Joint Calibration for Semantic Segmentation

Holger Caesar
holger.caesar@ed.ac.uk
Jasper Uijlings
jrr.uijlings@ed.ac.uk
Vittorio Ferrari
vittorio.ferrari@ed.ac.uk

School of Informatics
University of Edinburgh
Edinburgh, UK

Abstract

Semantic segmentation is the task of assigning a class-label to each pixel in an image. We propose a region-based semantic segmentation framework which handles both full and weak supervision, and addresses three common problems: (1) Objects occur at multiple scales and therefore we should use regions at multiple scales. However, these regions are overlapping which creates conflicting class predictions at the pixel-level. (2) Class frequencies are highly imbalanced in realistic datasets. (3) Each pixel can only be assigned to a single class, which creates competition between classes. We address all three problems with a joint calibration method which optimizes a multi-class loss defined over the final pixel-level output labeling, as opposed to simply region classification. Our method outperforms the state-of-the-art on the popular SIFT Flow [17] dataset in both the fully and weakly supervised setting.

1 Introduction

Semantic segmentation is the task of assigning a class label to each pixel in an image (Fig. 1). In the fully supervised setting, we have ground-truth labels for all pixels in the training images. In the weakly supervised setting, class-labels are only given at the image-level. We tackle both settings in a single framework which builds on region-based classification.

Our framework addresses three important problems common to region-based semantic segmentation. First of all, objects naturally occur at different scales within an image [3, 35]. Performing recognition at a single scale inevitably leads to regions covering only parts of an object which may have ambiguous appearance, such as *wheels* or *fur*, and to regions straddling over multiple objects, whose classification is harder due to their mixed appearance. Therefore many recent methods operate on pools of regions computed at multiple scales, which have a much better chance of containing some regions covering complete objects [3, 9, 10, 11, 15, 23, 28, 43]. However, this leads to overlapping regions which may lead to conflicting class predictions at the pixel-level. These conflicts need to be properly resolved.

Secondly, classes are often unbalanced [2, 7, 13, 18, 19, 28, 30, 33, 37, 39, 40, 41]: "cars" and "grass" are frequently found in images while "tricycles" and "gravel" are much rarer. Due to the nature of most classifiers, without careful consideration these rare classes are largely ignored: even if the class occurs in an image the system will rarely predict it.

Pages 29.1-29.13
DOI: https://dx.doi.org/10.5244/C.29.29

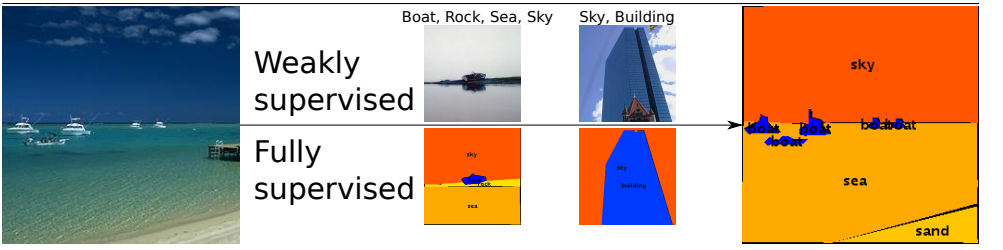

Figure 1: Semantic segmentation is the task of assigning class labels to all pixels in the image. During training, with full supervision we have ground-truth labels of all pixels. With weak supervision we only have labels at the image-level.

Since class-frequencies typically follow a power-law distribution, this problem becomes increasingly important with the modern trend towards larger datasets with more and more classes.

Finally, classes compete: a pixel can only be assigned to a single class (e.g. it can not belong to both "sky" and "airplane"). To properly resolve such competition, a semantic segmentation framework should take into account predictions for multiple classes jointly.

In this paper we address these three problems with a joint calibration method over an ensemble of SVMs, where the calibration parameters are optimized over all classes, and for the final evaluation criterion, i.e. the accuracy of pixel-level labeling, as opposed to simply region classification. While each SVM is trained for a single class, their joint calibration deals with the competition between classes. Furthermore, the criterion we optimize for explicitly accounts for class imbalance. Finally, competition between overlapping regions is resolved through maximization: each pixel is assigned the highest scoring class over all regions covering it. We jointly calibrate the SVMs for optimal pixel labeling *after* this maximization, which effectively takes into account conflict resolution between overlapping regions. Results on the SIFT Flow dataset [17] show that our framework outperforms the state-of-the-art in both the fully and the weakly supervised setting.

