Feedforward semantic segmentation with zoom-out features

Mohammadreza Mostajabi, Payman Yadollahpour and Gregory Shakhnarovich
Toyota Technological Institute at Chicago
{mostajabi,pyadolla,greg}@ttic.edu

Abstract

*We introduce a purely feed-forward architecture for semantic segmentation. We map small image elements (superpixels) to rich feature representations extracted from a sequence of nested regions of increasing extent. These regions are obtained by "**zooming out**" from the superpixel all the way to scene-level resolution. This approach exploits statistical structure in the image and in the label space without setting up explicit structured prediction mechanisms, and thus avoids complex and expensive inference. Instead superpixels are classified by a feedforward multilayer network. Our architecture achieves **69.6%** average accuracy on the PASCAL VOC 2012 test set.*

1. Introduction

We consider one of the central vision tasks, *semantic segmentation*: assigning to each pixel in an image a category-level label. Despite attention it has received, it remains challenging, largely due to complex interactions between neighboring as well as distant image elements, the importance of global context, and the interplay between semantic labeling and instance-level detection. A widely accepted conventional wisdom, followed in much of modern segmentation literature, is that segmentation should be treated as a structured prediction task, which most often means using a random field or structured support vector machine model of considerable complexity.

This in turn brings up severe challenges, among them the intractable nature of inference and learning in many "interesting" models. To alleviate this, many recently proposed methods rely on a pre-processing stage, or a few stages, to produce a manageable number of hypothesized regions, or even complete segmentations, for an image. These are then scored, ranked or combined in a variety of ways.

Here we consider a departure from these conventions, and approach semantic segmentation as a single-stage classification task, in which each image element (superpixel) is labeled by a feedforward model, based on evidence computed from the image. Surprisingly, in experiments on PASCAL VOC 2012 segmentation benchmark we show that this simple sounding approach leads to results significantly surpassing previously published ones, advancing the current state of the art to 69.6%.

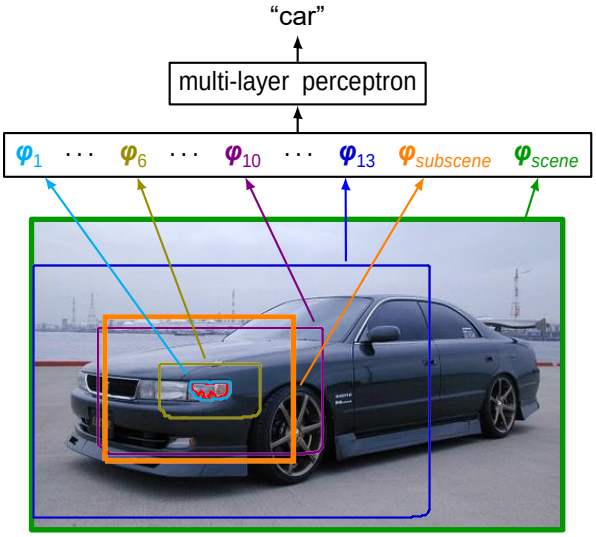

Figure 1. Schematic description of our approach. Features are extracted from a nested sequence of "zoom-out" regions around the superpixel at hand (red). Here we show four out of thirteen levels: 1 (cyan), 6 (olive), 10 (purple) and 13 (blue), as well as the subscene level (orange) and scene level (green). The features computed at all levels are concatenated and fed to a multi-layer perceptron that classifies the superpixel.

The "secret" behind our method is that the evidence used in the feedforward classification is not computed from a small local region in isolation, but collected from a sequence of levels, obtained by "zooming out" from the close-up view of the superpixel. Starting from the superpixel itself, to a small region surrounding it, to a larger region around it and all the way to the entire image, we compute a rich feature representation at each level and combine all the features before feeding them to a classifier. This allows us to exploit statistical structure in the label space and dependencies between image elements at different resolutions without explicitly encoding these in a complex model.

We do not mean to dismiss structured prediction or inference, and as we discuss in Section 5, these tools may be complementary to our architecture. In this paper we explore how far we can go without resorting to explicitly structured models.

We use convolutional neural networks (convnets) to extract features from larger zoom-out regions. Convnets, (re)introduced to vision in 2012, have facilitated a dramatic advance in classification, detection, fine-grained recognition and other vision tasks. Segmentation has remained conspicuously left out from this wave of progress; while image classification and detection accuracies on VOC have improved by nearly 50% (relative), segmentation numbers have improved only modestly. A big reason for this is that neural networks are inherently geared for "non-structured" classification and regression, and it is still not clear how they can be harnessed in a structured prediction framework. In this work we propose a way to leverage the power of representations learned by convnets, by framing segmentation as classification and making the structured aspect of it implicit. Finally, we show that use of multi-layer neural network trained with asymmetric loss to classify superpixels represented by zoom-out features, leads to significant improvement in segmentation accuracy over simpler models and conventional (symmetric) loss.

Below we give a high-level description of our method, then discuss related work and position our work in its context. Most of the technical details are deferred to Section 4 in which we describe implementation and report on results, before concluding in Section 5.

