Weakly-Supervised Semantic Segmentation by Redistributing Region Scores Back to the Pixels

Josip Krapac(✉) and Siniša Šegvić

Faculty of Electrical Engineering and Computing,
University of Zagreb, Zagreb, Croatia
josip.krapac@fer.hr

Abstract. We address the problem of semantic segmentation of objects in weakly supervised setting, when only image-wide labels are available. We describe an image with a set of pre-trained convolutional features and embed this set into a Fisher vector. We apply the learned image classifier on the set of all image regions and propagate the region scores back to the pixels. Compared to the alternatives the proposed method is simple, fast in inference, and especially in training. The method displays very good performance of on two standard semantic segmentation benchmarks.

1 Introduction

Semantic segmentation is one of the most challenging computer vision tasks with wide range of applications in scene and image understanding, class-specific attention, robot perception and autonomous navigation and planning. The goal of semantic segmentation is to assign a label to each pixel, where the label corresponds to an object class, *e.g.* "cat", "sofa" or "person". Semantic segmentation is difficult because the set of semantic concepts is very diverse, and objects may be located across a wide range of scales and poses. The segmentation model training in strongly supervised setting assumes that each image pixel is accompanied with its target class label. At test time the learned model is used to infer class labels for each pixel in a given input image. The main drawback of the strongly-supervised approach is the need for pixel-level annotations, because the acquisition of such precise labels is costly and requires substantial effort and time. A complex image may require more than 15 min of human attention. The best results are obtained with models based on deep convolutional nets which are known to be especially data hungry.

This is one of the main reasons for recent interest in approaches that relax the annotation effort [13–15,17,26]. These can be grouped into two main categories: weakly- and semi-supervised approaches. Weakly-supervised approaches learn the classification models without any pixel-level training data, which can be done by relying on bounding-box and image-wide annotations. Semi-supervised approaches assume that the bulk of the training data is weakly-supervised or unsupervised while a small part of the training dataset is annotated on the pixel-level. Due to the need to leverage weakly-supervised data, each semi-supervised

B. Rosenhahn and B. Andres (Eds.): GCPR 2016, LNCS 9796, pp. 377–388, 2016.
DOI: 10.1007/978-3-319-45886-1_31

approach is typically built on top of a weakly-supervised engine. This justifies further research in weakly-supervised approaches even though semi-supervised approaches are able to offer better performance.

In this paper we are concerned with semantic segmentation in the weakly-supervised setting, where we have access only to image-wide labels. The training phase requires only images annotated with image-wide labels, while at test time one should predict a class label for each pixel. Our method relies on a single per-class hyper-parameter $m(c)$ which modulates the extent of background in the processed images. The recent surge of interest for this very challenging problem is motivated by excellent performance of convolutional neural networks on related computer vision tasks: image categorization [10], object detection [19] and strongly-supervised semantic segmentation [12]. Convnets have significantly improved state-of-the-art in weakly-supervised semantic segmentation, but methods that employ them require either large amounts of weakly-annotated images [17] or suffer from computationally complex training [13] and inference [26].

We propose a simple yet very effective method for weakly-supervised semantic segmentation. The method is based on Fisher vector embedding of pre-trained convolutional features and linear classifiers learned from image-wide labels. We apply the learned classifier to all image regions, and employ a novel method to aggregate region-level scores into pixel-level decisions. To determine the class-specific hyper-parameters for the model we use a few tens of bounding box annotations per object class. The method requires only a hundreds of weakly-annotated images and a few tens bounding box annotations per class and displays fair performance and fast execution. The results are competitive compared to state-of-the-art methods as displayed by performance on standard and challenging semantic segmentation benchmarks.

2 Related Work

Most approaches to semantic segmentation operate in a strongly supervised context where training images are densely annotated on the pixel-level [2,5,21]. Impressive results in this context have recently been obtained using fully convolutional neural networks [12]. However, strongly-supervised approaches require pixel-level labels that are costly to obtain. Much recent interest has therefore been directed towards relaxing the annotation effort. This work can be grouped into two main categories: weakly-supervised approaches [13–15,17,26] and semi-supervised approaches [9,13,14,24]. Weakly supervised approaches relax the extent of supervision from ground-truth segmentation masks to image-level [13,14,17] or box-level [6,13,14,24] labels.

