SEMANTIC SEGMENTATION BASED ON ITERATIVE CONTRACTION AND MERGING

Tzu-Hao Yang, Jia-Hao Syu, Sheng-Jyh Wang

National Chiao Tung University, Hsinchu, Taiwan

ABSTRACT

The state-of-the-art models for semantic image segmentation usually contain a convolutional neural network (CNN) and a conditional random field (CRF). As a predictor, existing CNN techniques can generate a dense prediction result but may generate obvious boundary errors at the same time. As a refinement model, CRF improves the CNN outcomes by forcing the consistency of local labels. However, the use of CRF may cause fragmentation effect around object boundaries. In this paper, we propose the use of a so-called iterative contraction and merging (ICM) process to facilitate the semantic segmentation process. Guided by the high-level information from CNN, the ICM process is used as a tool to grow image segments in a bottom-up way and to produce more accurate outcomes in an iterative way. The ICM process can faithfully preserve the boundary information and maintain the consistency of local labels. Our experimental results demonstrate that the performance of the proposed approach is comparable to the state-of-the-art models but with more accurate boundaries.

Index Terms— Image Semantic Segmentation, Convolutional Neural Networks.

1. INTRODUCTION

During the rapid development of related techniques over the last decade, semantic segmentation has become one of the crucial issues in computer vision. Given an image/video, semantic segmentation techniques aim to assign each pixel/voxel a semantic label and generate a dense semantic prediction. As compared to the detection issue, semantic segmentation provides more precise description of object shapes.

The recent revival of convolutional neural networks (CNNs) [1,2,3,4] has brought a revolution to several topics of computer vision. Semantic segmentation is one of them. Based on CNN, obvious improvement over earlier models has been made in semantic segmentation, as presented in [5,6,7,8]. Among these works, the fully convolutional network (FCN) model proposed by Long *et al.* [5] has demonstrated its great potential as a boosting model and has been widely adopted in various applications. After [5], plentiful dense CNN models [9,10,11] have been proposed to further enhance the performance of the vanilla FCN.

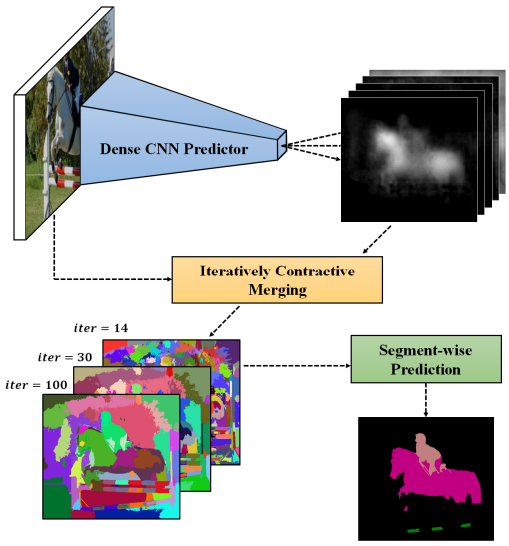

Fig. 1. *Illustration of the proposed method*

Since semantic segmentation mainly pursues an accurate global solution, it is inevitable that some errors may occur around the boundary area if we use convolution-based models. Up to now, several refinement models have been proposed to compensate the weak local label consistency in the outcomes of dense CNN models. Conditional Random Field (CRF) [12,13,14,15,16] is one of the most popular models. However, CRF has been criticized for being not intuitive and being relatively complex. Moreover, most CRF models are vulnerable to image noise which may cause the boundary fragmentation effect on the semantic prediction map. The above factors motivate us to construct a new refinement model, which aims to be more intuitive, less complicated, and less boundary fragmental.

On the other hand, hierarchical segmentation techniques [17,18,19,20] have been developed for a long time and have gradually become applicable to other issues in computer vision. Since most hierarchical segmentation techniques focus on low-level information, such as color and location features, they possess a great ability to maintain local label consistency and may preserve the boundary information well. These properties make unsupervised segmentation techniques suitable as an alternative for refinement models.