2. Zoom-out feature fusion

We cast category-level segmentation of an image as classifying a set of superpixels. Since we expect to apply the same classification machine to every superpixel, we would like the nature of the superpixels to be similar, in particular their size. In our experiments we use SLIC [1], but other methods that produce nearly-uniform grid of superpixels might work similarly well. Figures 2 provides a few illustrative examples for this discussion.

2.1. Scoping the zoom-out features

The main idea of our zoom-out architecture is to allow features extracted from different levels of spatial context around the superpixel to contribute to labeling decision at that superpixel. Before going into specifics of how we define the zoom-out levels, we discuss the role we expect different levels to play.

Local The narrowest scope is the superpixel itself. We expect the features extracted here to capture local evidence: color, texture, small intensity/gradient patterns, and other properties computed over a relatively small contiguous set of pixels. The local features may be quite different even for neighboring superpixels, especially if these straddle category or object boundaries.

Proximal As we zoom out and include larger spatial area around the superpixel, we can capture visual cues from surrounding superpixels. Features computed from these levels may capture information not available in the local scope; e.g., for locations at the boundaries of objects they will represent the appearance of both categories. For classes with non-uniform appearance they may better capture characteristic distributions for that class. We can expect somewhat more complex features to be useful at this level, but it is usually still too myopic for confident reasoning about presence of objects.

Two neighboring superpixels could still have quite different features at this level, however some degree of smoothness is likely to arise from the significant overlap between neighbors' proximal regions, e.g., A and B in Fig. 2. As another example, consider color features over the body of a leopard; superpixels for individual dark brown spots might appear quite different from their neighbors (yellow fur) but their proximal regions will have pretty similar distributions (mix of yellow and brown). Superpixels that are sufficiently far from each other could still, of course, have drastically different proximal features, e.g., A and C in Fig. 2.

Distant Zooming out further, we move to the distant levels : regions large enough to include sizeable fractions of objects, and sometimes entire objects. At this level our scope is wide enough to allow reasoning about shape, presence of more complex patterns in color and gradient, and the spatial layout of such patterns. Therefore we can expect more complex features that represent these properties to be useful here. Distant regions are more likely to straddle true boundaries in the image, and so this higher-level feature extraction may include a significant area in both the category of the superpixel at hand and nearby categories. For example, consider a person sitting on a chair; bottle on a dining table; pasture animals on the background of grass, etc. Naturally we expect this to provide useful information on both the appearance of a class and its context.

For nearby superpixels and far enough zoom-out level, distant regions will have a very large overlap, which will gradually diminish with distance between superpixels. This is likely to lead to somewhat gradual changes in features, and to impose a system of implicit smoothness "terms", which depend both on the distance in the image and on the similarity in appearance in and around superpixels. Imposing such smoothness in a CRF usually leads to a very complex, intractable model.

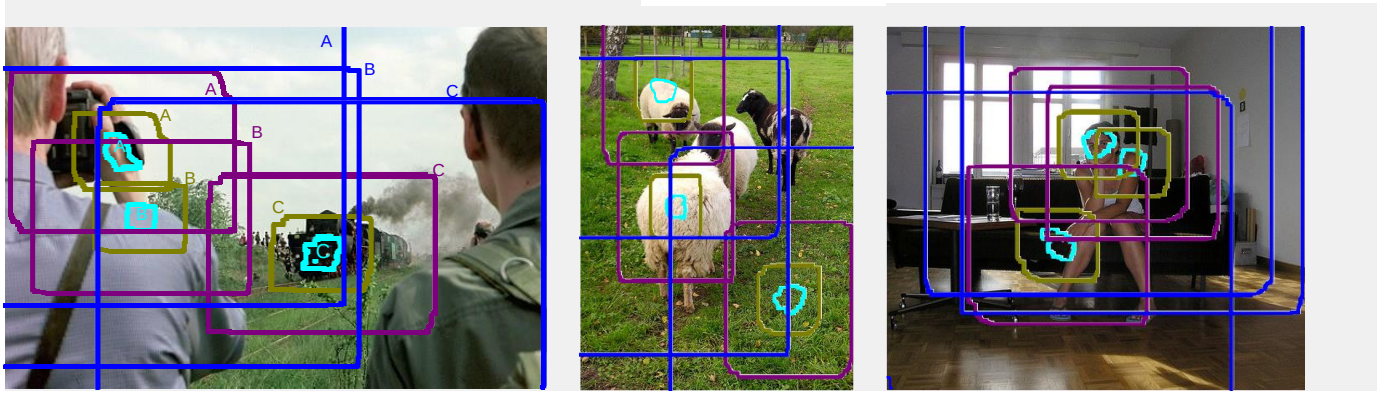

Figure 2. Examples of zoom-out regions. We show four out of fifteen levels: 1(cyan, nearly matching the superpixel boundaries), 6 (olive), 10 (purple) and 13 (blue). Levels 1-3 can be considered local, 3-7 proximal, and level 7-13 as well as the subscene level are distant.

Scene The final zoom-out scope is the entire scene. Features computed at this level capture "what kind of an image" we are looking at. One aspect of this global context is image-level classification: since state of the art in image classification seems to be dramatically higher than that of detection or segmentation [9, 31] we can expect image-level features to help determine presence of categories in the scene and thus guide the segmentation.