2 Related work

Early works on semantic segmentation used pixel- or patch-based features over which they define a Condition Random Field (CRF) [29, 36]. Many modern successful works use region-level representations, both in the fully supervised [1, 4, 9, 10, 11, 15, 19, 23, 28, 31, 32, 34, 41] and weakly supervised [37, 38, 39, 40, 42, 43] settings. A few recent works use CNNs to learn a direct mapping from image to pixel labels [7, 16, 20, 21, 22, 26, 27, 30, 44], although some of them [7, 27] use region-based post-processing to impose label smoothing and to better respect object boundaries. Other recent works use CRFs to refine the CNN pixel-level predictions [6, 16, 20, 26, 44]. In this work we focus on region-based semantic segmentation, which we discuss in light of the three problems raised in the introduction.

Overlapping regions. Traditionally, semantic segmentation systems use superpixels [1, 9, 19, 27, 31, 32, 34, 41], which are non-overlapping regions resulting from a single-scale oversegmentation. However, appearance-based recognition of superpixels is difficult as they typically capture only parts of objects, rather than complete objects. Therefore, many recent methods use overlapping multi-scale regions [3, 4, 10, 11, 15, 23, 43]. However, these may lead to conflicting class predictions at the pixel-level. Carreira et al. [4] address this simply by taking the maximum score over all regions containing a pixel. Both Hariharan et al. [11] and Girshick et al. [10] use non-maximum suppression, which may give problems

for nearby or interacting objects [15]. Li et al. [15] predict class overlap scores for each region at each scale. Then they create superpixels by intersecting all regions. Finally, they assign overlap scores to these superpixels using maximum composite likelihood (i.e. taking all multi-scale predictions into account). Plath et al. [23] use classification predictions over a segmentation hierarchy to induce label consistency between parent and child regions within a tree-based CRF framework. After solving their CRF formulation, only the smallest regions (i.e. leaf-nodes) are used for class prediction. In the weakly supervised setting, most works use superpixels [37, 38, 39, 40] and so do not encounter problems of conflicting predictions. Zhang et al. [42] use overlapping regions to enforce a form of class-label smoothing, but they all have the same scale. A different Zhang et al. [43] use overlapping region proposals at multiple scales in a CRF.

Class imbalance. As the PASCAL VOC dataset [6] is relatively balanced, most works that experiment on it did not explicitly address this issue [1, 4, 5, 10, 11, 15, 16, 18, 22, 23, 26, 44]. On highly imbalanced datasets such as SIFT Flow [17], Barcelona [31] and LM+SUN [33], rare classes pose a challenge. This is observed and addressed by Tighe et al. [33] and Yang et al. [41]: for a test image, only a few training images with similar context are used to provide class predictions, but for rare classes this constraint is relaxed and more training images are used. Vezhnevets et al. [37] balance rare classes by normalizing scores for each class to range $[0,1]$. A few works [19, 39, 40] balance classes by using an inverse class frequency weighted loss function.

Competing classes. Several works train one-vs-all classifiers separately and resolve labeling through maximization [4, 10, 11, 15, 19, 22, 23, 33]. This is suboptimal since the scores of different classes may not be properly calibrated. Instead, Tighe et al. [31, 33] and Yang et al. [41] use Nearest Neighbor classification which is inherently multi-class. In the weakly supervised setting appearance models are typically trained in isolation and remain uncalibrated [37, 37, 39, 40, 42]. To the best of our knowledge, Boix et al. [1] is the only work in semantic segmentation to perform joint calibration of SVMs. While this enables to handle competing classes, in their work they use non-overlapping regions. In contrast, in our work we use overlapping regions where conflicting predictions are resolved through maximization. In this setting, joint calibration is particularly important, as we will show in Sec. 4. As another difference, Boix et al. [1] address only full supervision whereas we address both full and weak supervision in a unified framework.

3 Method

3.1 Model

We represent an image by a set of overlapping regions [35] described by CNN features [10] (Sec. 3.4). Our semantic segmentation model infers the label o_p of each pixel p in an image:

$$o_p = \underset{c,\, r \ni p}{\arg\max}\ \sigma(w_c \cdot x_r,\ a_c, b_c) \tag{1}$$

As appearance models, we have a separate linear SVM w_c per class c. These SVMs score the features x_r of each region r. The scores are calibrated by a sigmoid function σ, with different parameters a_c, b_c for each class c. The arg max returns the class c with the highest score over all regions that contain pixel p. This involves maximizing over classes for a region, and over the regions that contain p.

