i,\ \delta_{j,r} = 1}{\operatorname{argmax}} S^C_{j,d} \tag{7}$$

We use the technique of [27] to backpropagate through the pooling layer, computing the subgradients of the loss with respect to each coordinate in the last convolutional feature map. For each coordinate and channel in the ROI pooling output of a region, the gradients are passed to the convolutional feature map coordinate with the highest activations during the forward pass:

$$\frac{\partial \mathcal{L}}{\partial S^C_{j,d}} = \sum_r \sum_{i \mid \pi(i,d,r) = j} \frac{\partial \mathcal{L}}{\partial S^R_{i,d,r}} \tag{8}$$

Advantages. Our free-form region representations focus better on the region of interest, leading to purer representations. Additionally, they solve a common problem with bounding boxes: when objects of two classes occur in a part-container relationship (i.e. a bird in the sky), their free-form region proposals degenerate to the same bounding box. Hence higher network layers will receive two identical feature vectors for two different regions covering different classes. This leads to confusion between the two classes, both at training and test time.

Incorporating region context. Several works have shown that including local region context improves semantic segmentation [4, 6, 22], as many object classes appear in a characteristic context (e.g. a lion is more likely to occur in the savanna than indoors). We take into account region context by performing ROI pooling also on their bounding boxes using [27]. Hence we combine the advantages of using context with the advantages of free-form region representations.

As shown in Fig. 3, we combine region and bounding box representations using one of two strategies: *(I) Tied weights.* We use the same fully connected layers with the same weights for both region and bounding box representations and add the corresponding activations scores after the classification layers. Hence the number of network parameters stays the same and the region and its context are handled identically. *(II) Separate weights.* We concatenate the representations of region and bounding box before applying the consecutive fully connected layers. This strategy roughly doubles the total number of weights of our overall network architecture, but can develop separate classifiers for each representation.

Since ROI pooling on bounding boxes and free-form regions are both differentiable, the combined representations are also differentiable and allow for end-to-end training. We compare all representations experimentally (Table 4).

Relation to [27, 6, 4]. Girshick et al. [27] use a differentiable ROI pooling layer in Fast R-CNN for bounding boxes only. Girshick et al. [6] use free-form regions in R-CNN for semantic segmentation. For each region proposal they set the color values of the background pixels to zero. In our scheme we do not alter the image pixels of the input but pool exclusively over pixels inside the region. Dai et al. [4] perform Convolutional Feature Masking on the last convolutional

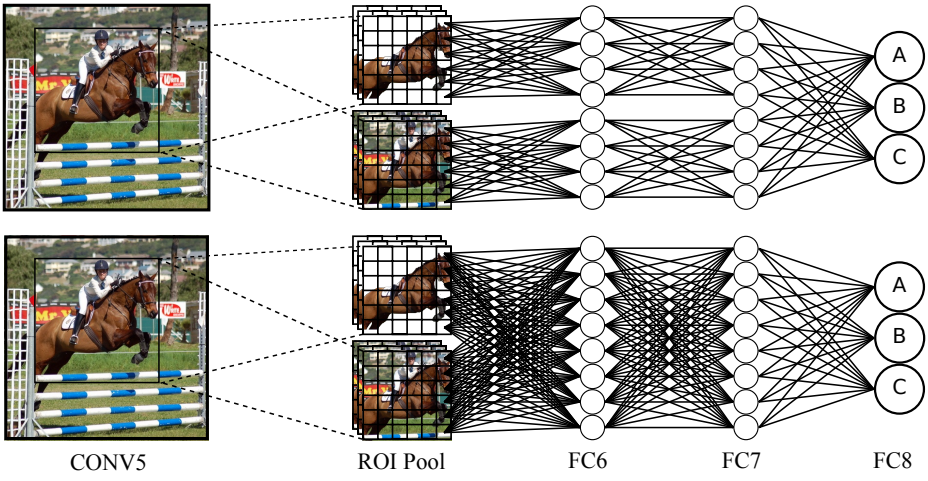

Fig. 3. *We combine free-form region representations, which focus on the appearance of the region itself, with bounding box based representations, which also capture context. We combine them using tied weights (above) and separate weights (below).*

feature map, followed by a Spatial Pyramid Pooling layer [39], but did not backpropagate through this layer. Both [4, 6] combined free-form and bounding box representations. Only [4] took representations after the convolutional layers, but their model was not able to perform backpropagation. Both [4, 6] optimized for region classification instead of semantic segmentation.