More subtly, features that are useful for classification can be directly useful for global support of local labeling decisions; e.g., lots of green in an image supports labeling a (non-green) superpixel as cow or sheep more than it supports labeling that superpixel as chair or bottle, other things being equal. Or, many straight lines in an image would perhaps suggest man-made environment, thus supporting categories relevant to indoors or urban scenes more than wildlife.

At this global level, all superpixels in an image will of course have the same features, imposing (implicit, soft) global constraints. This is yet another form of high-order interaction that is hard to capture in a CRF framework, despite numerous attempts [3].

2.1.1 Convnet-based zoom-out

In an early version of our work, we set the number and scope of zoom-out levels manually, and designed hand-crafted features to describe these levels. Details of this early approach can be found in an extended version of this paper [29]. However, we have found a different approach to work better. This approach relies on a convolutional neural network to provide both the feature values and the zoom-out levels used to accumulate these features.

A feature map computed by a convolutional layer with k filters assigns a k-dimensional feature vector to each receptive field of that layer. For most layers, this feature map is of lower resolution than the original image, due to subsampling induced by stride in pooling and possibly in filtering at previous layers. We upsample the feature map to the original image resolution, if necessary, using bilinear interpolation. This produces a k-dimensional feature vector for every *pixel* in the image. Pooling these vectors over a superpixel gives us a k-dimensional feature vector describing that superpixel. Figure 3 illustrated this feature computation for a superpixel with a toy network with three convolutional layers, interleaved with two pooling layers (2with 2×2 non-overlapping pooling receptive fields).

2.2. Learning to label with asymmetric loss

Once we have computed the zoom-out features we simply concatenate them into a feature vector representing a superpixel. For superpixel s in image $\mathbf{I}$, we will denote this feature vector as $\boldsymbol{\varphi}(s, \mathbf{I}) = [\boldsymbol{\varphi}_1(s, \mathbf{I}) \ldots, \boldsymbol{\varphi}_L(s, \mathbf{I})]$ where L is the number of levels in the zoom-out architecture. For the training data, we will associate a single category label y_s with each superpixel s. This decision carries some risk, since in any non-trivial over-segmentation some of the superpixels will not be perfectly aligned with ground truth boundaries. In section 4 we evaluate this risk empirically for our choice of superpixel settings and confirm that it is indeed minimal.

Now we are ready to train a classifier that maps s in image $\mathbf{I}$ to y_s based on $\boldsymbol{\varphi}(s, \mathbf{I})$; this requires choosing the empirical loss function to be minimized, subject to regularization. In semantic segmentation settings, a factor that must impact this choice is the highly imbalanced nature of the labels. Some categories are much more common than others, but our goal (encouraged by the way benchmark like VOC evaluate segmentations) is to predict them equally well. It is well known that training on imbalanced data without taking precautions can lead to poor results [10, 30, 22]. A common way to deal with this is to balance the training data; in practice this means that we throw away a large fraction of the data corresponding to the more common classes. We follow an alternative which we find less wasteful, and which in our experience often produces dramatically better results: use

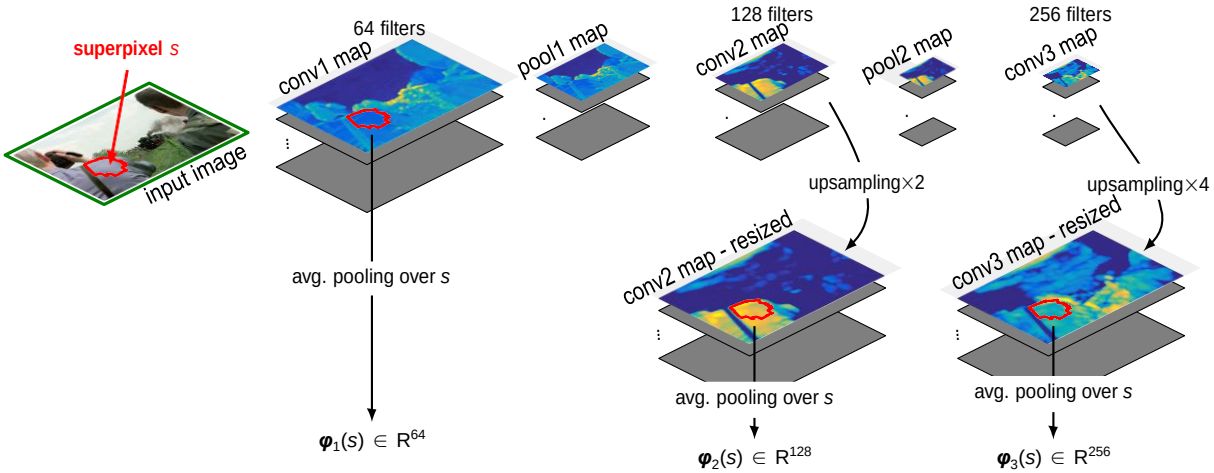

Figure 3. Illustration of zoom-out feature extraction procedure for a simple network with three convolutional and two pooling layers.

all the data, but change the loss. There has been some work on loss design for learning segmentation [37], but the simple weighted loss we describe below has to our knowledge been missed in segmentation literature, with the exception of [20] and [22], where it was used for binary segmentation.