In this paper, we combine the dense CNN predictor with a so-called iterative contraction and merging process (ICM) [17], which is a relatively fast high-performance hierarchical segmentation algorithm. Guided by the high-level information provided by the dense CNN predictor, the ICM process aims to grow image segments in a bottom-up way by including both low-level features and high-level information. A new shape feature, called tortuosity, is also proposed to help the ICM process explore the affinity among image segments.

The rest of this paper is organized as follows. In Section 2, we present the proposed ICM process and some technical details. Section 3 includes the information of the segment-wise prediction process. In Section 4, we present the experimental results to demonstrate the feasibility of the proposed algorithm, together with some comparisons with existing models. Finally, we conclude the paper in Section 5.

2. ITERATIVE CONTRACTION AND MERGING

Originally, the ICM process is proposed in [17] for unsupervised image segmentation. In this paper, as shown in Fig. 2, we propose the modified two-phase structure ICM for semantic segmentation. In our method, the CIE Lab color space, which is designed to approximate human's color perception, is adopted for describing the color information.

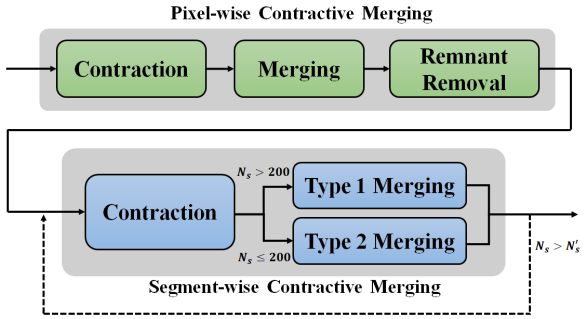

***Fig. 2**. Block diagram of the iteratively contraction and merging process*

2.1. Phase1 : Pixel-wise Contraction and Merging

The primary goal of Phase 1 is to merge similar image pixels into image segments. The processes are operated in a mixed feature space that consists of three subspaces: color space, spatial location space, and CNN score space. An image pixel i on the image is mapped into the feature space $(L_i, a_i, b_i, x_i, y_i, s_{i1}, s_{i2}, \dots, s_{iK})$, where (L,a,b) and (x,y) denote the color values and spatial coordinates of the pixel. On the other hand, the score vector $\boldsymbol{s}_i = (s_{i1}, s_{i2}, \dots, s_{iK})$ denote the class confident scores, which can be regarded as the high-level features extracted by the dense CNN predictor.

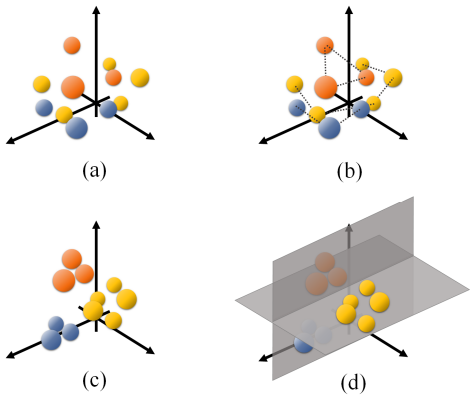

***Fig. 3.** Illustration of the contraction and merging process in the feature space. (a) Initial states. (b) Affinity between pixels. (c) Contracted states. (d) Grid-based merging.*

Contraction: This process aims to pull pixel pairs with similar color appearance closer in the feature space than pixel pairs with less similar color appearance. As to be explained later, this contraction process is very helpful for the subsequent merging process. In our design, the contraction is performed by minimizing the following energy function:

$$E(\tilde{\boldsymbol{\theta}}) = \sum_i^{N_p} \sum_{j \in \mathcal{G}_i} A_1(i,j)(\tilde{\theta}_i - \tilde{\theta}_j)^2 + \lambda_\theta \sum_i^{N_p} (\tilde{\theta}_i - \theta_i)^2, \quad (1)$$