Much semantic segmentation work relies on bottom-up segmentation, since similar neighboring pixels are likely to share the common class. Some of these approaches score the segments indirectly, by averaging pixel-based evidence, and redistribute the scores back to the pixels [5,26]. This often improves the results by regularizing the semantic segmentation and aligning it with the natural image boundaries. However, most recent bottom-up segmentation approaches have been trained on pixel-level annotations [1,23], which makes the weakly-supervised qualification questionable.

Many recent approaches improve weakly supervised segmentation results by fitting various hyper-parameters. For example, one can set the relative size of the objects with respect to the background [13,14], or the inherent difficulty of the particular class (per-class thresholds of pixel scores [17]). Setting these hyper-parameters requires cross-validation on a pixel-level annotated validation set [17] or an insight into inherent dataset bias [13,14].

Most semantic segmentation approaches smooth the produced segmentation masks with some kind of a conditional random field (CRF) [13,14,26]. The CRF parameters can be learned from the image-wide labels [26] or from a small held-out set of fully-annotated images [13].

Advanced weakly-supervised approaches train semantic segmentation models exclusively with image-wide labels [13–15,17,26]. One way to tackle this problem is to leverage the information recovered with a multi-label image classification model [17]. Fisher vector image representation [20] offers interesting opportunities along these lines due to capability to relate an image-wide classification model to the contributions at the patch-level [5,25]. Applications of Fisher vectors to semantic segmentation have so for been researched only in the strongly-supervised setting and with SIFT features [5]. Here, following [3], we use Fisher vectors in conjunction with convolutional features and use the learned object classification models for weakly-supervised semantic segmentation.

3 Method

Here we describe the proposed method for weakly-supervised semantic segmentation. First, we explain the Fisher vector embedding and the advantages over embedding given by deep neural nets. Next, we describe how to infer pixel-level predictions from a set of region scores obtained by applying the learned classifier to all image regions. Finally, we show how to convert multi-label pixel predictions to multi-class pixel predictions.

3.1 Learning the Image Classifier

The task of image classifier is to learn the mapping from a set of images $\mathcal{X}$ into a set of class labels $\mathcal{Y}$. This mapping is learned from a set of images associated with class labels $\{(\boldsymbol{X}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$. Each image is represented by a set of local descriptors $\boldsymbol{X}_i = \{\boldsymbol{x}_{i,1}, \cdots, \boldsymbol{x}_{i,N_i}\}$, *e.g.* convolutional features extracted from an inner layer of convolutional net. In the remainder of the paper we call *patch* the central part of convolutional feature's receptive field. Label vector $\boldsymbol{y}_i$ has 1 on positions corresponding to the classes that are present in the image. We assume a general multi-label setting where labels are not mutually exclusive, so $\boldsymbol{y}_i$ can have more than one non-zero entry. Recently it has been shown that Fisher vector embedding of image represented with a set of convolutional features in conjunction with linear classifiers yields very good classification performance, comparable to the performance of embedding produced by a fully-connected (FC) part of the net [3]. In this approach a patch descriptor $\boldsymbol{x}$ is first explicitly embedded in a higher-dimensional space defined by the parameters of a pre-trained generative model via function $\varPhi$. Next, all patch embeddings are averaged by spatial pooling into the global image representation $\varPhi(\boldsymbol{X}_i)$:

$$\varPhi(\boldsymbol{X}_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} \varPhi(\boldsymbol{x}_{ij}), \tag{1}$$

where N_i is the number of patches in the image $\boldsymbol{X}_i$. The additivity of patch Fisher vectors along with linearity of the learned one-vs-all classifier entails the additivity of patch scores $s(c|\cdot)$ for class c:

$$s(c|\boldsymbol{X}_i) = \boldsymbol{w}_c^\top \varPhi(\boldsymbol{X}_i) = \boldsymbol{w}_c^\top \sum_{j=1}^{N_i} \varPhi(\boldsymbol{x}_{ij}) = \sum_{j=1}^{N_i} s(c|\boldsymbol{x}_{ij}). \tag{2}$$