During training we find the SVM parameters w_c (Sec. 3.2) and calibration parameters a_c and b_c (Sec. 3.3). The training of the calibration parameters takes into account the effects of the two maximization operations, as they are optimized for the output pixel-level labeling performance (as opposed to simply accuracy in terms of region classification).

3.2 SVM training

Fully supervised. In this setting we are given ground-truth pixel-level labels for all images in the training set (Fig. 1). This leads to a natural subdivision into ground-truth regions, i.e. non-overlapping regions perfectly covering a single class. We use these as positive training samples. However, such idealized samples are rarely encountered at test time since there we have only imperfect region proposals [35]. Therefore we use as additional positive samples for a class all region proposals which overlap heavily with a ground-truth region of that class (i.e. Intersection-over-Union greater than 50% [6]). As negative samples, we use all regions from all images that do not contain that class. In the SVM loss function we apply inverse frequency weighting in terms of the number of positive and negative samples.

Weakly supervised. In this setting we are only given image-level labels on the training images (Fig. 1). Hence, we treat region-level labels as latent variables which are updated using an alternated optimization process (as in [37, 38, 39, 40, 43]). To initialize the process, we use as positive samples for a class all regions in all images containing it. At each iteration we alternate between training SVMs based on the current region labeling and updating the labeling based on the current SVMs (by assigning to each region the label of the highest scoring class). In this process we keep our negative samples constant, i.e. all regions from all images that do not contain the target class. In the SVM loss function we apply inverse frequency weighting in terms of the number of positive and negative samples.

3.3 Joint Calibration

We now introduce our joint calibration procedure, which addresses three common problems in semantic segmentation: (1) conflicting predictions of overlapping regions, (2) class imbalance, and (3) competition between classes.

To better understand the problem caused by overlapping regions, consider the example of Fig. 2. It shows three overlapping regions, each with different class predictions. The final goal of semantic segmentation is to output a pixel-level labeling, which is evaluated in terms of pixel-level accuracy. In our framework we employ a winner-takes all principle: each pixel takes the class of the highest scored region which contains it. Now, using uncalibrated SVMs is problematic (second row in Fig. 2). SVMs are trained to predict class labels at the region-level, not the pixel-level. However, different regions have different area, and, most importantly, not all regions contribute all of their area to the final pixel-level labeling: Predictions of small regions may be completely suppressed by bigger regions (e.g. in Fig. 2, row 3, the inner-boat region is suppressed by the prediction of the complete boat). In other cases, bigger regions may be *partially* overwritten by smaller regions (e.g. in Fig. 2 the boat region partially overwrites the prediction of the larger boat+sky region). Furthermore, the SVMs are trained in a one-vs-all manner and are unaware of other classes. Hence they are unlikely to properly resolve competition between classes even within a single region. The problems above show that without calibration, the SVMs are optimized for the wrong criterion. We propose to jointly calibrate SVMs for the correct criterion, which corresponds better to the evaluation measure typically used for semantic segmentation (i.e. pixel labeling

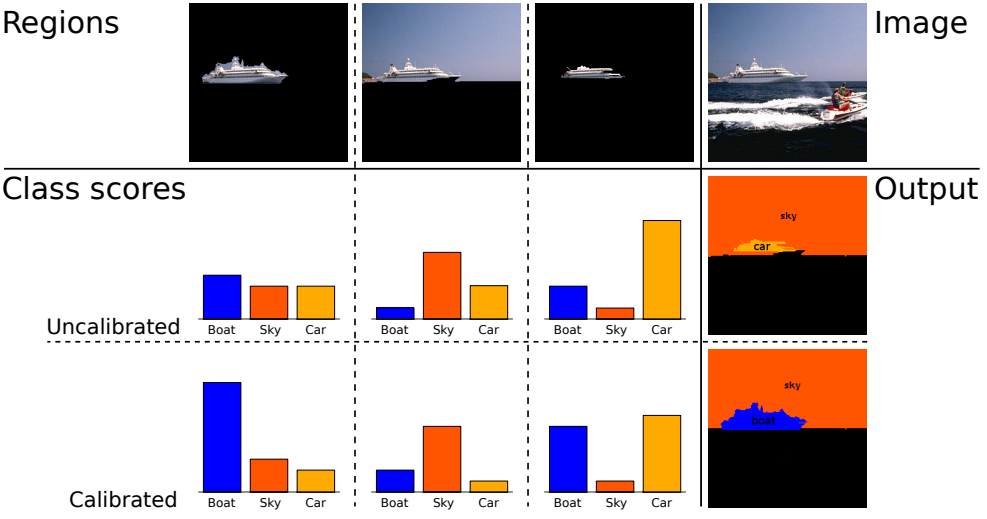

Figure 2: The first row shows multiple region proposals (left) extracted from an image (right). The following rows show the per-class SVM scores of each region (left) and the pixel-level labeling (right). Row 2 shows the results before and row 3 after joint calibration.