Let the frequency of class c in the training data be f_c, with $\sum_c f_c = 1$. Our choice of loss is log-loss. We scale it by the inverse frequency of each class, effectively giving each pixel of less frequent classes more importance :

$$-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{f_{y_i}}\log\hat{p}\left(y_i|\boldsymbol{\phi}(s_i,\mathcal{I}_i)\right), \qquad (1)$$

where $p\left(y_i|\boldsymbol{\phi}(s_i,\mathbf{I}_i)\right)$ is the estimated probability of the correct label for segment s_i in image $\mathbf{I}_i$, according to our model. The loss in (1) is still convex, and only requires minor changes in implementation.[1]

3. Related work

The literature on segmentation is vast, and here we only mention work that is either significant as having achieved state of the art performance in recent times, or is closely related to ours in some way. In Section 4 we compare our performance to that of most of the methods mentioned here.

Many prominent segmentation methods rely on conditional random fields (CRF) over nodes corresponding to pixels or superpixels. Such models incorporate local evidence in unary potentials, while interactions between label assignments are captured by pairwise and possibly higher-order potentials. This includes various hierarchical CRFs [32, 21, 22, 3]. In contrast, we let the zoom-out features (in CRF terminology, the unary potentials) to capture higher-order structure.

[1]We found that further increasing the weight on background class improves the results; perhaps this is due to the fact that other classes are more often confiused with it than with each other.

Another recent trend has been to follow a multi-stage approach: First a set of proposal regions is generated, by a category-independent [6, 38] or category-aware [2] mechanism. Then the regions are scored or ranked based on their compatibility with the target classes. Work in this vein includes [4, 17, 2, 5, 24]. A similar approach is taken in [39], where multiple segmentations obtained from [4] are re-ranked using a discriminatively trained model. Recent advances along these lines include [15], which uses convnets and [8], which improves upon the re-ranking in [39], also using convnet-based features. In contrast to most of the work in this group, we do not rely on region generators, and limit preprocessing to over-segmentation of the image into a large number of superpixels.

The idea of using non-local evidence in segmentation, and specifically of computing features over a neighborhood of superpixels, was introduced in [11] and [25]; other early work on using forms of context for segmentation includes [32]. A study in [27] concluded that non-unary terms may be unnecessary when neighborhood and global information is captured by unary terms, but the results were significantly inferior to state of the art at the time.

Recent work closest to ours includes [10, 30, 34, 28, 16]. In [10], the same convnet is applied on different resolutions of the image and combined with a tree-structured graph over superpixels to impose smoothness. In [28] the features applied to multiple levels are also homogeneous, and handcrafted rather than learned. In [30] there is also a single convnet, but it is applied in a recurrent fashion, i.e., input to the network includes, in addition to the scaled image, the feature maps computed by the network at a previous level. A similar idea is pursued in [16], where it is applied to boundary detection in 3D biological data. In contrast with all of these, we use different feature extractors across levels, some of them with a much smaller receptive field than

any of the networks in the literature. We show in Section 4 that our approach obtains better performance (on Stanford Background Dataset) than that reported for [10, 30, 34, 28]; no comparison to [16] is available.

Finally, our work shares some ideas with other concurrent efforts. The main differences with [26, 13] are (i) that we incorporate a much wider range of zoom-out levels, (ii) we combine features, rather than predictions, across levels. Another difference is that these methods fine-tune the convnets on the segmentation task as part of an end-to-end learning, while we use a network pre-trained on the ImageNet classification task as-is. Despite this lack of fine-tuning, we achieve a significantly better performance on VOC 2012 test set than either of these methods (Table 3).

4. Experiments

Our main set of experiments focuses on the PASCAL VOC category-level segmentation benchmark with 21 categories, including the catch-all background category. VOC is widely considered to be the main semantic segmentation benchmark today[2]. The original data set labeled with segmentation ground truth consists of `train` and `val` portions (about 1,500 images in each). Ground truth labels for additional 9,118 images have been provided by authors of [14], and are commonly used in training segmentation models. In all experiments below, we used the combination of these additional images with the original `train` set for training, and `val` was used only as held out validation set, to tune parameters and to perform "ablation studies".

The main measure of success is accuracy on the test, which for VOC 2012 consists of 1,456 images. No ground truth is available for test, and accuracy on it can only be obtained by uploading predicted segmentations to the evaluation server. The standard evaluation measure for category-level segmentation in VOC benchmarks is per-pixel accuracy, defined as intersection of the predicted and true sets of pixels for a given class, divided by their union (IoU in short). This is averaged across the 21 classes to provide a single accuracy number, mean IoU, usually used to measure overall performance of a method.

4.1. Zoom-out feature computation

We obtained roughly 500 SLIC superpixels [1] per image (the exact number varies per image), with the parameter m that controls the tradeoff between spatial and color proximity set to 15, producing superpixels which tend to be of uniform size and regular shape, but adhere to local boundaries when color evidence compels it. This results in average superpixel region of 450 pixels.