where N_p denotes the total number of image pixels, $\mathcal{G}_i$ denotes the set of neighboring pixels around i, $\theta_i \in \{L_i, a_i, b_i, x_i, y_i, s_{i1}, \dots, s_{iK}\}$, $\tilde{\theta}_i \in \{\tilde{L}_i, \tilde{a}_i, \tilde{b}_i, \tilde{x}_i, \tilde{y}_i, \tilde{s}_{i1}, \dots, \tilde{s}_{iK}\}$, and $\tilde{\boldsymbol{\theta}} = [\tilde{\theta}_1\ \tilde{\theta}_2 \dots \tilde{\theta}_N]^T$. The regularization parameters are set to be $\lambda_L = \lambda_a = \lambda_b = \lambda_{s_{ik}} = 0.01$ and $\lambda_x = \lambda_y = 0.001$. Please refer to [17] for more details of the contraction process.

In this paper, the affinity function $A_1(i,j)$ is defined as below for any two adjacent pixels i and j:

$$A_1(i,j) = exp\left(-\frac{D_1(i,j)}{\rho}\right), \quad (2)$$

where $D_1(i,j)$ is the metric between i and j as defined below:

$$D_1(i,j) = \left\| [L_i\ a_i\ b_i\ ]^T - [L_j\ a_j\ b_j\ ]^T \right\|_2 + \alpha_1 \left\| [s_{i1} \dots s_{iK}]^T - [s_{j1} \dots s_{jK}]^T \right\|_2 \quad (3)$$

Different from [17], here we add in the score information into the affinity function. The score weight α_1 controls the strength of the impact from the score information. Similar to [17], the parameter ρ is adjusted to satisfy the condition that 70% of the $A_1(i,j)$ values in the image is larger than 0.01. The use of ρ is to prevent the contraction process from being overly slow. Besides, since the energy function in (1) is convex, the optimal solution $\boldsymbol{\theta}^{new} = \min_{\tilde{\boldsymbol{\theta}}} E(\tilde{\boldsymbol{\theta}})$ has a closed-form solution and can be efficiently found. More details can be found in [17].

Merging: We use an intuitive merging strategy to merge nearby pixels into groups. Let r_θ denote the range between the upper bound and lower bound of the feature. That is, $r_\theta = \max(\theta) - \min(\theta)$. We uniformly divide the feature space into $\left\lceil \frac{r_L}{k_L} \right\rceil \times \left\lceil \frac{r_a}{k_a} \right\rceil \times \left\lceil \frac{r_b}{k_b} \right\rceil \times \left\lceil \frac{r_x}{k_x} \right\rceil \times \left\lceil \frac{r_y}{k_y} \right\rceil \times \left(\frac{r_{s_1}}{r_{s_1}}\right) \times \ldots \times \left(\frac{r_{s_K}}{r_{s_K}}\right)$ cells and merge all the pixels belonging to the same cell into an image segment. In our experiments, the division parameters are to be $k_L = \left\lceil \frac{r_L}{15} \right\rceil, k_a = \left\lceil \frac{r_a}{15} \right\rceil, k_b = \left\lceil \frac{r_b}{15} \right\rceil, k_x = \left\lceil \frac{r_x}{25} \right\rceil$, and $k_y = \left\lceil \frac{r_y}{25} \right\rceil$. Such merging process can constantly reduce the number of segments while most of the edge information is still preserved.

Remnant Removal: After the contraction and merging process, most regions have been merged into valid segments. However, there may exist a bunch of small segments around the boundary areas which look very noisy. A remnant removal process is thus added to eliminate these noisy pieces. In our algorithm, any small segment whose size doesn't reach a predefined threshold will be merged to one of its adjacent segments that has the most similar color appearance.

