LEARNING DENSE CONVOLUTIONAL EMBEDDINGS FOR SEMANTIC SEGMENTATION

Adam W. Harley & Konstantinos G. Derpanis
Department of Computer Science
Ryerson University
Toronto, ON, M5B 2K3, Canada
{aharley,kosta}@scs.ryerson.ca

Iasonas Kokkinos
Center for Visual Computing
CentraleSupélec and INRIA
Chatenay-Malabry, 92095, France
iasonas.kokkinos@ecp.fr

ABSTRACT

This paper proposes a new deep convolutional neural network (DCNN) architecture that learns pixel embeddings, such that pairwise distances between the embeddings can be used to infer whether or not the pixels lie in the same region. That is, for any two pixels on the same object, the embeddings are trained to be similar; for any pair that straddles an object boundary, the embeddings are trained to be dissimilar. Experimental results show that when the embeddings are used in conjunction with a DCNN trained on semantic segmentation, there is a systematic improvement in per-pixel classification accuracy. These contributions are integrated in the popular Caffe deep learning framework, and consist in straightforward modifications to convolution routines. As such, they can be exploited for any task involving convolution layers.

1 INTRODUCTION

Deep convolutional neural networks (DCNNs) (LeCun et al., 1998) are the method of choice for a variety of high-level vision tasks (Razavian et al., 2014). Fully-convolutional DCNNs have recently been a popular approach to semantic segmentation, because they can be efficiently trained end-to-end for pixel-level classification (Sermanet et al., 2014; Chen et al., 2014; Long et al., 2014).

A weakness of DCNNs is that they tend to produce smooth and low-resolution predictions, partly due to the subsampling that is a result of cascaded convolution and max-pooling layers. Many different strategies have been explored to overcome this issue. One popular strategy is to add a dense conditional random field (CRF) to the end of the DCNN, introducing contextual information to the segmentation via long-range dependencies in the CRF (Chen et al., 2014; Lin et al., 2015). Another strategy is to reduce the subsampling effected by convolution and pooling, by using the "hole" algorithm for convolution (Chen et al., 2014). A third strategy is to add trainable up-sampling stages to the network via "de-convolution" layers in the DCNN (Noh et al., 2015; Long et al., 2014).

This paper's strategy, which is complementary to those previously explored, is to train the network to produce segmentation-like information, so that foreground pixels and background pixels within local patches can be treated differently. For instance, as can be seen in Figure 1, if a DCNN is centered on a "boat" pixel, but the surrounding patch includes some pixels from the background, the DCNN's final prediction will typically reflect the presence of the distractors by outputting a mix of "boat" and "background". The approach of this paper is to learn and use semantic affinities between pixels, so that the DCNN output centered at a "boat" pixel can be strengthened by using information from other "boat" pixels within the patch. More generally, the approach allows the prediction at any pixel to be replaced with a weighted average of similar neighboring predictions. This has the effect of sharpening the predictions made at object boundaries, while smoothing in region interiors.

The key to accomplishing this is to have the network produce internal representations that lend themselves to pairwise comparisons, such that any pair that lies on the same object will produce a high affinity measure, and pairs that straddle a boundary produce a low affinity measure. Prior work has investigated the use of affinity cues in similar contexts (Ren & Malik, 2003; Dai et al., 2014), but required handcrafted algorithms for computing the affinity information. This would typically be

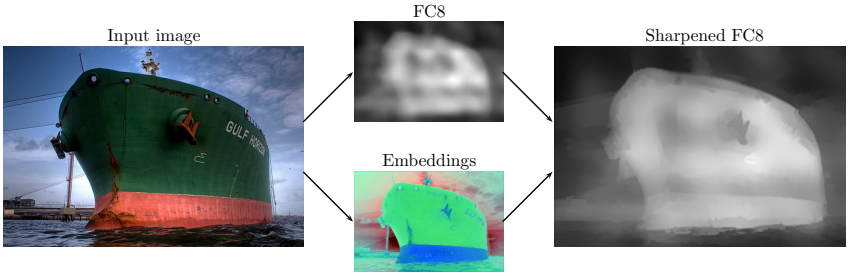

Figure 1: Given an input image (left), a DCNN produces a smooth prediction map (middle top), and an embedding network produces dense segmentation cues (middle bottom). Filtering the predictions via the learned embeddings produces a sharp prediction map (right).

pre-computed in a separate process. The current work is unique for learning the cues directly from image data, and for computing the affinities densely and "on the fly" within a DCNN.