[2]The Microsoft Common Objects in Context (COCO) promises to become another such benchmark, however at the time of writing it is not yet fully set up with test set and evaluation procedure

Given a convnet, we associate a zoom-out level with every convolutional layer in the network. In our main experiments, we use the 16-layer network from [33], with the final classification layer removed. This network, which we refer to as VGG-16, has 13 convolutional layers, yielding 13 zoom-out levels (see Fig. 3).

Group	Level	Dim	Unit RF size	Region size
G1	1	64	3	32
G1	2	64	5	36
G2	3	128	10	45
G2	4	128	14	52
G3	5	256	24	70
G3	6	256	32	84
G3	7	256	40	98
G4	8	512	60	133
G4	9	512	76	161
G4	10	512	92	190
G5	11	512	132	250
G5	12	512	164	314
G5	13	512	196	365
S1	subscene	4096	–	130
S2	scene	4096	–	varies

Table 1. Statistics of the zoom-out features induced by the 16 layer convnet. Dim: dimension of feature vector. Unit RF size is size of receptive field of a convnet unit in pixels of 256×256 input image. Region size: average size of receptive field of the zoom-out feature in pixels of original VOC images. The group designation is referred to by design of experiments in Section 4.3.

To compute subscene level features, we take the bounding box of superpixels within radius three from the superpixel at hand (i.e., neighbors, their neighbors and their neighbors' neighbors). This bounding box is warped to canonical resolution of 256×256 pixels, and fed to the convnet; the activations of the last fully connected layer are the subscene level features. Finally,the scene level features are computed by feeding the entire image, again resized to canonical resolution of 256×256 pixels, to the convnet, and extracting the activations of the last fully connected layer.

Feature parameters are summarized in Table 1. Concatenating all the features yields a 12,416-dimensional representation for a superpixel.

Following common practice, we also extract the features at all levels from the mirror image (left-right reflection, with the superpixels mirrored as well), and take element-wise max over the resulting two feature vectors.

4.2. Learning setup

To rule significant loss of accuracy due to reduction of image labeling to superpixel labeling, we evaluated the achievable accuracy under "oracle" labeling. Assigning each superpixel a category label based on the majority vote by pixels in it produces mean IoU of 94.4% on VOC 2012

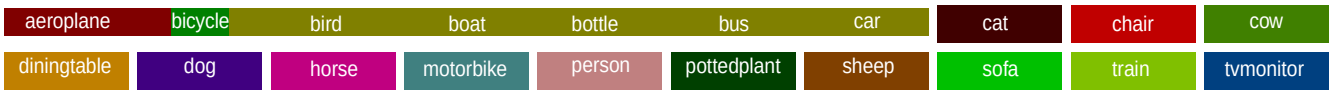

Figure 4. Color code for VOC categories. Background is black.

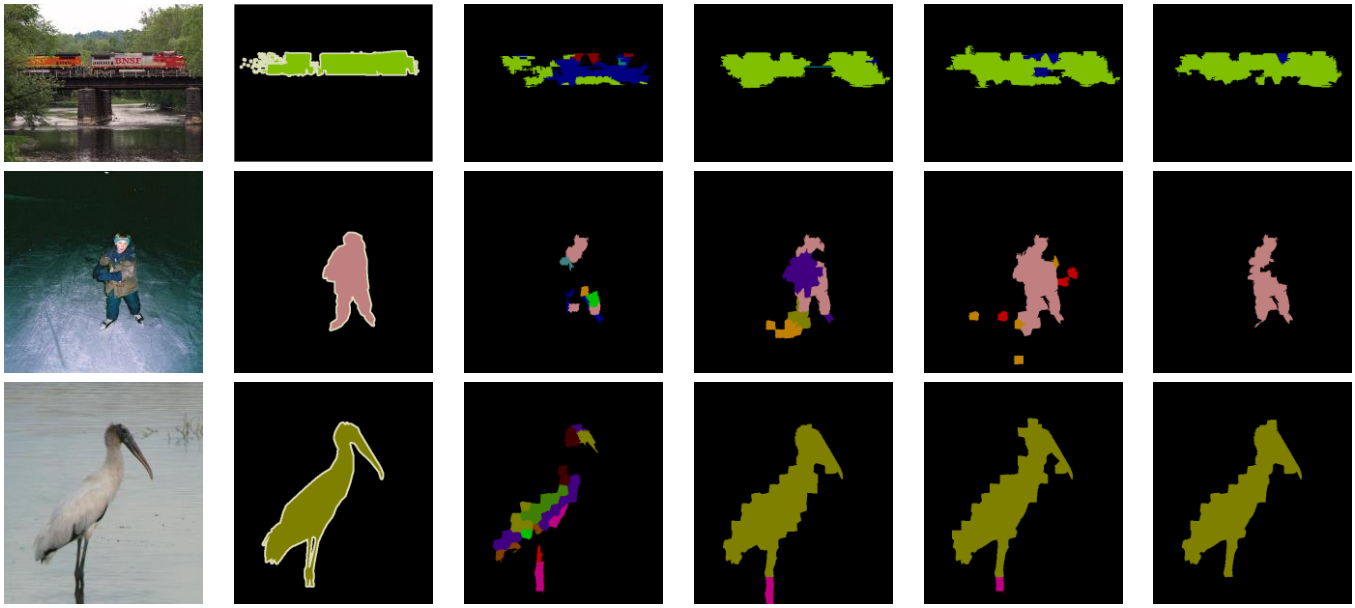

Figure 5. Examples illustrating the effect of zoom-out levels. From left: original image, ground truth, levels G1:3, G1:5, G1:5+S1, and the full set of zoom-out features G1:5+S1+S2. In all cases a linear model is used to label superpixels. See Figure 4 for category color code, and Table 1 for level notation.

val. Thus, we can assume that loss of accuracy due to our commitment to superpixel boundaries is going to play a minimal role in our results.