2.2. Phase2：Segment-wise Contraction and Merging

Contraction: Similar to Phase 1, each segment R_m is mapped into the feature space with the coordinate $(L_m, a_m, b_m, x_m, y_m, s_{m1}, s_{m2}, \ldots, s_{mK})$. Likewise, we define the energy function as

$$E(\tilde{\boldsymbol{\theta}}) = \sum_m^{N_s} \sum_{n \in \mathcal{G}_m} A_2(R_m, R_n)(\tilde{\theta}_m - \tilde{\theta}_n)^2 + \lambda_\theta \sum_m^{N_s} (\tilde{\theta}_m - \theta_m)^2, \quad (4)$$

where N_s denotes the total number of segments, $\mathcal{G}_m$ denotes the set of neighboring segments of R_m. The affinity function in Phase 2 is defined as

$$A_2(R_m, R_n) = \begin{cases} exp(-D_2(R_m, R_n)/\rho), & N_s > 500 \\ exp(-D_2(R_m, R_n)/\rho) \times CT(R_m, R_n), & N_s \le 500 \end{cases}. \quad (5)$$

In the condition that $N_s > 200$, ρ is adaptively set to make 10% of the affinity larger than 0.01 to maintain the speed of contraction. While $N_s \le 200$, to cooperate with the Type 2 merging, ρ is adaptively set to make only 1% of the affinity larger than 0.01. In Phase 2, the metric between two segments R_m and R_n is defined as

$$D_2(R_m, R_n) = \frac{D_N(R_m, R_n)\left(D_C(R_m, R_n) + \beta D_T(R_m, R_n) + \gamma \widetilde{D_C^B}(R_m, R_n)\right)}{\sqrt{80 + SI(R_m, R_n)}} + \alpha_2 D_S(R_m, R_n) \quad (6)$$

where D_C, $\widetilde{D_C^B}$, D_N, D_T, and SI are the color, boundary-color, region-size, texture, and spatial-intertwining terms, which have been introduced in [17]. In this paper, we propose the inclusion of two additional terms, the score term D_S in (6) and the shape-combination-tendency term CT in (5), to enhance the performance of contraction. Besides, the score weight α_2 in (6) is added to control the significance of the score information. Similar to Phase 1, we perform the contraction operation by finding the optimal solution of (4). That is, $\boldsymbol{\theta}^{new} = \min_{\tilde{\boldsymbol{\theta}}} E(\tilde{\boldsymbol{\theta}})$. In the following, we explain more details about the score term and the shape-combination-tendency term.

(1) **Score term**

The score metric describes the difference between two segments R_m and R_n in the score space:

$$D_S(R_m, R_n) = \|[s_{m1} \ldots s_{mK}]^T - [s_{n1} \ldots s_{nK}]^T\|_2. \quad (7)$$

Based on the above definition, segments with similar semantic information tend to be pulled closer.

(2) **Shape-combination-tendency term**

As the iterations of contraction proceeds, the size of segments gets larger and the shape information becomes more reliable. In (5), we choose to adopt the shape term when $N_s \le 500$. In general, the boundary of an object is expected to be "neat" and less tortuous. In other words, if the combination of two segments generate a segment with curling boundary, we tend to believe this combination is not a proper one. Hence, given the shape map $SP_m(i) = [l_i = m]$ of the segment m, we define the "tortuosity" over the pixel i as

$$T(i|SP_m) = \frac{\sum_{j \in \mathcal{G}_i, G_{dir}(j|SP_m)=1} |G_{dir}(i|SP_m) - G_{dir}(j|SP_m)|}{\sum_{j \in \mathcal{G}_i, G_{dir}(j|SP_m)>0} 1}, \quad (8)$$

where $G_{dir}(p|SP_m) \in [0, 2\pi]$ denotes the gradient direction at the pixel p. Since the tortuosity is only defined on the border, the average tortuosity T_m of the shape map SP_m is defined as

$$T_m = \frac{\sum_{i \in m, G_{dir}(i|SP_m)=1} T(i|SP_m)}{\sum_{i \in m, G_{dir}(i|SP_m)=1} 1}. \quad (9)$$

To measure the improvement of tortuosity after the combination, the shape-combination-tendency is defined as

$$CT(R_m, R_n) = \left(T_m - T_{m,n}\right) \times \frac{N_m^{1/2}}{N_m^{1/2} + N_n^{1/2}} + \left(T_n - T_{m,n}\right) \times \frac{N_n^{1/2}}{N_m^{1/2} + N_n^{1/2}}. \quad (10)$$

The weights $\frac{N_m^{1/2}}{N_m^{1/2} + N_n^{1/2}}$ and $\frac{N_n^{1/2}}{N_m^{1/2} + N_n^{1/2}}$ are used to enhance the stability of large segments.