Thus the score of an image $s(c|\boldsymbol{X})$ is given as the sum of patch scores. This allows us to determine the contribution of each patch to the image score, and therefore propagate the image score back to the patches, which is the main advantage of Fisher vector embedding over the FC part of the net. We use a logistic regression classifier so the posterior for a class is $p_c(c = 1|\boldsymbol{X}_i) = \sigma(s(c|\boldsymbol{X}_i))$. Note that we do not use improved Fisher vector [16], since preliminary experiments have shown only a marginal classification performance improvement, and using non-linear normalizations would break linearity of the image score *w.r.t.* patch scores.

3.2 From Image-Level Predictions to Pixel-Level Predictions

The classifier learned on the full images is then applied to image regions. We consider two settings: in the first setting regions correspond to patches, while in the second setting we consider all rectangular regions in the image. We first normalize region descriptor by the number of patches the region contains, and then apply the classifier learned on full images. This classifier score is computed at the resolution of convolutional feature maps and upsampled using nearest neighbor interpolation to the resolution of the full image. In the first setting the classifier learned on images is applied to patches. In case of general object classes displayed against complex backgrounds this setting is not likely to produce good results. In the second setting one has to arrive at patch scores from a set of region scores. We do this by considering the scores of all regions that contain the patch. To efficiently compute the scores for all image regions and all classifiers we use integral images, as in [11]. The score for each pixel is then computed as:

$$p_c(c = 1|\boldsymbol{x}, m) = \prod_{k=1}^{M} p_c(c = 1|\boldsymbol{Z}_k)^{\frac{m}{M}} \tag{3}$$

where $\boldsymbol{Z}_i$ is the Fisher vector of the i^{th} out of M regions that contain image patch described by $\boldsymbol{x}$. The scoring is illustrated in Fig. 1. When $m = M$ we

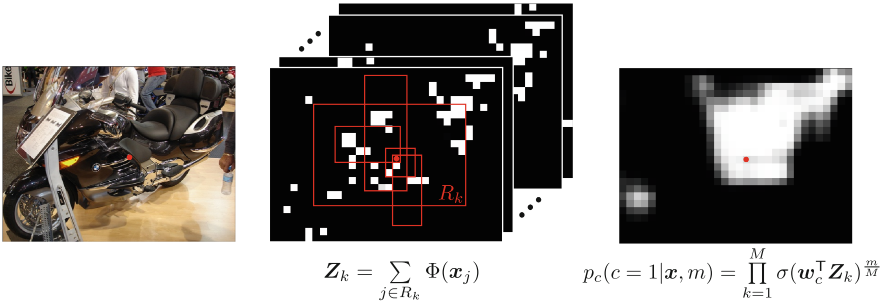
R_k
$\boldsymbol{Z}_k = \sum_{j \in R_k} \Phi(\boldsymbol{x}_j)$
$p_c(c=1|\boldsymbol{x},m) = \prod_{k=1}^{M} \sigma(\boldsymbol{w}_c^\top \boldsymbol{Z}_k)^{\frac{m}{M}}$

Fig. 1. Illustration of pixel scoring based on the regions that contain it. First the convolutional features $\boldsymbol{x}$ are extracted and embedded in the Fisher vector $\Phi(\boldsymbol{x})$. Then, Fisher vector is computed for each R_k that contains location of feature $\boldsymbol{x}$, using integral images. Finally, the class prediction for the pixel at location of feature $\boldsymbol{x}$ is computed by combining the predictions of all regions that contain it.

assume that the region descriptors are independent, which is clearly invalid as region overlap contains at least region descriptor $\boldsymbol{x}$, so we alleviate this by setting $m < M$. For smaller objects we would like to put more weight on independent decisions of regions, which means that we expect that higher m yields better performance. For larger objects we would like to put more weight on the interplay of overlapping regions, which means that we expect that smaller m yields better performance. In the remainder of the section we leave out dependence on m.