With more than 10,000 images and roughly 500 superpixels per image, we have more than 5 million training examples. We trained various classifiers on this data, with asymmetric log-loss (1), using Caffe [18] on a single machine equipped with a Tesla K40 GPU. During training we used fixed learning rate of 0.0001, and weight decay factor of 0.001.

4.3. Analysis of contribution of zoom-out levels

To assess the importance of features extracted at different zoom-out levels, we experimented with various feature subsets, as shown in Table 2. For each subset, we train a linear (softmax) classifier on VOC 2012 train and evaluate performance on VOC 2012 val. The feature set designations refer to the groups listed in Table 1.

It is evident that each of the zoom-out levels contributes to the eventual accuracy. Qualitatively, we observe that complex features computed at sub-scene and scene levels play a role in establishing the right set of labels for an image, while features derived from convolutional layers of the convnet are important in localization of object boundaries; a few examples in Figure 5 illustrate this. We also confirmed empirically that learning with asymmetric loss leads to better performance compared to standard, symmetric loss (and no data balancing).

Finally, we investigated the effect of replacing VGG-16 from [33] with a previously widely used 7-layer convnet referred to as AlexNet [19]. Our experience is consistent with previously reported results where the two networks were compared as feature extractors on a variety of tasks [12, 33, 26]: there is a significant drop in performance when we use the full zoom-out architecture induced by AlexNet compared to that induced by VGG-16 (mean IoU 45.4 vs. 58.6).

Feature set	mean accuracy
G1	6.0
G1-2	10.1
G1-3	16.3
G1-4	26.3
G1-5	41.8
G1-5+S1	51.21
G1-5+S2	57.3
G4-5+S1+S2	58.0
full zoom-out: G1-5+S1+S2	**58.6**

Table 2. Ablation study: importance of features from different levels under linear superpixel classification. Results on VOC 2012 val, mean class IoU.

Next we explored the impact of switching from linear softmax models to multilayer neural networks, evaluating a sequence of models on VOC 2012 val. Introducing a single layer of 1024 hidden units, with RELU nonlinearity, increased IoU from 58.6 to 68.4; additional hidden units (1500 or 2048) didn't increase it further. Adding another layer with 1024 units, and introducing dropout [35] improved IoU to 69.9, and this is the model we adopt for final evaluation on the test set. Results of this evaluation, in comparison to some related work, are summarized in Table 3.

Method	VOC2010	VOC2011	VOC2012
zoom-out (ours)	**69.9**	**69.4**	**69.6**
Hypercolumns [13]	–	–	62.6
FCN-8s [26]	–	62.7	62.2
DivMbest+convnet [8]	–	–	52.2
SDS [15]	–	52.6	51.6
DivMbest+rerank [39]	–	–	48.1
Codemaps [24]	–	–	48.3
O2P [4]	–	47.6	47.8
Regions & parts[2]	–	40.8	–
D-sampling [27]	33.5	–	–
Harmony potentials [3]	40.1	–	–

Table 3. Results on VOC 2010, 2011 and 2012 test. Mean IoU is shown, see Table 4 for per-class accuracies of the zoom-out method.

To evaluate importance of our reliance on superpixels we also evaluated an architecture in which SLIC superpixels are replaced by an equal number of rectangular regions. The achievable accuracy on VOC 2012 val with this oversegmentation is 87.2, compared to 94.4 with superpixels. The difference is due to failure of the rectangular grid to adhere to boundaries around thin structures or fine shape elements. Similar gap persists when we apply the full zoom-our architecture to the rectangular regions instead of superpixels: we get mean IoU of 64.3, more than 5 points below the result with superpixels. It would be interesting to run an experiment in which individual pixels are being classified (so the achievable accuracy would be 100%) but we have not done it so far, due to a prohibitive cost of such a run.

Figure 6 displays example segmentations. Many of the segmentations have moderate to high accuracy, capturing correct classes, in correct layout, and sometimes including level of detail that is usually missing from over-smoothed segmentations obtained by CRFs or generated by region proposals. On the other hand, despite the smoothness imposed by higher zoom-out levels, the segmentations we get do tend to be *under*-smoothed, and in particular include little "islands" of often irrelevant categories. To some extent this might be alleviated by post-processing; we found that we could learn a classifier for isolated regions that with reasonable accuracy decides when these must be "flipped" to the surrounding label, and this improves results on val by about 0.5 IoU, while making the segmentations more visually pleasing. We did not pursue this ad-hoc approach.

