Semantic Segmentation with Deep Learning

Michael Cogswell and Dhruv Batra
Virginia Tech, Blacksburg, VA
{cogswell, dbatra}@vt.edu

Abstract

We present a deep convolutional neural network approach for producing semantic segmentations. First, we generalize the architecture of the successful Alexnet network [7] to directly predict coarse segmentations. Second, we produce full resolution segmentations by re-ranking a diverse set of plausible segmentation proposals generated from a recent state of the art approach [9].

1. Introduction and Related Work

Training large Convolutional Neural Networks (CNNs) with lots of labeled training data has produced impressive results for classification and detection [7, 3, 5, 8]. Predicting a full semantic segmentation – an object level image labeling – is the next frontier.

A number of recent approaches have used CNNs to localize interesting image parts, but we will only mention some of the most recent and relevant examples here.

R-CNN [5] achieves excellent performance on both detection and segmentation using PASCAL VOC. Their approach produces object region proposals then classifies those regions with a CNN. OverFeat [8] performs detection using their deep net to determine likely bounding box locations as well as categorical labels for these bounding boxes.

Perhaps the closest to our approach, [4] trains a CNN to label images, however they ignore these predictions and use the intermediate layers as features in a more complex pipeline. The key differences of our approach are that we build on the successful Alexnet architecture and do not throw out the coarse labeling.

Finally, our approach is different from [5] in the sense that they re-score region proposals while we re-score the complete segmentation proposals from [9] via predicted coarse segmentations.

2. Approach

2.1. Predicting Coarse Segmentations

To predict a coarse segmentation instead of a bounding box or a class label we start with Caffenet [6], which is nearly identical to Alexnet [7] [1]. The convolutional layers are shift equivariant, but the fully connected layers are not, so even small translations in the input image could result in non-equivariant changes in the fully connected layer activations. Thus we remove the output layer and 2 fully connected layers, replacing them with two hidden convolutional layers, each producing 128 feature maps, and one convolutional layer which produces 21 feature maps. Before the final output we pass each pixel of these 13×13 feature maps through a 21-way softmax function instead of ReLU non-linearities, giving $13 \times 13 = 169$ distributions over the 21 classes (PASCAL classes plus background). The result is a CNN with 7 hidden convolutional layers taking images as input and producing coarse, soft image labelings. An example output is visualized in figure 2b.

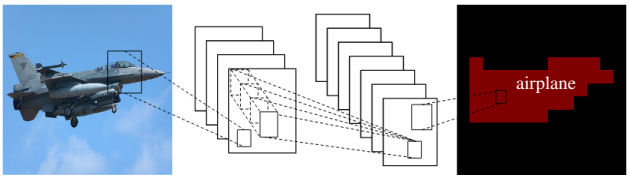
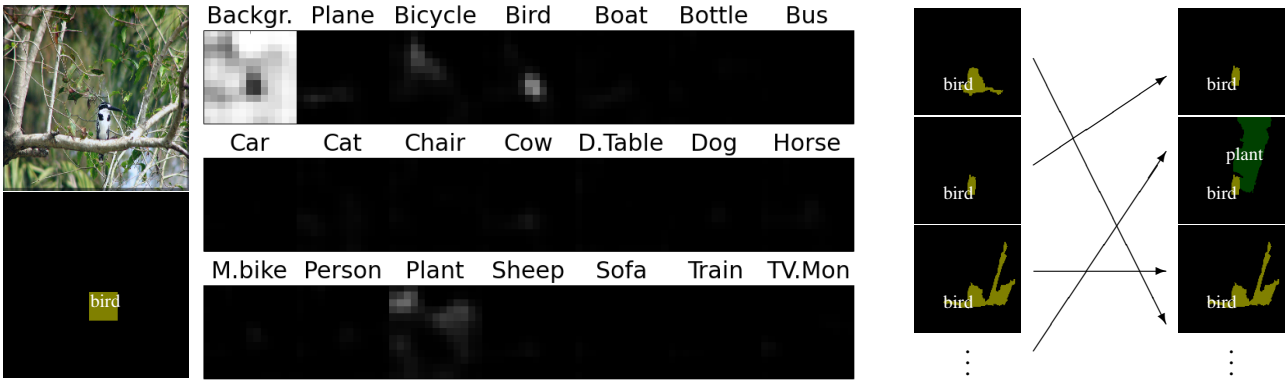


Figure 1: We predict a coarse image labeling directly from our Convolutional Neural Network.