accuracy averaged over classes). We do this by applying sigmoid functions σ to all SVM outputs:

$$\sigma(w_c \cdot x_r,\, a_c, b_c) = (1 + \exp(a_c \cdot w_c \cdot x_r + b_c))^{-1} \tag{2}$$

where a_c, b_c are the calibration parameters for class c. We calibrate the parameters of all classes jointly by minimizing a loss function $\mathcal{L}(o,l)$, where o is the pixel labeling output of our method on the full training set ($o = \{o_p;\, p = 1 \ldots P\}$) and l the ground-truth labeling.

We emphasize that the pixel labeling output o is the result *after* the maximization over classes and regions in Eq. (1). Since we optimize for the accuracy of this final output labeling, and we do so jointly over classes, our calibration procedure takes into account both problems of conflicting class predictions between overlapping regions and competition between classes. Moreover, we also address the problem of class imbalance, as we compensate for it in our loss functions below.

Fully supervised loss. In this setting our loss directly evaluates the desired performance measure, which is typically pixel labeling accuracy averaged over classes [7, 18, 27, 31, 41]

$$\mathcal{L}(o,l) = 1 - \frac{1}{C} \sum_{c=1}^{C} \frac{1}{P_c} \sum_{p;l_p=c} [l_p = o_p] \tag{3}$$

where l_p is the ground-truth label of pixel p, o_p is the output pixel label, P_c is the number of pixels with ground-truth label c, and C is the number of classes. $[\cdot]$ is 1 if the condition is true and 0 otherwise. The inverse frequency weighting factor $1/P_c$ deals with class imbalance.

Weakly supervised loss. Also in this setting the performance measure is typically class-average pixel accuracy [37, 38, 40, 43]. Since we do not have ground-truth pixel labels, we cannot directly evaluate it. We do however have a set of ground-truth image labels l_i which we can compare against. We first aggregate the output pixel labels o_p over each image m_i into output image labels $o_i = \cup_{p \in m_i} o_p$. Then we define as loss the difference between

the ground-truth label set l_i and the output label set o_i, measured by the Hamming distance between their binary vector representations

$$\mathcal{L}(o,l) = \sum_{i=1}^{I} \sum_{c=1}^{C} \frac{1}{I_c} |l_{i,c} - o_{i,c}| \tag{4}$$

where $l_{i,c} = 1$ if label c is in l_i, and 0 otherwise (analog for $o_{i,c}$). I is the total number of training images. I_c is the number of images having ground-truth label c, so the loss is weighted by the inverse frequency of class labels, measured at the image-level. Note how also in this setting the loss looks at performance after the maximization over classes and regions (Eq. (1)).

Optimization. We want to minimize our loss functions over the calibration parameters a_c, b_c of all classes. This is hard, because the output pixel labels o_p depend on these parameters in a complex manner due to the max over classes and regions in Eq. (1), and because of the set-union aggregation in the case of the weakly supervised loss. Therefore, we apply an approximate minimization algorithm based on coordinate descent. Coordinate descent is different from gradient descent in that it can be used on arbitrary loss functions that are not differentiable, as it only requires their evaluation for a given setting of parameters.

Coordinate descent iteratively applies line search to optimize the loss over a single parameter at a time, keeping all others fixed. This process cycles through all parameters until convergence. As initialization we use constant values ($a_c = -7$, $b_c = 0$). During line search we consider 10 equally spaced values (a_c in $[-12,-2]$, b_c in $[-10,10]$).

This procedure is guaranteed to converge to a local minimum on the search grid. While this might not be the global optimum, in repeated trials we found the results to be rather insensitive to initialization. Furthermore, in our experiments the number of iterations was roughly proportional to the number of parameters.