4.4. Results on Stanford Background Dataset

For some of the closely related recent work results on VOC are not available, so to allow for empirical comparison, we also ran an experiment on Stanford Background Dataset (SBD). It has 715 images of outdoor scenes, with dense labels for eight categories. We applied the same zoom-out architecture to this dataset as to VOC, except that the classifier had only 128 hidden units (due to much smaller size of the data set).

There is no standard train/test partition of SBD; the established protocol calls for reporting 5-fold cross validation results. There is also no single performance measure; two commonly reported measures are per-pixel accuracy and average class accuracy (the latter is different from the VOC measure in that it does not directly penalize false positives).

Method	pixel accuracy	class accuracy
zoom-out (ours)	**86.1**	**80.9**
Multiscale convnet [10]	81.4	76.0
Recurrent CNN [30]	80.2	69.9
Pylon [22]	81.9	72.4
Recursive NN [34]	78.1	–
Multilevel [28]	78.4	–

Table 5. Results on Stanford Background Dataset

The results in Table 5 show that the zoom-out architecture obtains results better than those in [30] and [10], both in class accuracy and in pixel accuracy.

5. Conclusions

The main point of this paper is to explore how far we can push feedforward semantic labeling of superpixels when we use multilevel, zoom-out feature construction and train non-linear classifiers (multi-layer neural networks) with asymmetric loss. The results are perhaps surprising: we can far surpass previous state of the art, despite apparent simplicity of our method and lack of explicit representation of the structured nature of the segmentation task. Another important conclusion that emerges from this is that we finally have shown that segmentation, just like image classification, detection and other recognition tasks, can benefit from the advent of deep convolutional networks.

We are working on implementing the zoom-out architecture as a single feed-forward network, to allow fine tuning of all the parameters jointly on segmentation data. We also plan to investigate the role inference could play in further improving the results of the zoom-out approach. We hope

class	mean	bg																				
acc	69.6	91.9	85.6	37.3	83.2	62.5	66	85.1	80.7	84.9	27.2	73.3	57.5	78.1	79.2	81.1	77.1	53.6	74	49.2	71.7	63.3

Table 4. Detailed results of our method on VOC 2012 test.