3.4 Attention to rare classes

Pixel-level class frequencies are often unbalanced [2, 9, 11, 12, 14, 16, 20, 21, 40–42]. This is typically addressed by using an inverse class frequency weighting $\frac{1}{P_c}$ [9, 11, 20, 21]. Since we have a pixel-level loss, we can simply plug this into Eq. (4). However, we found that rare classes lead to large weight updates resulting in exploding gradients and numerical problems. To avoid these issues, we re-normalize the inverse frequency weights by a factor Z so that the total sum of weights for each training image is 1: $\frac{1}{Z}\sum_c \frac{1}{P_c}\sum_{p=1}^{P} y_{p,c} = 1$.

3.5 Efficient evaluation of the pixel-level loss

Evaluating the loss for each pixel separately is computationally expensive and redundant, because different pixels belonging to the same highest scoring region for a class are assigned the same score $S_{r,c}$. Hence we partition the set of region proposals for a training image into a set of non-overlapping, single-class regions using the ground truth. We then reformulate Eq. (4) into an equivalent loss in terms of these regions. This reduces the cost of loss evaluation by a factor 1000.

4 Experiments

4.1 Setup

Datasets. We evaluate our method on two challenging datasets: SIFT Flow [43] and PASCAL Context [17]. SIFT Flow contains 33 classes in 2688 images. The dataset is known for its extreme class imbalance [43, 21, 20]. We use the provided fixed split into 2488 training images and 200 test images.

PASCAL Context provides complete pixel-level annotations for both things and stuff classes in the popular PASCAL VOC 2010 [44] dataset. It contains 4998 training and 5105 validation images. As there is no dedicated test set available, we use the validation images exclusively for testing. We use the 59 classes plus background commonly used in the literature [4, 19, 23, 26].

Evaluation measures. Semantic segmentation methods typically measure global accuracy and class-average accuracy. Global accuracy is the percentage of correctly labeled pixels in the dataset. But since class frequencies typically follow a power-law distribution, it is mostly influenced by a few common classes. Class-average accuracy instead takes all classes into account equally and it is generally considered a better measure. It first computes the accuracy for each class separately, and then averages over classes. Both measures are standard for SIFT Flow. The most common evaluation measure on PASCAL Context is mean Intersection-over-Union (IOU) [44]. For each class one divides the number of pixels of the intersection of the predicted and ground truth class by their union. Then the average is taken over classes.

Network. We use the state-of-the-art classification network VGG-16 [33] pre-trained for image classification on ILSVRC 2012 [35]. We use the layers up to CONV5, discarding all higher layers, as the basis of our network. We then append a free-form ROI pooling layer (Section 3.3), a region-to-pixel layer, a softmax layer and pixel-level loss (Section 3.2, Fig. 2c). To include local context, we combine region and entire bounding box using separate weights (Section 3.3).

Regions. We use Selective Search [29], which delivers three sets of region proposals, one per color space (RGB, HSV, LAB). During training we change the set of region proposals in each mini-batch to have a more diverse set of proposals without the additional overhead of having three times as many regions. We use region proposals with a minimum size of 100 pixels for SIFT Flow, and 400 pixels for PASCAL Context. This results in an average of 370 proposals for SIFT Flow and 150 proposals for PASCAL Context, for each of the three color spaces. Additionally we use all ground truth regions at training time. This is especially important for very small objects that are not tightly covered by region proposals.

Training. The network is trained using Stochastic Gradient Descent (SGD) with momentum. For 20 epochs we use a learning rate of 1e-3, followed by 10 epochs using learning rate 1e-4. All other SGD hyperparameters are taken from Fast R-CNN [27]. We use either an inverse-class frequency weighted loss (referred to as *balanced* below) or a natural frequency weighted loss (*unbalanced*).