Efficient evaluation. On a typical training set with $C = 30$ classes, our joint calibration procedure evaluates the loss thousands of times. Hence, it is important to evaluate pixel-level accuracy quickly. As the model involves a maximum over classes and a maximum over regions at every pixel, a naive per-pixel implementation would be prohibitively expensive. Instead, we propose an efficient technique that exploits the nature of the Selective Search region proposals [35], which form a bottom-up hierarchy starting from superpixels. As shown in Fig. 3, we start from the region proposal that contains the entire image (root node). Then we propagate the maximum score over all classes down the region hierarchy. Eventually we assign to each superpixel (leaf nodes) the label with the highest score over all regions that contain it. This label is assigned to all pixels in the superpixel. To compute class-average pixel accuracy, we normally need to compare each pixel label to the ground-truth label. However since we assign the same label to all pixels in a superpixel, we can precompute the ground-truth label distribution for each superpixel and use it as a lookup table. This reduces the runtime complexity for an image from $O(P_i \cdot R_i \cdot C)$ to $O(R_i \cdot C)$, where P_i and R_i are the number of pixels and regions in an image respectively, and C is the number of classes.

Why no Platt scaling. At this point the reader may wonder why we do not simply use Platt scaling [24] as is commonly done in many applications. Platt scaling is used to convert SVM scores to range $[0,1]$ using sigmoid functions, as in Eq. (2). However, in Platt scaling the parameters a_c, b_c are optimized for each class in isolation, ignoring class competition. The loss function $\mathcal{L}_c$ in Platt scaling is the cross-entropy function

$$\mathcal{L}_c(\sigma_c, l) = -\sum_r t_{r,c} \log(\sigma_c(x_r)) + (1 - t_{r,c}) \log(1 - \sigma_c(x_r)) \tag{5}$$

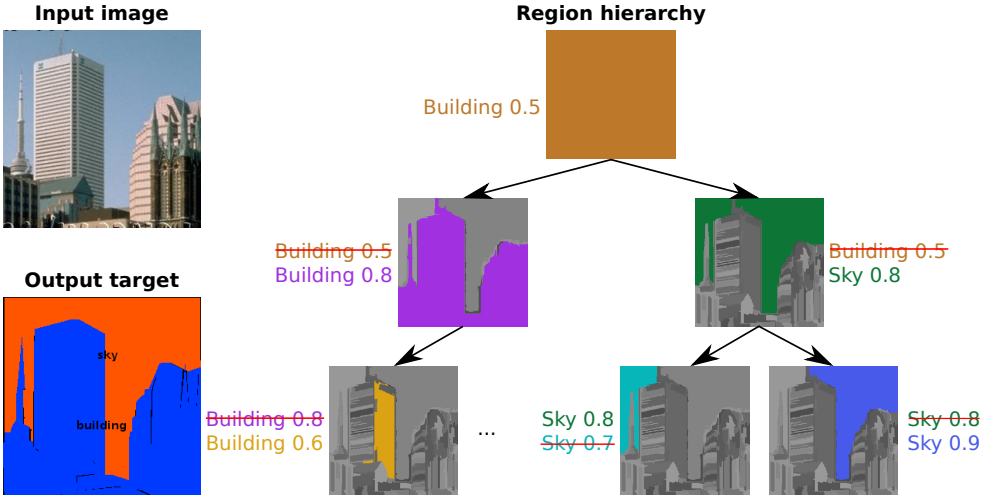

Figure 3: Our efficient evaluation algorithm uses the bottom-up structure of Selective Search region proposals to simplify the spatial maximization. We start from the root node and propagate the maximum score with its corresponding label down the tree. We label the image based on the labels of its superpixels (leaf nodes).

where N_+ is the number of positive samples, N_- the number of negative samples, and $t_{r,c} = \frac{N_+ + 1}{N_+ + 2}$ if $l_r = c$ or $t_{r,c} = \frac{1}{N_- + 2}$ otherwise; l_r is the region-level label. This loss function is inappropriate for semantic segmentation because it is defined in terms of accuracy of training samples, which are regions, rather than in terms of the final pixel-level accuracy. Hence it ignores the problem of overlapping regions. There is also no inverse frequency term to deal with class imbalance. We experimentally compare our method with Platt scaling in Sec. 4.

3.4 Implementation Details

Region proposals. We use Selective Search [35] region proposals using a subset of the "Fast" mode: we keep the similarity measures, but we restrict the scale parameter k to 100 and the color-space to RGB. This leads to two bottom-up hierarchies of one initial oversegmentation [8].

Features. We compute R-CNN features [10] using the Caffe implementation [12] of AlexNet [14]. Regions are described using all pixels in a tight bounding box, warped to a square image, and fed to the CNN. Since regions are free-form, Girshick et al. [10] additionally proposes to set pixels not belonging to the region to zero (i.e. not affecting the convolution). However, in our experiments this did not improve results so we do not use it.