The learned embeddings and their distance functions are implemented efficiently as convolution-like layers in Caffe (Jia et al., 2014). The embedding layers can either be trained independently, or integrated in the full DCNN pipeline and trained end-to-end (with or without per-pixel labels). Source code is publicly available at `https://github.com/aharley/embeddings`.

2 TECHNICAL APPROACH

The goal of this work is to train a set of convolutional layers to create dense "embeddings", which can be used to calculate pixel affinities relating to the semantic similarity of the underlying regions. Pixel pairs that share a semantic category should produce similar embeddings (i.e., a high affinity), and pairs with different categories should produce dissimilar embeddings (i.e., a low affinity).

This goal is represented in a loss function, $\mathcal{L}$, which accumulates the quality of embedding pairs sampled across the image. In this work, pairwise comparisons are made between each pixel i and its spatial neighbours, $j \in N(i)$. Collecting pairs within a fixed window lends simplicity and tractability to the approach, although in general the pairs can be collected at any range. Denoting the quality of a particular pair of embeddings with ℓ_{ij}, the overall loss is defined as $\mathcal{L} = \sum_{i \in I} \sum_{j \in N(i)} \ell_{ij}$. The network is trained to minimize this loss through stochastic gradient descent.

The inner loss, ℓ_{ij}, represents how well a pair of embeddings, e_i and e_j, respect the affinity goal. Pixel-wise labels are a convenient resource for quantifying this loss, since they can indicate whether or not pairs of pixels share a semantic label. Using this information, the distance between embeddings can be optimized according to label parity. That is, same-label pairs can be optimized to have a small distance, and different-label pairs can be optimized to have a large distance. Denoting the label of pixel i with l_i, and the embedding at that pixel with e_i, the inner loss is defined as

$$\ell_{ij} = \begin{cases} \max\left(|e_i - e_j| - \alpha, 0\right) & \text{if } l_i = l_j \\ \max\left(\beta - |e_i - e_j|, 0\right) & \text{if } l_i \neq l_j \end{cases}, \tag{1}$$

where α and β are design parameters that specify the "near" and "far" thresholds against which the embedding distances are compared. The embedding distances can be computed with any distance function. Figure 2 shows visualizations of L_1-based learned embeddings, with $\alpha = 0.5$, and $\beta = 2$.

Once the embeddings are learned, they can be used to create segmentation masks. For a pixel i and a neighbour pixel $j \in N(i)$, one can define

$$m_i = \exp(-\lambda |e_i - e_j|) \tag{2}$$

to be the weight applied to pixel j in a mask centered on i, where λ is a parameter specifying the hardness of the mask. This parameter can be learned inside a DCNN. When the masks are applied convolutionally, the effect is to replace each input with a weighted average of its similar neighbours. Note that if the mask were a Gaussian that jointly captured RGB and geometric distance between pixels i and j, it would be equivalent to the bilateral filter (Tomasi & Manduchi, 1998), which is a well-known technique in signal processing for smoothing while preserving edges. The current approach represents a generalization of the bilateral filter. The Krähenbühl & Koltun (2011) algorithm