Type-1 Merging: Type-1 merging is operated under the condition where $N_s > 200$. Like the merging process in Phase 1, Type-1 merging uses the grid-based method to merge all the segments in the same cell. Such a merging method can quickly condense segments into larger ones at the early stage. However, once the total number of segments is lower than 200, the fast merging strategy may sometimes generate unexpected errors.

Type-2 Merging: When $N_s \le 200$, Type-2 merging is adopted, where only one pair of image segments is to be merged in each iteration and the merging metric is defined as

$$D_M(R_m, R_n) = D_2(R_m, R_n) + \|[\tilde{x}_m \; \tilde{y}_m]^T - [\tilde{x}_n \; \tilde{y}_n]^T\|_2. \quad (11)$$

That is, in each iteration, the segment pair that has the smallest merging metric is to be merged.

3. SEGMENT-WISE PREDICTION

At the end of every iteration, a segmentation outcome is generated. The predicted class c_i of a pixel i is inferred as

$$c_i = arg\ \max_k(\tilde{s}_{l_{ik}}).$$

By aggregating all the pixel-wise predictions, a semantic prediction map is produced.

Please note that different semantic prediction maps are generated at different segmentation resolutions N_s. In our observations, the final semantic segmentation map is quite sensitive to segmentation resolution and we empirically fix the segmentation resolution to be 100 image segments to generate the final semantic segmentation result.

4. EXPERIMENTAL RESULTS

Database: Our method is tested over the PASCAL VOC 2012 dataset [21], which comprises 20 object categories and one background category. The dataset contains the training set, validation set, and test set. The training set is augmented by the extra annotations provided by Hariharan *et al.* [22] and the MS COCO dataset [23]. The performance is measured by the mean intersection-over-union (mean IoU) score across the 21 categories.

Implementation Details: In our experiments, we have chosen two dense CNN models, FCN-8s and Dilated CNN, as the dense prediction model in our method. The pre-trained models of FCN-8s and Dilated CNN are obtained from the open sources [24] and [25], respectively. These two models are implemented by Caffe [26] on a single NVIDIA Titan X.

Besides, we use the validation set for parameter optimization. In our experiments, the weights in (3) and (6) are learned to be $\alpha_1 = 5, \alpha_2 = 15, \beta = 3,$ and $\gamma = 6$.

Comparison with State-of-the-art: In our experiments, we compare our approach with DeepLab model [12,13], which is composed by FCN-8s and the vanilla CRF. The scores show that our model can achieve similar performance with respect to the DeepLab model. This proves that the ICM process can be a good substitute of the refinement model. Besides, the CRF-RNN model [14] combines the CRF with the RNN structure and trains the whole network end-to-end. This model can be thought as an intensified version of the vanilla CRF. Our experiments show that the combination of Dilated CNN and ICM can achieve similar performance with respect to the CRF-RNN model. Although the mean IoU score of our approach does not rank the top, the visual observation on the object boundary still demonstrates the superiority of the proposed approach. As mentioned before, the outcomes refined by CRF often possess fragmental boundaries. In comparison, the ICM approach can generate more accurate boundaries in the semantic segmentation maps.