Method	Year	Class Acc.	Global Acc.	Method	Year	Class Acc.	Global Acc.
Byeon [41]	2015	22.6	68.7	Sharma [11]	2014	48.0	79.6
Gould [45]	2014	25.7	78.4	Yang [16]	2014	48.7	79.8
Tighe [13]	2010	29.4	76.9	George [5]	2015	50.1	81.7
Pinheiro [25]	2014	30.0	76.5	Farabet [21]	2013	50.8	78.5
Gatta [46]	2014	32.1	78.7	Long [23]	2015	51.7	**85.2**
Singh [47]	2013	33.8	79.2	Sharma [12]	2015	52.8	80.9
Shuai [42]	2015	39.7	80.1	Caesar [2]	2015	55.6	-
Tighe [14]	2013	41.1	78.6	Eigen [20]	2015	55.7	**86.8**
Kekeç [40]	2014	45.8	70.4	Ours	2016	**64.0**	**84.3**

Table 1. *Evaluation on SIFT Flow test. We show results for our model trained for either a balanced or an unbalanced loss. We also show results of previous works, where we report the maximum result for each metric if multiple results are given.*

4.2 Main results

SIFT Flow. We compare our method to other works on SIFT Flow test in Table 1. We first compare in the balanced setting, which takes rare classes into account. Hence we train our model for the loss described in Section 3.4 and compare to methods using class-average accuracy. We achieve 64.0%, which substantially outperforms the previous state-of-the-art, including the fully convolutional method [20] by +8.3% and the region-based method [2] by +8.4%.

We also compare in the unbalanced setting using global accuracy, which mainly measures performance on common classes. Hence we train our model for the loss in Eq. (4). This yields a competitive 84.3% global accuracy, outperforming most previous methods, and coming close to the state-of-the-art [20] (86.8%).

PASCAL Context. We also evaluate our method on the recent PASCAL Context dataset [17]. In Table 2 we show the results using either a balanced or an unbalanced loss. Our balanced model achieves 49.9% class-average accuracy, outperforming the only work that reports results for that measure [23] by +3.4%. Our unbalanced model achieves competitive results on global accuracy (62.4%) and reasonable results on mean IOU (32.5%).

Qualitative analysis. Fig. 4 and 5 show example labelings generated by our method on SIFT Flow test and PASCAL Context validation. Notice how our method accurately adheres to object boundaries, such as buildings (Fig. 4e, 4h), birds (Fig. 5a, 5c) and boat (Fig. 5i). This is one of the advantages of using a region-based approach. Furthermore, our method correctly identifies small objects like pole (Fig. 4a) and the streetlight (Fig. 4d). This is facilitated by our method's ability to adaptively select the scale on which to do recognition. Finally, notice that our method sometimes even correctly labels parts of the image missing in the ground truth, such as fence (Fig. 4d) and cat whiskers (Fig. 5d).