For the weakly supervised setting we use the CNN network pre-trained for image classification on ILSVRC 2012 [25]. For the fully supervised setting we start from this network and finetune it on the training set of SIFT Flow [17]; the semantic segmentation dataset we experiment on. For both settings, following [10] we use the output of the 6th layer of the network as features.

SVM training. Like [10] we set the regularization parameter C to a fixed value in all our experiments. The SVMs minimize the L2-loss for region classification. We use hard-negative mining to reduce the memory consumption of our system.

4 Experiments

Datasets. We evaluate our method on the challenging SIFT Flow dataset [17]. It consists of 2488 training and 200 test images, pixel-wise annotated with 33 class labels. The class distribution is highly imbalanced in terms of overall region count as well as pixel count. As evaluation measure we use the popular class-average pixel accuracy [7, 18, 21, 28, 31, 34, 38, 40, 41, 43]. For both supervision settings we report results on the test set.

Fully supervised setting. Table 1 evaluates various versions of our model in the fully supervised setting, and compares to other works on the SIFT Flow dataset. Using uncalibrated SVMs, our model achieves a class-average pixel accuracy of 28.7%. If we calibrate the SVM scores with traditional Platt scaling results do not improve (27.7%). Using our proposed joint calibration to maximize class-average pixel accuracy improves results substantially to 55.6%. This shows the importance of joint calibration to resolve conflicts between overlapping regions at multiple scales, to take into account competition between classes, and generally to optimize a loss mirroring the evaluation measure. Fig. 4 (column "SVM") shows that larger background regions (i.e. road, building) swallow smaller foreground regions (i.e. person, awning). Many of these small objects become visible after calibration (column "SVM+JC"). This issue is particularly evident when working with overlapping regions. Consider a large region on a building which contains a awning. As the surface of the awning is small, the features of the large region will be dominated by the building, leading to strong classification score for the 'building' class. When these are higher than the classification score for 'awning' on the small awning region, the latter gets overwritten. Instead, this problem does not appear when working with superpixels [1]. A superpixel is either part of the building or part of the awning, so a high scoring awning superpixel cannot be overwritten by neighboring building superpixels. Hence, joint calibration is particularly important when working with overlapping regions. Our complete model outperforms the state-of-the-art [28] by 2.8%. For comparison we show the results of [33] (column "Tighe et al.").

Weakly supervised setting. Table 1 shows results in the weakly supervised setting. The model with uncalibrated SVMs achieves an accuracy of 21.2%. Using traditional Platt scaling the result is 16.8%, again showing it is not appropriate for semantic segmentation. Instead, our joint calibration almost doubles accuracy (37.4%). Fig. 5 illustrates the power of our weakly supervised method. Again rare classes appear only after joint calibration. Our complete model outperforms the state-of-the-art [40] in this setting by 2.4%. Xu et al. [40] additionally report results on the transductive setting (41.4%), where all (unlabeled) test images are given to the algorithm during training.

Region proposals. To demonstrate the importance of multi-scale regions, we also analyze oversegmentations that do not cover multiple scales. To this end, we keep our framework the same, but instead of Selective Search (SS) [35] region proposals we used a single oversegmentation using the method of Felzenszwalb and Huttenlocher (FH) [8] (for which we optimized the scale parameter). As Table 2 shows, SS regions outperform FH regions by a good margin of 12.2% in the fully supervised setting. This confirms that overlapping multi-scale regions are superior to non-overlapping oversegmentations.

CNN finetuning. As described in 3.4 we finetune our network for detection in the fully supervised case. Table 3 shows that this improves results by 6.2% compared to using a CNN trained only for image classification on ILSVRC 2012.