Table 1. *Mean IoU score on PASCAL VOC 2012 test.*

Method	Mean IoU
FCN [5]	62.2%
DeepLab [12,13]	72.7%
Dilated CNN [18] (Context)	73.5%
CRF-RNN [14]	74.7%
FCN+ICM	72.5%
Dilated CNN (Context)+ICM	74.0%

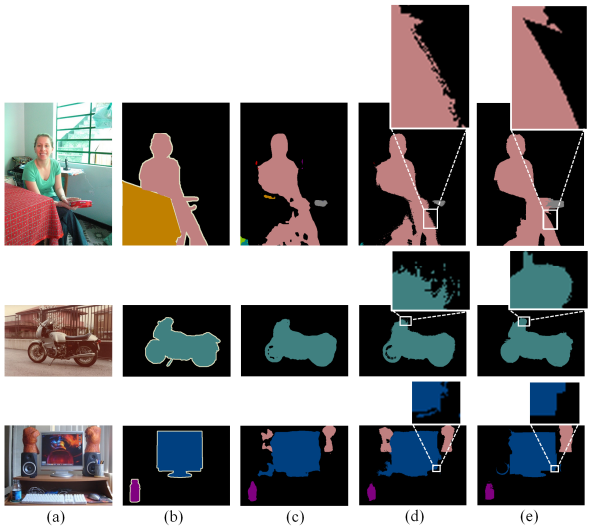

Fig. 4. *Comparison of object boundary. (a) Original image. (b) Ground truth. (c) FCN pre-trained by CRF-RNN* [14]. *(d) CRF-RNN. (e) FCN+ICM.*

5. CONCLUSION

In this paper, we propose a new approach that incorporates the ICM process with the dense CNN classifiers to enhance the performance of semantic segmentation. We have demonstrated that the low-level and mid-level features used by the ICM process can help in improving the accuracy of boundary precision.

Besides, our work reveals a graceful approach to introduce the high-level information into the unsupervised segmentation process. The newly introduced concept of "tortuosity" is also a useful shape feature for bottom-up region merging. With more shape features to be discovered, we believe that the performance of the proposed approach can be further improved in the future.

6. ACKNOWLEDGMENTS

This work was supported by Ministry of Science and Technology, Taiwan. (Grant no. NSC-103-2221-E-009-064-MY3).

7. REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in NIPS, 2012.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *CVPR*, pp. 1-9, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning For Image Recognition," in *arXiv:1512.03385*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning For Image Recognition," in *arXiv:1512.03385*, 2015.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *CVPR*, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *CVPR*, 2014.

[7] B. Hariharan, P. Arbel á ez, R. Girshick, and J. Malik, "Simultaneous Detection and Segmentation," in *ECCV*, 2014.

[8] J. Dai, K. He, and J. Sun, "BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation," in *ICCV*, 2015.

[9] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," in *arXiv:1511.07122*, 2015.

[10] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in *arXiv:1606.00915*, 2016.

[11] Z. Wu, C. Shen, and A. van den Hengel, "High-Performance Semantic Segmentation Using Very Deep Fully Convolutional Networks," in *arXiv:1604.04339*, 2016.

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," in *arXiv:1412.7062*, 2014.

[13] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, "Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation," in *arXiv:1502.02734*, 2015.

[14] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional Random Fields as Recurrent Neural Networks," in *ICCV*, pp. 1529-1537, 2015.

[15] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *ICML*, 2001.

[16] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials," in *NIPS*, 2011.

[17] J. Syn, S. Wang, L. Wung, "Hierarchical Image Segmentation based on Iterative Contraction and Merging," in *IEEE Transactions on Image Processing*, 2017.

[18] P. Arbelaez. "Boundary Extraction in Natural Images Using Ultrametric Contour Maps," in *CVPR*, 2006.

[19] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour Detection and Hierarchical Image Segmentation," in *PAMI*, 2011.

[20] T. H. Kim, K. M. Lee and S. U. Lee, "Learning Full Pairwise Affinities for Spectral Segmentation," in *PAMI*, 2013.

[21] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserma, "The Pascal Visual Object Classes Challenge a Retrospective," in *IJCV*, 2014.

[22] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic Contours from Inverse Detectors," in *ICCV*, 2011.

[23] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft COCO: Common Objects in Context," in *arXiv:1405.0312*, 2014.

[24] https://github.com/torrvision/crfasrnn.

[25] https://github.com/fyu/dilation.

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *ACM-MM*, 2014.