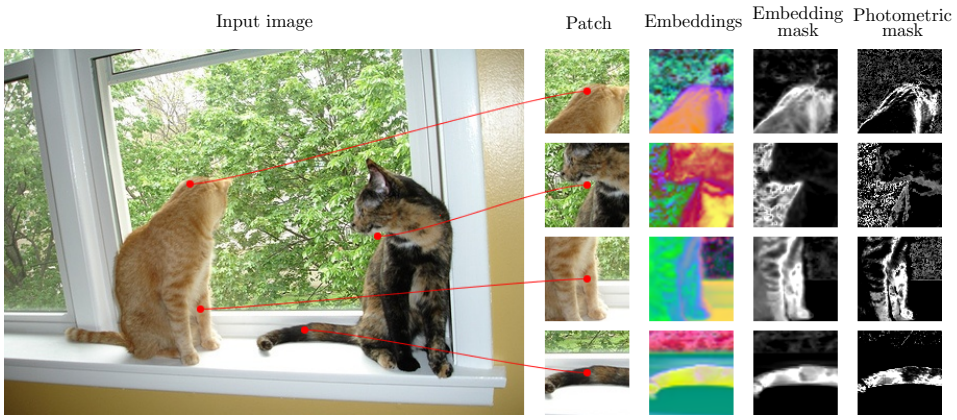

Figure 2: Embeddings and local masks are computed densely for input images. For four locations in the image shown on the left, the figure shows the extracted patch, embeddings (compressed to three dimensions by PCA, for visualization purposes), and embedding-based mask (left-to-right). For comparison, the mask generated by photometric color distances is shown on the far right.

Table 1: VOC 2012 test results (IOU %)

Method	**Without embeddings**	**With embeddings**
DeepLab	70.31	71.54
DeepLab-CRF	73.60	74.00

for dense CRFs is also related to the bilateral filter, in the sense that the inference step, through mean field approximation, accomplishes a repeated application of a non-linear filter. This shared connection is appropriate, since CRFs and embedding-based segmentation masks have common goals: to sharpen predictions at object boundaries while smoothing the interior.

3 Evaluation

The baseline for the evaluation is the current best publicly-released DeepLab network ("Deep-MSc-Coco-LargeFOV"; Chen et al., 2014), which is a strong baseline for semantic segmentation. The current work augments this network with embeddings learned on COCO (Lin et al., 2014) and PASCAL VOC 2012 (Everingham et al., 2012), which interact with DeepLab by filtering layer FC8.

Validation experiments explored the effects of two design parameters: filter window size, and the number of times to apply the filter. A wider window improves performance by allowing information from a wider radius to contribute to each prediction. Similarly, running the masking process repeatedly strengthens the contribution from similar neighbours, and also effectively increases the number of contributing neighbours. The best embedding configuration (9×9 filters applied seven times recursively) was fine-tuned together with DeepLab, and submitted to the VOC test server. As shown in Table 1, performance improved 1.2% over the baseline. Finally, a dense CRF (Krähenbühl & Koltun, 2011) was trained on top of the sharpened FC8 outputs, to test whether or not the embeddings would still contribute an improvement (despite the similarity of the CRF approach). As shown in Table 1, the improvement added in this context was 0.4%.

4 Conclusion

This paper proposed a new deep convolutional neural network architecture for learning embeddings. Results showed that integrating the embeddings into a strong DCNN baseline systematically improved results by a noticeable margin on the PASCAL VOC 2012 test set. Although semantic segmentation is the target application of the current work, the overall approach does not depend on pixel-wise labels, and the embedding and masking layers can be used in any task involving DCNNs.

REFERENCES

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., , and Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2014.

Dai, J., He, K., and Sun, J. Convolutional feature masking for joint object and stuff segmentation. *arXiv*, 2014.

Everingham, M., Van-Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. *arXiv*, 2014.

Krähenbühl, P. and Koltun, V. Efficient inference in fully connected CRFs with Gaussian edge potentials. *NIPS*, pp. 109–117, 2011.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

Lin, G., Shen, C., Reid, I. D., and van den Hengel, A. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv*, 2015.

Lin, T.-Yi, Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *ECCV*, pp. 740–755, 2014.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR*, 2014.

Noh, H., Hong, S., and Han, B. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

Razavian, A.S., Azizpour, H., Sullivan, J., and Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPR*, pp. 512–519, 2014.

Ren, X. and Malik, J. Learning a classification model for segmentation. In *ICCV*, pp. 10–17, 2003.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

Tomasi, C. and Manduchi, R. Bilateral filtering for gray and color images. In *ICCV*, pp. 839–846, 1998.