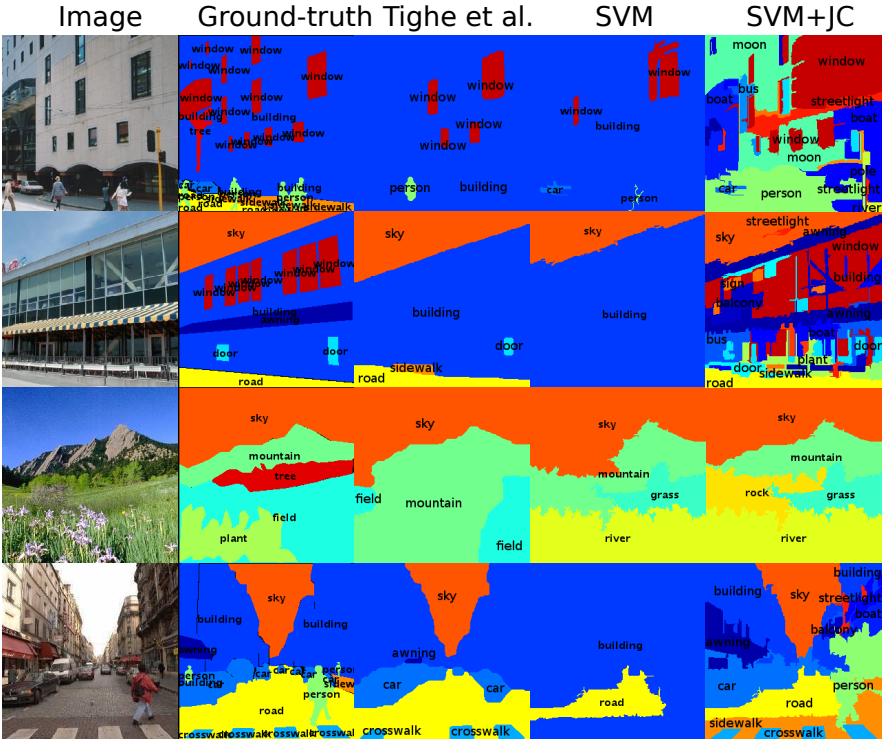

Figure 4: Fully supervised semantic segmentation on SIFT Flow. We present uncalibrated SVM results (SVM), jointly calibrated results (SVM+JC) and the results of Tighe et al. [33].

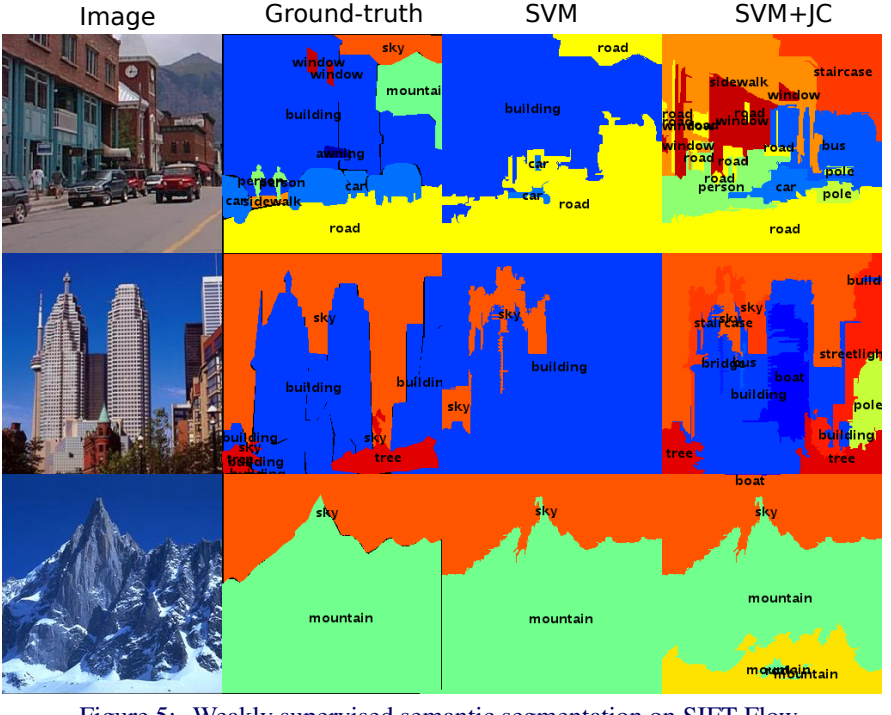

Figure 5: Weakly supervised semantic segmentation on SIFT Flow.

Method	Class Acc.
Byeon et al. [2]	22.6%
Tighe et al. [31]	29.1%
Pinheiro et al. [21]	30.0%
Shuai et al. [30]	39.7%
Tighe et al. [33]	41.1%
Kekeç et al. [13]	45.8%
Sharma et al. [27]	48.0%
Yang et al. [41]	48.7%
George et al. [9]	50.1%
Farabet et al. [7]	50.8%
Long et al. [18]	51.7%
Sharma et al. [28]	52.8%
Ours SVM	28.7%
Ours SVM+PS	27.7%
Ours SVM+JC	**55.6%**

Method	Class Acc.
Vezhnevets et al. [37]	14.0%
Vezhnevets et al. [38]	21.0%
Zhang et al. [42]	27.7%
Xu et al. [39]	27.9%
Zhang et al. [43]	32.3%
Xu et al. [40]	35.0%
Xu et al. (transductive) [40]	41.4%
Ours SVM	21.2%
Ours SVM+PS	16.8%
Ours SVM+JC	**37.4%**

Table 1: Class-average pixel accuracy in the fully supervised (left) and the weakly supervised setting (right) setting. We show results for our model on the test set of SIFT Flow using uncalibrated SVM scores (SVM), traditional Platt scaling (PS) and joint calibration (JC).