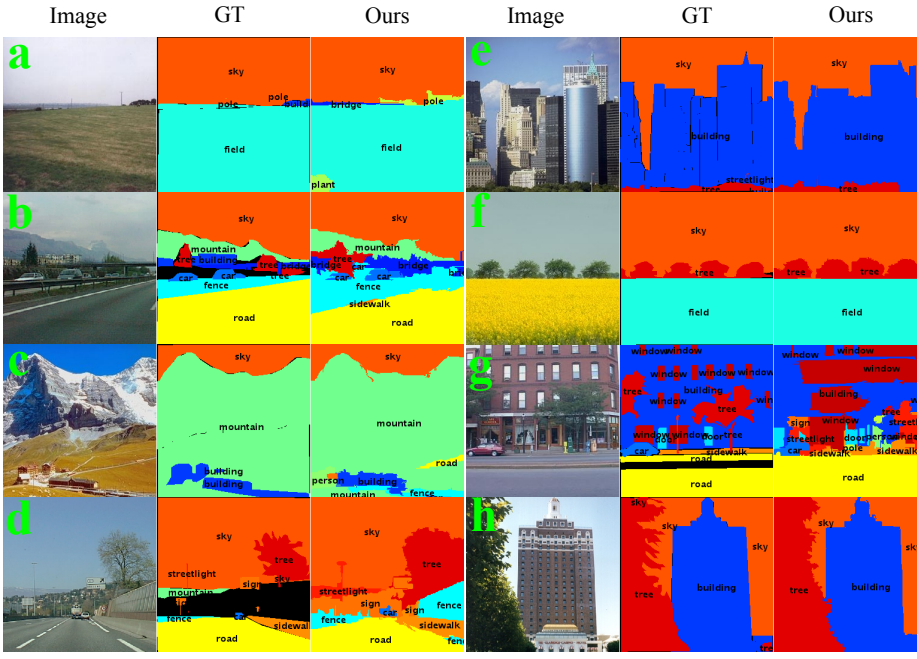

Fig. 4. *Example labelings on SIFT Flow test. We show an image, the ground truth labeling and the output of our balanced model.*

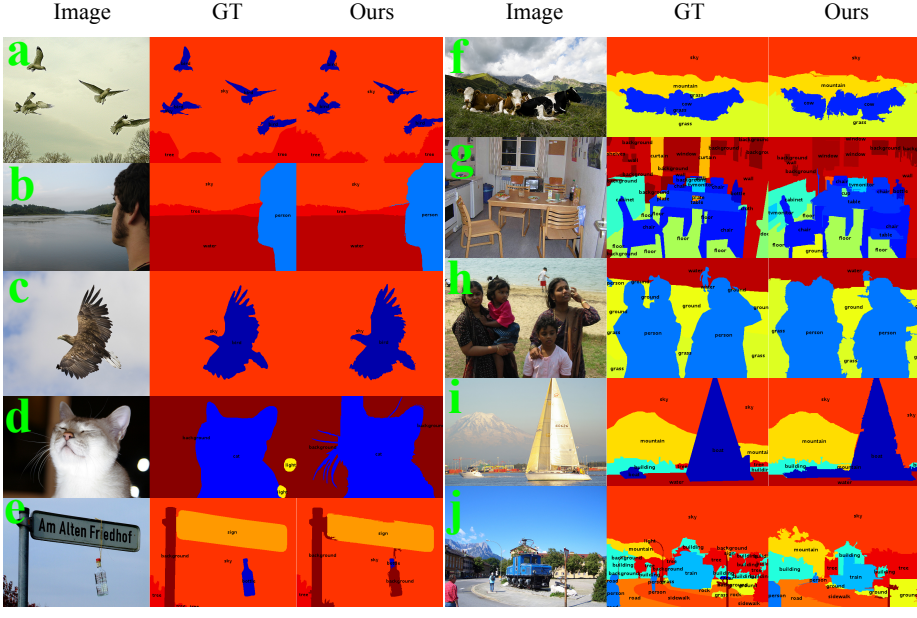

Fig. 5. *Example labelings on PASCAL Context validation. We show an image, the ground truth labeling and the output of our unbalanced model.*

Method	Year	Class Acc.	Global Acc.	Mean IOU
O2P [3]	2012	-	-	18.1
Dai et al. [4]	2015	-	-	34.4
Long et al. [23]	2015	46.5	**65.9**	35.1
Dai et al. [19]	2015	-	-	35.7
Zheng et al. [26]	2015	-	-	**39.3**
Dai et al. (add. boxes) [19]	2015	-	-	40.5
Ours	2016	**49.9**	62.4	32.5

Table 2. *Evaluation on PASCAL Context validation. We show results using a balanced and an unbalanced version of our method, as well as the current state-of-the art, where we always report the maximum result for each metric. O2P results are from the errata of Mottaghi et al. [17]. Dai et al. [19] train using additional bounding box annotations.*

	Boundaries	Full image
FCN-16s	37.9	49.3
Ours	57.3	64.0
difference	*+19.4*	*+14.7*
FCN-16s	34.0	48.1
Ours	38.9	49.9
difference	*+4.9*	*+1.8*

Table 3. *Class-average accuracy at object boundaries on SIFT Flow test (top) and PASCAL Context validation (bottom). Improvements on boundaries are consistently larger than on full images.*