To construct ground-truth 13×13 coarse segmentations, we downsample full-sized PASCAL segmentation annotations. Specifically, we divide the full-sized segmentations into patches of roughly equal size; PASCAL image sizes vary, but for a 500×350 image one such patch is about 38×27. Inside this patch we have $38 \times 27 = 1026$ pixel labels, so we can extract a distribution over the 21 classes at each patch. Note that even though localization is coarse, some detailed sub-patch reasoning is still present because each patch predicts how much of each class it contains instead of a hard class label.

During training we did not update the first 5 layers (initialized from the pre-trained Caffenet); even in the final stages of training the net tended to overfit when these layers were finetuned. Unlike previous architectues, we applied dropout before the 3 added convolutional layers.

[1] Differences are summarized in https://github.com/BVLC/caffe/issues/296



(a) (b) (c)

2a Image and Coarse Segmentation produced by our CNN 2b Class-wise breakdown of predictions by our CNN 2c Top 3 of 10 O_2P+DivMBest segmentations before (left) and after (right) re-ranking by similarity to coarse segmentations

Figure 2: Examples

	Backgr.	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	D.Table	Dog	Horse	M.bike	Person	Plant	Sheep	Sofa	Train	TV.Mon	Average
O_2P [1]	84.8	63.7	23.4	44.9	40.8	45.1	58.0	58.8	57.6	12.1	43.8	31.0	44.8	**56.2**	56.8	52.3	**37.1**	44.0	29.5	48.6	42.9	46.5
O_2P DivMBest+ReRank [9]	**85.7**	62.7	**25.6**	46.9	**43.0**	**54.8**	58.4	58.6	55.6	**14.6**	**47.5**	**31.2**	44.7	51.0	**60.9**	53.5	36.6	50.9	**30.1**	50.2	**46.8**	**48.1**
Ours	85.1	**69.5**	25.4	**58.0**	42.7	33.4	**69.3**	**59.1**	**60.2**	13.8	38.1	21.9	**52.9**	42.4	52.1	**55.5**	34.4	**61.9**	23.8	**60.3**	43.2	47.8

Table 1: PASCAL VOC 2012 segmentation results. [1] is the same as picking the highest scoring DivMBest solution.

2.2. Re-Ranking to Select a Full Sized Segmentation

In order to produce full-resolution segmentations, we build on the recent O_2P+DivMBest work of [1, 9]. They generate approximately 150 CPMC segments [2] for each image then score them using Support Vector Regressors over second-order pooled features [1], finally greedily pasting the segments. The sum of the scores of the pasted segments is the score of a segmentation and DivMBest is used to produce diverse segmentation maps. This is currently the highest performing system on the PASCAL VOC 2012 segmentation challenge. In the interest of simplicity, we do not utilize their re-ranking features, instead using our coarse segmentations to re-rank the DivMBest segmentations.

First, in the same manner as for ground truth, we downsample each of the DivMBest segmentations to 13×13 soft segmentations. Let $\hat{p}_{ik}$ denote the predicted probability of class k at patch i from our net, and $\hat{q}_{ik}^m$ be the probability of the m^{th} DivMBest segmentation downsampled. We measure the consistency score of $\hat{p}$ and $\hat{q}^m$ with the symmetric KL augmented with a background penalty term:

$$S(m) = \sum_i \left[D_{KL}(\hat{p}_i || \hat{q}_i^m) + D_{KL}(\hat{q}_i^m || \hat{p}_i) + 0.02\hat{q}_{i,0}^m \right]. \quad (1)$$

where D_{KL} is the Kullback-Leibler divergence and $0.02\hat{q}_{i,0}^m$ is a regularizer that penalizes background prediction. The reason for the background penalty is that we observed the background (class 0) tends to be overpredicted; adding this term consistently improved validation performance. Note that our final predictions are actually generated from the O_2P+DivMBest pipline. Coarse segmentations are only used to do the re-ranking step.

3. Results / Conclusion

Results on the `val` set from the PASCAL 2012 segmentation challenge using 5-fold cross validation are reported in table 1. Our approach performs well on some classes and poorly on others. On average, our approach is able to outperform the O_2P baseline, which corresponds to picking the highest scoring DivMBest solution. We perform worse than the re-ranking approach of [9], however, we note that they have access to a richer set of features.

In future work we hope to further explore this method by looking at different ways of training the net and utilizing the re-ranking features of [9] in addition to these coarse segmentations to get the best of both worlds.

Acknowledgements. This work was partially supported by the National Science Foundation under Grant No. IIS-1353694.

References

[1] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443, 2012. 2

[2] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. 2

[3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 1

[4] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013. 1

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 1

[6] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013. 1

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, volume 1, page 4, 2012. 1

[8] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. 1

[9] P. Yadollahpour, D. Batra, and G. Shakhnarovich. Discriminative re-ranking of diverse segmentations. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1923–1930. IEEE, 2013. 1, 2