Regions	Class Acc.
FH [8]	43.4%
SS [35]	55.6%

Table 2: Comparison of single-scale (FH) and multi-scale (SS) regions.

Finetuned	Class Acc.
no	49.4%
yes	55.6%

Table 3: Effect of CNN finetuning for fully supervised semantic segmentation.

5 Conclusion

We addressed three common problems in semantic segmentation based on region proposals: (1) overlapping regions yield conflicting class predictions at the pixel-level; (2) class-imbalance leads to classifiers unable to detect rare classes; (3) one-vs-all classifiers do not take into account competition between multiple classes. We proposed a joint calibration strategy which optimizes a loss defined over the final pixel-level output labeling of the model, after maximization over classes and regions. This tackles all three problems: *joint* calibration deals with multi-class predictions, while our loss explicitly deals with class imbalance and is defined in terms of pixel-wise labeling rather than region classification accuracy. As a result we take into account conflict resolution between overlapping regions. Results show that our method outperforms the state-of-the-art in both the fully and the weakly supervised setting on the popular SIFT Flow [17] benchmark.

Acknowledgements. This work was supported by the ERC Starting Grant VisCul.

References

[1] X. Boix, J. Gonfaus, J. van de Weijer, A. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials: Fusing global and local scale for semantic image segmentation. *IJCV*, 2012.

[2] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *CVPR*, 2015.

[3] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010.

[4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[7] C. Farabet, C. Couprie, and L. Najman. Learning hierarchical features for scene labeling. *IEEE Trans. on PAMI*, 35(8):1915–1929, 2013.

[8] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[9] M. George. Image parsing with a wide range of classes and scene-level context. In *CVPR*, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[12] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[13] T. Kekeç, R. Emonet, E. Fromont, A. Trémeau, and C. Wolf. Contextually constrained deep networks for scene labeling. In *BMVC*, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[15] F. Li, J. Carreira, G. Lebanon, and C. Sminchisescu. Composite statistical inference for semantic segmentation. In *CVPR*, 2013.

[16] G. Lin, C. Shen, I. Reid, and A. van dan Hengel. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015.

[17] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. on PAMI*, 33(12):2368–2382, 2011.

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[19] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015.

[20] G. Papandreou, L. Chen, K. Murphy, and A. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015.

[21] P. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene parsing. In *ICML*, 2012.

[22] P. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[23] N. Plath, M. Toussaint, and S. Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *ICML*, 2009.

[24] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 1999.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. doi: 10.1007/s11263-015-0816-y.

[26] A. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015.

[27] A. Sharma, O. Tuzel, and M. Liu. Recursive context propagation network for semantic scene labeling. In *NIPS*, pages 2447–2455, 2014.

[28] A. Sharma, O. Tuzel, and D. W. Jacobs. Deep hierarchical parsing for semantic segmentation. In *CVPR*, 2015.

[29] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. *IJCV*, 81(1):2–23, 2009.

[30] B. Shuai, G. Wang, Z. Zuo, B. Wang, and L. Zhao. Integrating parametric and non-parametric models for scene labeling. In *CVPR*, 2015.

[31] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[32] J. Tighe and S. Lazebnik. Understanding scenes on many levels. In *ICCV*, 2011.

[33] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013.

[34] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014.

[35] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[36] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *CVPR*, 2007.

[37] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised semantic segmentation with multi image model. In *ICCV*, 2011.

[38] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *CVPR*, 2012.

[39] J. Xu, A. Schwing, and R. Urtasun. Tell me what you see and i will show you where it is. In *CVPR*, 2014.

[40] J. Xu, A. G. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015.

[41] J. Yang, B. Price, S. Cohen, and Y. Ming-Hsuan. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014.

[42] L. Zhang, Y. Gao, Y. Xia, K. Lu, J. Shen, and R. Ji. Representative discovery of structure cues for weakly-supervised image segmentation. In *IEEE Transactions on Multimedia*, volume 16, pages 470–479, 2014.

[43] W. Zhang, S. Zeng, D. Wang, and X. Xue. Weakly supervised semantic segmentation for social images. In *CVPR*, 2015.

[44] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, and Z. Su. Conditional random fields as recurrent neural networks. *arXiv preprint arXiv:1502.032405*, 2015.