3.3 From Multi-label to Multi-class Pixel Predictions

Class posteriors that we propagate from an image to the pixels are based on learned one-vs-all classifiers, which are learned in multi-label setting, so in general $\sum_{c=1}^{C} p_c(c=1|\boldsymbol{x}) \neq 1$. On the other hand, the semantic segmentation is a multi-class problem since each pixel belongs to only one semantic class. Thus, we need to arrive at probability distribution over $C+1$ class labels for each pixel. To this end we couple the predictions of one-vs-all classifiers by first determining the probability that a pixel belongs to the background via noisy-and model [18]:

$$p_b(b=1|\boldsymbol{x}) = \prod_{c=1}^{C} 1 - p_c(c=1|\boldsymbol{x}), \tag{4}$$

and then we normalize the probabilities to obtain class posteriors for each pixel:

$$p_C(C=c|\boldsymbol{x}) = \frac{p_c(c=1|\boldsymbol{x})}{\sum\limits_{c=1..C,b} p_c(c=1|\boldsymbol{x})} \tag{5}$$

The use of noisy-and model discourages assignment of a pixel to the background if any of one-vs-all classifiers has high class posterior. In other words, pixel can have high background posterior $p_b(b = 1|\boldsymbol{x})$ only if all C classifiers have low $p_c(c = 1|\boldsymbol{x})$. Finally, we determine the class for each pixel using MAP:

$$\hat{c}(\boldsymbol{x}) = \underset{c=\{1..C,b\}}{\arg\max}\, p_C(C = c|\boldsymbol{x}) \tag{6}$$

4 Experimental Evaluation

We perform evaluation on two standard semantic segmentation benchmarks: Pascal VOC 2007 (VOC'07) and Pascal VOC 2012 (VOC'12) [8]. When training the classifier we also include the image-wide labels from the objects designated as "difficult" and "truncated", since Cinbis *et al.* [4] showed that this improves the results of weakly-supervised localization. Both datasets contain 20 same classes, VOC'07 contains 9963 images, divided in approximately equal train and test splits. VOC'12 is around two times the size of VOC'07. We report intersection-over-union (IoU) averaged over all classes as the standard performance measure for semantic segmentation [8].

4.1 Experimental Setup

We used convolutional features from the conv5.4 layer of the 19-layer convolutional network VGG-E [22] pre-trained on ImageNet [7]. The number of feature maps in the selected convolutional layer is $d = 512$ which corresponds to the dimensionality of the feature vector. Each feature in the output map has a receptive field of 252×252 pixels, but we consider that a feature vector is mostly influenced by a small central patch. The patch size is determined from the number of max-pooling layers and the size of the pooling region: there are 4 2×2 max pooling layers up to 5^{th} convolutional layer net, so a pixel in the selected feature map corresponds to the patch of size 16×16 in the input image. The maximal size of images in both datasets is 500×500 pixels so the size of the largest feature map is 32×32. Unlike [3] we use just one scale. Preliminary results showed that classification performance with single scale features drops only slightly, while significantly reducing the run-time per image. We use convolutional part of the net just as local pre-trained feature extractor and do not perform any fine-tuning.

We use the Fisher vector embedding instead of fully-connected layers and use the same setting as in [3]: GMM with $K = 64$ components with diagonal covariance matrices. We also noticed that PCA on convolutional features yields lower classification performance, so we do not perform any pre-processing of convolutional features. This yields $D = K(1 + 2d)$ dimensional Fisher vectors for each image patch. Following standard Fisher vector classification pipeline [16] these are pooled by mean pooling into an image representation. We use additive normalization (zero-mean) and multiplicative normalization with the inverse of the Fisher matrix, which we assume diagonal. However we do not use non-linear normalizations of improved Fisher vector [16], to be able to keep additivity of the patch classification scores into the image classification score. Differently from [3], we use logistic regression instead of SVM, as our method requires class posterior estimates, as described in Sects. 3.2 and 3.3.

4.2 Results

Our first set of experiments quantitatively explores the influence of parameter m on results. Figure 2 shows the influence of m on the performance for particular classes. In Fig. 3 we display segmentations for different values of m. Parameter m influences the spatial extent of the segmented object: for small objects a higher value of m yields better performance, while for larger objects a lower m is better. High value of m yields high precision, indicating that classifier assigns high scores to class-specific patches. However, a classifier assigns low scores to large non-specific object parts, which for high m results in poor recall. Our region scoring allows propagation of scores from small class-specific parts to the greater spatial extent, increasing thus recall.

