From Image-level to Pixel-level Labeling with Convolutional Networks

Pedro O. Pinheiro[1,2], Ronan Collobert[1,3,†]
[1]Idiap Research Institute, Martigny, Switzerland.
[2]Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.
[3]Facebook AI Research, Menlo Park, CA, USA.

We are interested in inferring object segmentation by leveraging only object class information, and by considering only minimal priors on the object segmentation task. This problem could be viewed as a kind of weakly supervised segmentation task, and naturally fits the Multiple Instance Learning (MIL)) [2] framework: every training image is known to have (or not) at least one pixel corresponding to the image class label, and the segmentation task can be rewritten as inferring the pixels belonging to the class of the object (given one image, and its object class). Figure 1 shows a general illustration of our approach.

We rely on Convolutional Neural Networks (CNNs) [1], an important class of algorithms which have been shown to be state-of-the-art on large object recognition tasks. One advantage of CNNs is that they learn sufficiently general features, and therefore they can excel in transfer learning: *e.g.* CNN models trained on the Imagenet classification database could be exploited for different vision tasks. Their main disadvantage, however, is the need of a large number of fully-labeled dataset for training.

Our CNN-based model is not trained with segmentation labels, nor bounding box annotations. Instead, we only consider a single object class label for a given image, and the model is constrained to put more weight on important pixels for classification. In this context, every image is known to have (or not) – through the image class label – one or several pixels matching the class label. However, the positions of these pixels are unknown, and have to be inferred.

Our CNN is quite standard, with 10 levels of convolutions and (optional) pooling. The first 6 layers correspond to the feature learning layers from `Overfeat` [3], trained to perform object classification on the ILSVRC13 challenge. Each of the last 4 convolutional layers (but the last one $\mathbf{Y}$) is followed by a pointwise rectification non-linearity (ReLU):

$$\begin{aligned} \mathbf{H}^p &= \max(0, \mathbf{W}^p \mathbf{H}^{p-1} + \mathbf{b}^p)\,, \; p \in \{7,8,9\}\,, \\ \mathbf{Y} &= \mathbf{W}^{10}\mathbf{H}^9 + \mathbf{b}^{10}\,. \end{aligned} \tag{1}$$

Parameters of the p^{th} layer are denoted with $(\mathbf{W}^p, \mathbf{b}^p)$.

The network produces one score $s^k_{i,j} = Y^k_{i,j}$ for each pixel location (i,j) from the subsampled image I, and for each class $k \in \mathcal{C}$. Given that at training time we have only access to image classification labels, we need a way to aggregate these pixel-level scores into a single image-level classification score $s^k = aggreg_{i,j}(s^k_{i,j})$, that will then be maximized for the right class label $k^\star$. We chose as the aggregation function a smooth version and convex approximation of the *max* function, called *Log-Sum-Exp*:

$$s^k = \frac{1}{r} \log \left[\frac{1}{h^o w^o} \sum_{i,j} \exp(r\, s^k_{i,j}) \right]. \tag{2}$$

The hyper-parameter r controls how smooth one wants the approximation to be. The advantage of this aggregation is that pixels having similar scores will have a similar weight in the training procedure, r controlling this notion of "similarity".

We interpret image-level class scores as class conditional probabilities by applying a *softmax*:

$$p(k|I,\theta) = \frac{e^{s^k}}{\sum_{c \in \mathcal{C}} e^{s^c}}\,, \tag{3}$$

where $\theta = \{\mathbf{W}^p, \mathbf{b}^p \; \forall p\}$ represents all the trainable parameters of our architecture. We then maximize the log-likelihood (with respect to θ), over all the training dataset pairs $(I, k^\star)$:

$$\mathcal{L}(\theta) = \sum_{(k^\star, I)} \left[s^{k^\star} - \log \sum_{c \in \mathcal{C}} e^{s^c} \right]. \tag{4}$$

Training is achieved with stochastic gradient, backpropagating through the softmax, the aggregation procedure, and up the to first non-frozen layers of our network.

At test time, we feed the padded and normalized RGB test image I to our network, where the aggregation layer has been removed. Given we do not fine-tune our model on segmentation data, we observed our approach is subject to false positive. To circumvent this issue, we consider simple post-processing techniques, namely image-level prior (ILP) and three different smoothing priors (SP), with increasing amount of information.

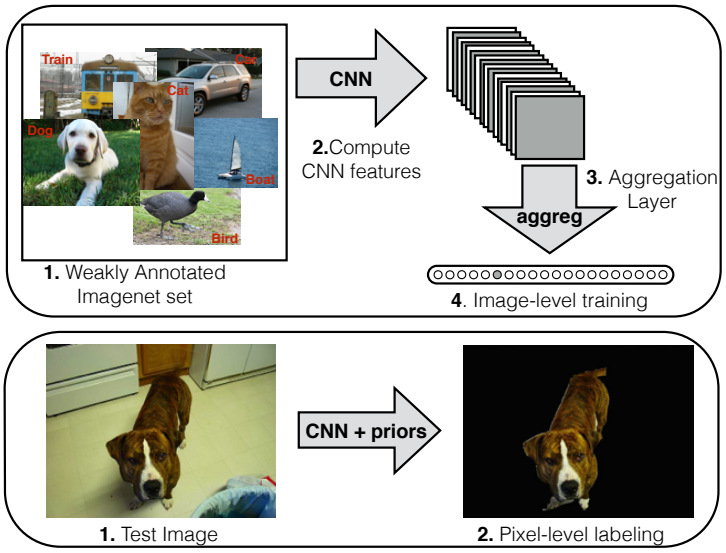

Figure 1: **A schematic illustration of our method**. *Top*: (1) The model is trained using weakly annotated data (only image-level class information) from Imagenet. (2) The CNN generates feature planes. (3) These planes pass through an aggregation layer. (4) The system is trained by classifying the correct image-level label. *Bottom*: During test, the aggregation layer is removed and the CNN densely classifies every pixel of the image (considering only few segmentation priors).

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[2] O. Maron and T. Lozano-Pérez. A framework for multiple instance learning. In *Advances in Neural Information Processing Systems (NIPS)*, 1998.

[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[†]All research was conducted at the Idiap Research Institute, before Ronan Collobert joined Facebook AI Research.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.