ROI pooling		Class Acc.
bounding box		62.3
region		62.8
region + box	tied weights	63.4
region + box	separate weights	64.0
bounding box	purely rect.	59.3

Table 4. *Results on SIFT Flow test using free-form pooling, bounding box pooling or both. We also report results when regions are rectangular even in the region-to-pixel layer (purely rectangular).*

4.3 Extra analysis

Accuracy at object boundaries. Following [48, 49], we evaluate the performance on image pixels that are within 4 pixels of a ground truth object boundary. We compare our method to the MatConvNet [50] reimplementation of Fully Convolutional Networks (FCN) [23] in Table 3. On SIFT Flow test, FCN-16s obtains 37.9% class-average accuracy on boundaries, while our method gets to 57.3%. When evaluated on all pixels in the image, FCN-16s brings 49.3%, vs 64.0% by our method. Hence, our method is +19.4% better on boundaries and +14.7% on complete images. Analogously, on PASCAL Context we get +4.9% on boundaries and +1.8% on complete images. Since our improvements are consistently larger on object boundaries, we conclude that our method is especially good at capturing them, compared to the basic FCN architecture (Fig. 2a).

End-to-end training. Our region-to-pixel layer enables end-to-end training of region-based semantic segmentation models. We analyze how this end-to-end training influences performance, by comparing the baseline model (Fig. 2b) to our model (Fig. 2c). To isolate the effect of end-to-end training, in both models

we perform ROI pooling on the bounding box only. Hence all components of the two models are identical, apart from the region-to-pixel layer and the loss they are trained for. On SIFT Flow test the baseline model achieves a global accuracy of 60.9%, compared to our 83.7%. We conclude that end-to-end training yields considerable accuracy gains over the baseline architecture in Fig. 2b.

Softmax before max. Our application of the max before the softmax (Eq. 3) enables us to recognize each object at its appropriate scale (Sec. 3.2). However, using the softmax before the max (Eq. 1) yields an alternative model. Interestingly, on SIFT Flow test our proposed order outperforms the alternative by +8.7% class-average accuracy.

Importance of multi-scale regions. We argue that overlapping, multi-scale regions are important to unleash the full potential of region-based methods. To show this, we train and test our model with non-overlapping regions [51]. This yields 60.0% class-average accuracy on SIFT Flow test, which is below the results when using multi-scale overlapping regions (64.0% class-average accuracy).

Free-form versus bounding box representations. We analyze the influence of the different representations resulting from different ROI pooling methods (Sec. 3.3). Keeping all else constant, we compare (I) free-form ROI pooling, (II) bounding box ROI pooling, (III) their combination with tied weights and (IV) their combination with separate weights. Results are shown in Table 4.

Free-form representations perform +0.5% better than bounding box representations, demonstrating that focusing accurately on the object is better. Their combination does even better, yielding another +0.6% gain with tied weights (same number of model parameters) and +0.6% with separate weights. Hence both representations are complementary and best treated separately.

In all above experiments the region-to-pixel layer operates on free-form regions. To verify the importance of the free-form regions themselves, we perform an extra experiment using purely rectangular regions (both in the region-to-pixel layer and during ROI pooling). This lowers class-average accuracy by -4.7%, demonstrating the value of free-form regions.

5 Conclusion

We propose a region-based semantic segmentation model with an accompanying end-to-end training scheme based on a CNN architecture. This architecture combines the advantages of crisp object boundaries and adaptive, multi-scale representations found in region-based methods with end-to-end training directly optimized for semantic segmentation found in fully convolutional methods. We achieve this by introducing a differentiable region-to-pixel layer and a differentiable free-form ROI pooling layer. In terms of class-average pixel accuracy, our method outperforms the state-of-the-art on two datasets, achieving 49.9% on PASCAL Context and 64.0% on SIFT Flow.

Acknowledgements. Work supported by the ERC Starting Grant VisCul.