In Table 1 (left) we demonstrate the influence of region scoring on the results. When pixels' class posteriors are determined from a limited spatial extent corresponding to the the patch that covers it, the performance is poor. The performance is improved by including image level prior (ILP), as in [17]: the classes not detected in the image are downweighted, so the number of false positives is reduced.

From Fig. 2 it is clear that the best m depends on the class. For each class we determine the value $m(c)$ that maximizes segmentation IoU by treating a the bounding boxes from VOC'07 dataset as ground-truth segmentations. However, setting the determined $m(c)$ for each class in each image would yield different ranges of estimated class posteriors $p_c(c = 1|\mathbf{x})$. This would adversely

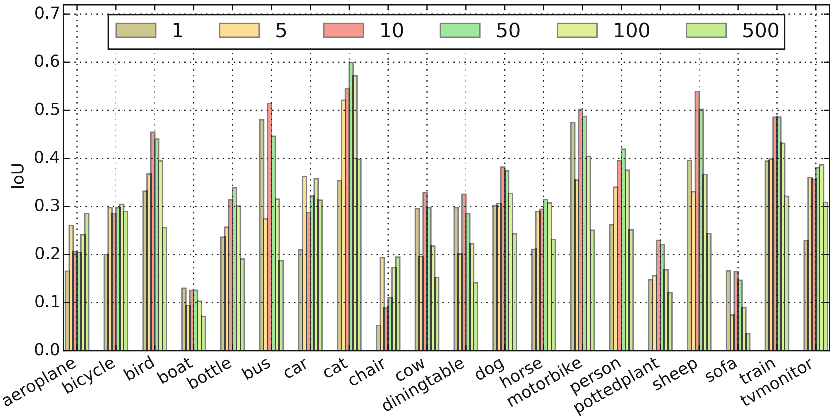

Fig. 2. Semantic segmentation performance on VOC'07 when varying the value of parameter m. Optimal value of m depends on the object class: classes with smaller objects *e.g.* airplanes and tv monitor benefit from higher m, while bigger objects benefit from low value of m *e.g.* motorbikes and buses.

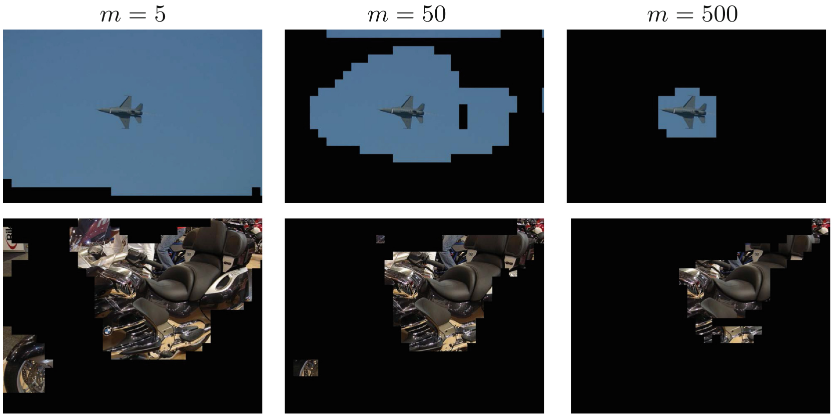

Fig. 3. The demonstration of different values of parameter m to the semantic segmentation. High m suppresses the influence of the regions with low prediction of the class. Therefore the parts of the sky, although highly correlated with airplane are removed when more weight is put on the contribution of smaller regions and individual patches. On the other hand, objects that cover large parts of the image benefit from the interplay of many larger object regions to propagate the decision from class-specific object parts to full spatial extent of the object, benefiting from lower values of parameter m.

influence segmentation performance because it introduces a bias towards classes with smaller $m(c)$. To this end we propose that m is determined per-image in inference as expectation over the classes, where the distribution over the classes is estimated by full image classifier: $\hat{m}(\mathbf{X}) = \mathbb{E}[m|\mathbf{X}] = \sum_c p_C(C = c|\mathbf{X})m(c)$.