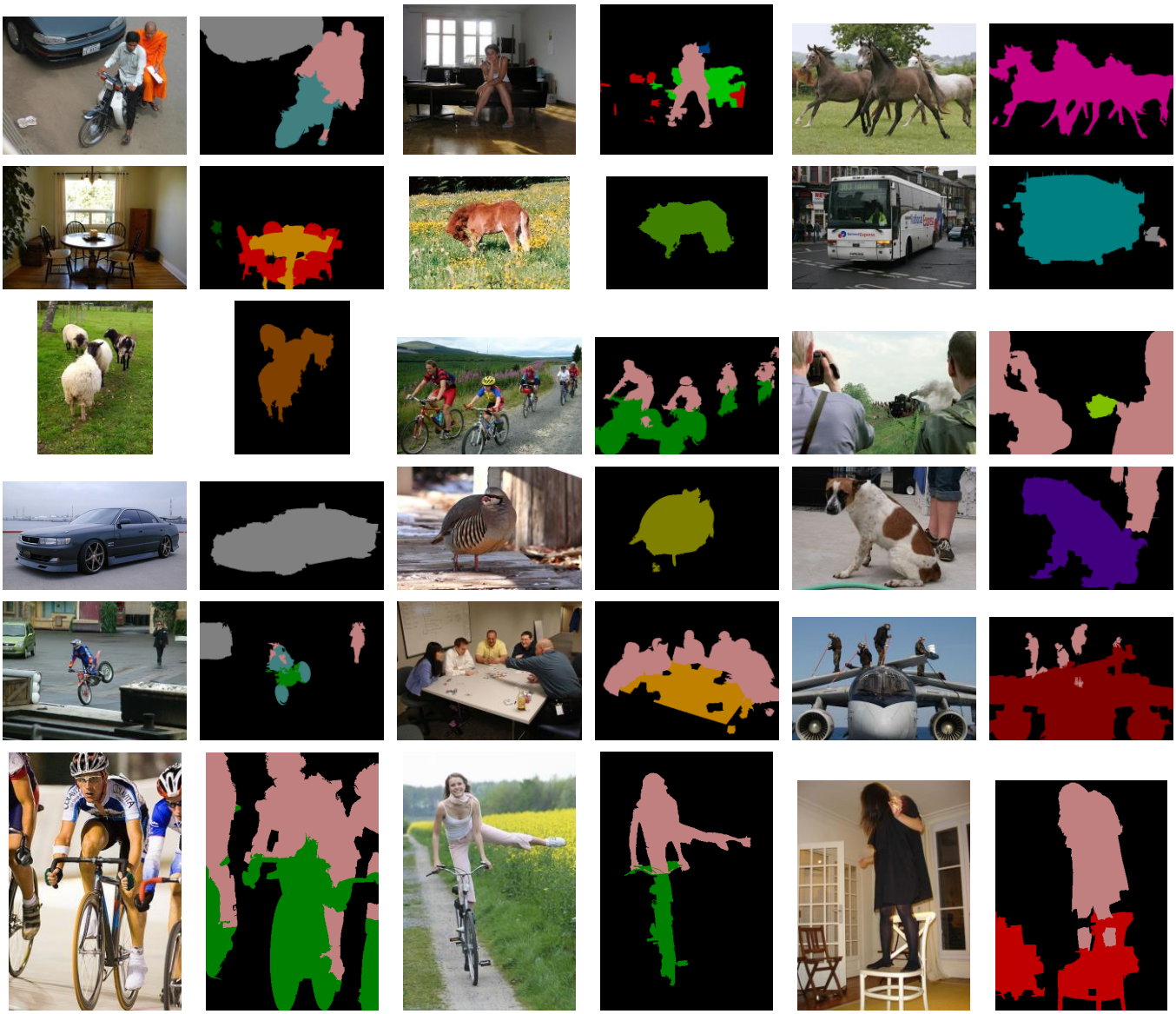

Figure 6. Example segmentations on VOC 2012 val with 3-layer neural network used to classify full zoom-out representation of superpixels (15 zoom-out levels). See Figure 4 for category color code.

that this can be done without giving up the feedforward nature of our approach; one possibility we are interested in exploring is to "unroll" approximate inference into additional layers in the feedforward network [23, 36]. This is in part motivated by recent success of work [7, 40] that uses a combination of convnets for classification with a CRF framework to explicitly impose higher-order constraints. These methods achieve results better than ours, although the gap is small, considering that they fine-tune the convnets to the task while we do not. Training the systems on the recently released COCO dataset further improves accuracy on VOC test. We plan to pursue all of these directions (end-to-end training, additional training data, and adding inference) to improve our system.

Acknowledgments

We gratefully acknowledge a gift of GPUs from NVIDIA corporation. We thank Gustav Larsson for help with Caffe, and for providing his code implementing weighted loss training. GS was partially supported by NSF award 1409837.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 2012.

[2] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012.

[3] X. Boix, J. M. Gonfaus, J. van de Weijer, A. D. Bagdanov, J. S. Gual, and J. Gonzàlez. Harmony potentials - fusing global and local scale for semantic image segmentation. *IJCV*, 96(1):83–102, 2012.

[4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *Computer Vision–ECCV 2012*, pages 430–443. Springer, 2012.

[5] J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *IJCV*, 98(3):243–262, 2012.

[6] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7), 2012.

[7] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint http://arxiv.org/abs/1412.7062*, 2015.

[8] M. Cogswell, X. Lin, and D. Batra. Personal communication. November 2014.

[9] M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 2014.

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE TPAMI*, 35(8), 2013.

[11] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *CVPR*, 2009.

[12] R. Girshick, J. Donohue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint http://arxiv.org/abs/1311.2524*, 2014.

[13] B. Hariharan, P. A. an R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *arXiv preprint http://arxiv.org/abs/1411.5752*, 2015.

[14] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

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

[16] G. B. Huang and V. Jain. Deep and wide multiscale recursive networks for robust image labeling. *arXiv preprint arXiv:1310.0354*, 2013.

[17] A. Ion, J. Carreira, and C. Sminchisescu. Probabilistic joint image segmentation and labeling. In *NIPS*, pages 1827–1835, 2011.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint http://arxiv.org/abs/1408.5093*, 2014.

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

[20] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *ECCV*, 2012.

[21] L. Ladickỳ, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. *ICCV*, 2009.

[22] V. Lempitsky, A. Vedaldi, and A. Zisserman. Pylon model for semantic segmentation. In *NIPS*, pages 1485–1493, 2011.

[23] Y. Li and R. Zemel. Mean field networks. In *ICML Workshop on Learning Tractable Probabilistic Models*, 2014.

[24] Z. Li, E. Gavves, K. E. A. van de Sande, C. G. M. Snoek, and A. W. M. Smeulders. Codemaps segment, classify and search objects locally. In *ICCV*, 2013.

[25] J. J. Lim, P. Arbeláez, C. Gu, and J. Malik. Context by region ancestry. In *ICCV*, 2009.

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint http://arxiv.org/abs/1411.4038*, 2015.

[27] A. Lucchi, Y. Li, X. Boix, K. Smith, and P. Fua. Are spatial and global constraints really necessary for segmentation? In *ICCV*, 2011.

[28] M. Mostajabi and I. Gholampour. A robust multilevel segment description for multi-class object recognition. *Machine Vision and Applications*, 2014.

[29] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. *arXiv preprint http://arxiv.org/abs/1412.0774*, 2015.

[30] P. H. O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv:1409.0575*, 2014.

[32] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1), 2009.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint http://arxiv.org/abs/1409.1556*, 2014.

[34] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *ICML*, 2011.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 2014.

[36] V. Stoyanov, A. Ropson, and J. Eisner. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *AISTATS*, 2011.

[37] D. Tarlow and R. S. Zemel. Structured output learning with high order loss functions. In *AISTATS*, 2012.

[38] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2), 2013.

[39] P. Yadollahpour, D. Batra, and G. Shakhnarovich. Discriminative re-ranking of diverse segmentations. In *CVPR*, 2013.

[40] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. *arXiv preprint* http://arxiv.org/abs/1502.03240, 2015.