References

1. Boix, X., Gonfaus, J., van de Weijer, J., Bagdanov, A.D., Serrat, J., Gonzàlez, J.: Harmony potentials: Fusing global and local scale for semantic image segmentation. IJCV (2012)
2. Caesar, H., Uijlings, J., Ferrari, V.: Joint calibration for semantic segmentation. In: BMVC. (2015)
3. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: ECCV. (2012)
4. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: CVPR. (2015)
5. George, M.: Image parsing with a wide range of classes and scene-level context. In: CVPR. (2015)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
7. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV. (2014)
8. Li, F., Carreira, J., Lebanon, G., Sminchisescu, C.: Composite statistical inference for semantic segmentation. In: CVPR. (2013)
9. Mostajabi, M., Yadollahpour, P., Shakhnarovich, G.: Feedforward semantic segmentation with zoom-out features. In: CVPR. (2015)
10. Plath, N., Toussaint, M., Nakajima, S.: Multi-class image segmentation using conditional random fields and global classification. In: ICML. (2009)
11. Sharma, A., Tuzel, O., Liu, M.Y.: Recursive context propagation network for semantic scene labeling. In: NIPS. (2014)
12. Sharma, A., Tuzel, O., Jacobs, D.W.: Deep hierarchical parsing for semantic segmentation. In: CVPR. (2015)
13. Tighe, J., Lazebnik, S.: Superparsing: Scalable nonparametric image parsing with superpixels. In: ECCV. (2010)
14. Tighe, J., Lazebnik, S.: Finding things: Image parsing with regions and per-exemplar detectors. In: CVPR. (2013)
15. Tighe, J., Niethammer, M., Lazebnik, S.: Scene parsing with object instances and occlusion ordering. In: CVPR. (2014)
16. Yang, J., Price, B., Cohen, S., Yang, M.H.: Context driven scene parsing with attention to rare classes. In: CVPR. (2014)
17. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR. (2014)
18. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR. (2015)
19. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: ICCV. (2015)
20. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV. (2015)
21. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE Trans. on PAMI 35(8) (2013) 1915–1929
22. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR. (2015)

23. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
24. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: ICCV. (2015)
25. Pinheiro, P., Collobert, R.: Recurrent convolutional neural networks for scene parsing. In: ICML. (2014)
26. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: ICCV. (2015)
27. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
28. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: CVPR. (2010)
29. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV (2013)
30. Endres, I., Hoiem, D.: Category-independent object proposals with diverse ranking. IEEE Trans. on PAMI 36(2) (2014) 222–234
31. Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. (2014)
32. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
34. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. IJCV 81(1) (2009) 2–23
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L.: ImageNet large scale visual recognition challenge. IJCV (2015)
36. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014)
37. Bishop, C.: Neural networks for pattern recognition. Oxford University Press (1995)
38. Ripley, B.: Pattern recognition and neural networks. Cambridge University Press (1996)
39. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. (2014)
40. Kekeç, T., Emonet, R., Fromont, E., Trémeau, A., Wolf, C.: Contextually constrained deep networks for scene labeling. In: BMVC. (2014)
41. Byeon, W., Breuel, T.M., Raue, F., Liwicki, M.: Scene labeling with LSTM recurrent neural networks. In: CVPR. (2015)
42. Shuai, B., Wang, G., Zuo, Z., Wang, B., Zhao, L.: Integrating parametric and non-parametric models for scene labeling. In: CVPR. (2015)
43. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing via label transfer. IEEE Trans. on PAMI 33(12) (2011) 2368–2382
44. Everingham, M., Eslami, S., van Gool, L., Williams, C., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge: A retrospective. IJCV (2015)
45. Gould, S., Zhao, J., He, X., Zhang, Y.: Superpixel graph label transfer with learned distance metric. In: ECCV. (2014)
46. Gatta, C., Romero, A., van de Veijer, J.: Unrolling loopy top-down semantic feedback in convolutional deep networks. In: Workshop at CVPR. (2014)

47. Singh, G., Kosecka, J.: Nonparametric scene parsing with adaptive feature relevance and semantic context. In: CVPR. (2013)
48. Kohli, P., Ladicky, L., Torr, P.: Robust higher order potentials for enforcing label consistency. IJCV (2009)
49. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with gaussian edge potentials. In: NIPS. (2011)
50. Vedaldi, A., Lenc, K.: Matconvnet – convolutional neural networks for MATLAB. In: ACM Multimedia. (2015)
51. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)