For each class we randomly select the same number of bounding boxes, and explore the influence of the number and the random box sampling on segmentation performance by reporting mean and standard deviation. Using more boxes results in significantly better performance, so in the remainder of the paper we use per-class values of m determined from 20 training bounding boxes. We think that annotating 20 bounding boxes per class presents a modest effort compared to the benefit of improved segmentation performance.

In Table 1 (right) we compare our method to the state-of-the-art on challenging VOC'12 dataset. We use per-class m values that give best segmentation performance for 20 bounding boxes per class on VOC'07. We already achieve very competitive result without any post-processing. Pinheiro and Collobert [17] uses 760000 images from ImageNet (76x more than training set of VOC'12) to learn the segmentation model. Their model is trained in multi-class setting, which assumes one object per image, our training works in a more general multi-label setting. The per-class thresholds are determined from the ground-truth annotations of the VOC'12 train split. Finally, to improve precision, they use bottom up segmentation for post-processing. Their best results are achieved with costly multiscale combinatorial grouping (MCG) [1] segmentation. When superpixel segmentation is used, the proposed model outperforms their method, even

Table 1. *Left*: Results on VOC'07 show the proposed region voting significantly improves the segmentation performance. As little as 5 bounding boxes per class are sufficient for good performance. *Right*: Results on VOC'12: the proposed method gives comparable performance to the state-of-the-art methods that use more data, are more computationally complex and apply post-processing steps. Per-class performance is available at Pascal submission server.

Patch scores	IoU × 100
Patches only	10.63
Patches + ILP	18.66

Region scores (bboxes/class)	IoU × 100
5	32.24 ± 0.80
10	33.17 ± 0.65
20	33.59 ± 0.32

Method	IoU × 100
[17] with superpixel	36.6
[17] with MCG	42.0
[13] with CRF	39.6
Region score pooling (our)	38.0

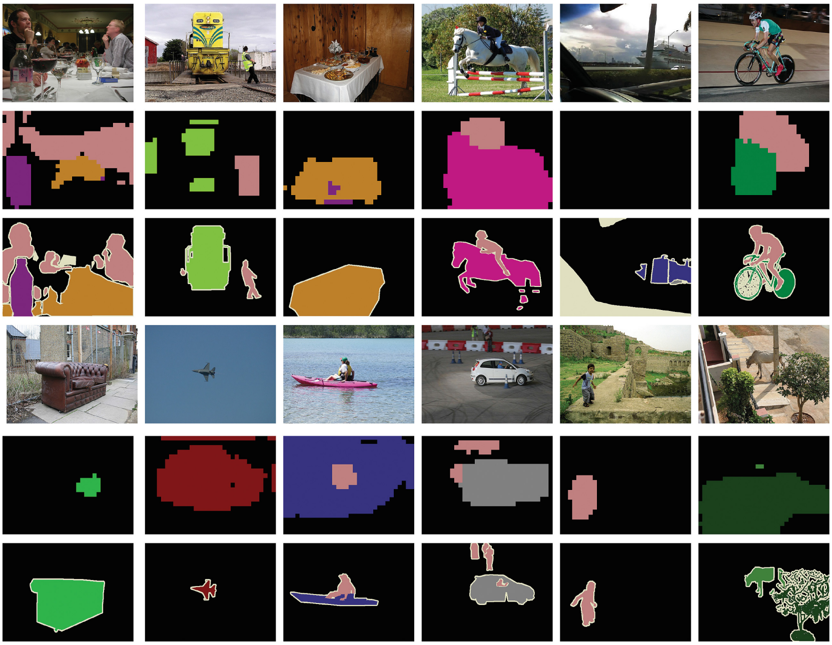

Fig. 4. Examples of segmentation produced by the proposed method.

though we do not perform any post-processing. We also achieve results comparable to the ones of Papandreou *et al.* [13] trained on the same data. Their method employs costly iterative training to fine-tune network weights, and a conditional random field (CRF) for post-processing.

In Fig. 4 we show some successful segmentations and some failure cases. The method is able to segment multiple objects in the image, either when they belong to the same of different classes. In some cases our method detects the objects that are not in ground-truth segmentations, *e.g.* part of the bottle that is seen through the glass. The main failure cases are due to the weak response of classification model which causes misdetections and due to the context of the object that occurs frequently with the object, *e.g.* water is not frequently seen in images that do not contain boats, and characteristic horseback-riding obstacles are not seen in images that do not contain horses. The problem of over and under segmentation is mostly due to the use of inappropriate value of parameter m indicating that better results can be obtained with better selection of hyper-parameters.

5 Conclusion

We have presented a simple and effective method for semantic segmentation of objects a in weakly supervised setting. Our method learns from image-wide labels and delivers pixel-level annotation of test images. Similarly to most other recent computer vision approaches, we build on the success of convolutional features learned on large image collections such as ImageNet. The novelty in our method addresses the heart of the weakly supervised segmentation problem: relating the pixel-level class posterior with a model trained on image-wide labels. Many previous works address that problem by aggregating independent pixel-level evidence. However, such formulation results in many false positives and relies on an image-level prior to alleviate this problem. We propose to relate the pixel-level posterior with the posteriors of all encompassing regions as determined by the image-wide classification model: a pixel is considered foreground if most of encompassing regions classify as foreground. This requirement effectively reduces the problem of both false positives and false negatives and significantly improves the performance. The proposed approach requires fast calculation of region-level classification scores, which we solve efficiently by using linear classifiers on top of Fisher embedding and integral images. The main advantages of our approach are conceptual simplicity and the capability to deliver competitive results without any kind of post-processing. The resulting method has only one per-class hyper-parameter which can be validated on few bounding box annotations. Experiments show competitive semantic segmentation performance on the standard test datasets of PASCAL VOC 2007 and 2012. Interesting directions for the future work include integration of information from multiple scales at region and pixel level via binary and higher-order CRFs.

Acknowledgement. This work has been fully supported by Croatian Science Foundation under the project I-2433-2014.

References

1. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014)
2. Boix, X., Gonfaus, J.M., van de Weijer, J., Bagdanov, A.D., Gual, J.S., Gonzàlez, J.: Harmony potentials - fusing global and local scale for semantic image segmentation. IJCV (2012)
3. Cimpoi, M., Maji, S., Kokkinos, I., Vedaldi, A.: Deep filter banks for texture recognition, description, and segmentation. IJCV 1–30 (2016)
4. Cinbis, R.G., Verbeek, J., Schmid, C.: Weakly supervised object localization with multi-fold multiple instance learning. In: PAMI (2016)
5. Csurka, G., Perronnin, F.: An efficient approach to semantic segmentation. IJCV (2011)
6. Dai, J., He, K., Sun, J.: BoxSup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: ICCV (2015)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
8. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. IJCV (2010)
9. Hong, S., Noh, H., Han, B.: Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation. CoRR abs/1506.04924 (2015)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS (2012)
11. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: a branch and bound framework for object localization. In: PAMI (2009)
12. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
13. Papandreou, G., Chen, L., Murphy, K.P., Yuille, A.L.: Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: ICCV (2015)
14. Pathak, D., Krähenbühl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: ICCV (2015)
15. Pathak, D., Shelhamer, E., Long, J., Darrell, T.: Fully convolutional multi-class multiple instance learning. In: ICLR Workshop (2015)
16. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
17. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: CVPR (2015)
18. Ramachandran, S., Mooney, R.J.: Revising Bayesian network parameters using backpropagation. In: ICNN (1996)
19. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
20. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.J.: Image classification with the fisher vector: theory and practice. IJCV (2013)
21. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. IJCV (2009)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)

23. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV (2013)
24. Xu, J., Schwing, A.G., Urtasun, R.: Learning to segment under various forms of weak supervision. In: CVPR (2015)
25. Zadrija, V., Krapac, J., Verbeek, J.J., Segvic, S.: Patch-level spatial layout for classification and weakly supervised localization. In: GCPR (2015)
26. Zhang, W., Zeng, S., Wang, D., Xue, X.: Weakly supervised semantic segmentation for social images. In: CVPR (2015)