Adversarial Structure Matching Loss for Image Segmentation

Jyh-Jing Hwang 1,2, Tsung-Wei Ke 1, Jianbo Shi 2, Stella X. Yu 1

1 UC Berkeley / ICSI
Berkeley, CA, USA
{jyh, twke, stellayu}@berkeley.edu
2 University of Pennsylvania
Philadelphia, PA, USA
jshi@cis.upenn.edu

Abstract

The per-pixel cross-entropy loss (CEL) has been widely used in structured output prediction tasks as a spatial extension of generic image classification. However, its i.i.d. assumption neglects the structural regularity present in natural images. Various attempts have been made to incorporate structural reasoning mostly through structure priors in a cooperative way where co-occurring patterns are encouraged. We, on the other hand, approach this problem from an opposing angle and propose a new framework for training such structured prediction networks via an adversarial process, in which we train a structure analyzer that provides the supervisory signals, the adversarial structure matching loss (ASML). The structure analyzer is trained to maximize ASML, or to exaggerate recurring structural mistakes usually among co-occurring patterns. On the contrary, the structured output prediction network is trained to reduce those mistakes and is thus enabled to distinguish fine-grained structures. As a result, training structured output prediction networks using ASML reduces contextual confusion among objects and improves boundary localization. We demonstrate that ASML outperforms its counterpart CEL especially in context and boundary aspects on figure-ground segmentation and semantic segmentation tasks with various base architectures, such as FCN, U-Net, DeepLab, and PSPNet.

1 Introduction

The per-pixel cross-entropy loss (CEL) is widely used in structured output prediction tasks such as contour detection, semantic segmentation, and instance segmentation [1,2,3,4] as a spatial extension of generic image recognition [5,6]. However, the disadvantage of CEL is also obvious due to its additive nature and i.i.d. assumption of predictions. As toy examples in Fig. 1 (top block), CEL would yield the same overall error in either situation. However, it is clear that mistakes in either scenario should incur different overall errors, which should be calculated from the structure of entire patch. Therefore, structural reasoning is highly desirable for structured output prediction tasks.

Various attempts have been made to incorporate structural reasoning into structured output prediction in a cooperative way, including two mainstreams, bottom-up Conditional Random Fields (CRFs) [9,10] and top-down shape priors [11,12] or Generative Adversarial Networks (GANs) [13,14]: (1) CRF enforces label consistency between pixels and is commonly employed as a post-processing step [2,15], or as a plug-in module inside deep neural networks [10,16] that coordinate bottom-up information. Effective as it is, CRF is usually sensitive to input appearance changes and needs expensive iterative inference. (2) As an example of learning top-down shape priors, GANs emerge as an alternative to enforce structural regularity in the structured prediction space. Specifically, the discriminator network is trained to distinguish the predicted mask from the ground truth mask.
Figure 1: (Top block) Motivation for ASML: Given incorrect structured predictions, the per-pixel cross-entropy loss (CEL) under i.i.d. assumption applies penalty equally; prior-based losses (such as GANs) penalize more on abnormal structures and thus encourage frequently co-occurring patterns and common shapes; ASML behaves oppositely after learning statistical structural mistakes and thus enables fine-grained structure discriminative power. (Bottom block) Real examples on PASCAL VOC 2012 [7] validation set. PSPNet [8] trained using CEL mostly fails at confusing context (top row) and ambiguous boundaries (bottom row) whereas using our ASML improves these two aspects.

Promising as it is, GANs suffer from inaccurate boundary localization as a consequence of generic shape modeling.

Before we dive into our proposed framework, let us examine the toy examples in Fig. 1 again. Top-down cooperative approaches prefer an additional loss (together with CEL) that penalizes more on the abnormal structures that are deemed undesirable. Such trained networks are thus aware of intra-category shape invariance and inter-category object co-occurrences. However, we notice that in real examples as in Fig. 1 (bottom block), complex and deformable shapes and confusing co-occurrences are the most common mistakes in structured output prediction especially when the visual cues are ambiguous. As a result, training with shape priors sometimes deteriorates the prediction as shown in the bicycle example. We are thus inspired to tackle this problem from an opposing angle: top-down approaches should shift the focus to confusing co-occurring backgrounds or ambiguous boundaries of normal shapes so as to make the structured output prediction network learn harder.

We propose a new framework, which replaces CEL, for training structured prediction networks via an adversarial process, in which we train a structure analyzer to provide supervisory signals, the adversarial structure matching loss (ASML). By maximizing ASML, or learning to exaggerate structural mistakes from the structured prediction networks, the structure analyzer not only becomes aware of complex shapes of objects but adaptively emphasize those confusing co-occurrences. As a result, training structured prediction networks by minimizing ASML reduces contextual confusion among co-occurring objects and improves boundary localization. To improve the stability of training, we append a structure regularizer on the structure analyzer to compose a structure autoencoder. By training the autoencoder to reconstruct ground truth, which contains complete structures, we ensure the filters in the structure analyzer form a good structure basis. We demonstrate that structured output prediction networks trained using ASML outperforms its counterpart CEL on the figure-ground segmentation task on Weizmann horse dataset [17] and the semantic segmentation task on PASCAL VOC 2012 dataset [7] with various base architectures, such as FCN [3], U-Net [18], DeepLab [15].
Figure 2: Framework overview: A structure analyzer extracts structure features (red arrows) from structured output predictions. The structure analyzer is trained to maximize the adversarial structure matching loss (ASML), or discrepancy between structure features extracted from ground truth and from predictions of a segmentation network. The structure analyzer thus learns to exaggerate the structural mistakes and to distinguish fine-grained structures. The segmentation network on the contrary is trained to minimize ASML. To make sure the filters in structure analyzer form a good structure basis, we introduce a structure regularizer, which together with the structure analyzer form a structure autoencoder that is trained to reconstruct ground truth.

and PSPNet [8]. We further verify the effectiveness of ASML particularly on resolving confusing context and improving boundary localization.

2 Related Work

Semantic Segmentation. The field of semantic segmentation has progressed fast in the last few years since the introduction of fully convolutional networks [3]. Both deeper [8, 19] and wider [20, 18, 21] network architectures have been proposed and have dramatically boosted the performance on standard benchmarks like PASCAL VOC 2012 [7]. For example, Yu et al. [21] enabled fine-detailed segmentation results using dilated (i.e., enlarged kernel) convolutions whereas Zhao et al. [8] exploited global context information through pyramid pooling module. Though these methods yield impressive performance w.r.t. mIoU (mean intersection over union), they fail to capture abundant structure information present in natural scenes as shown in Fig. 1.

Structure Modeling. To overcome the aforementioned drawback, people have explored several ways to incorporate structure information [9, 22, 10, 16, 23, 24, 11, 12, 25]. For example, Chen et al. [15] utilized denseCRF [9] as post-processing to refine the final segmentation results. Zheng et al. [10] and Liu et al. [16] further made the CRF module differentiable within the deep neural network. Besides, low-level cues, such as affinity [26, 27, 28, 24] and contour [29, 30] have also been leveraged to encode image structures. However, these methods either are sensitive to transient appearance traits or require expensive iterative inference.

3 Method

We provide an overview of the framework in Fig. 2 and summarize the training procedure in Alg. 1.

3.1 Adversarial Structure Matching Loss

We consider semantic segmentation as an example of structured output prediction tasks, in which a segmentation network (segmenter) $S: x \mapsto \hat{y}$, which usually is a deep CNN, is trained to map an input image $x \in \mathbb{R}^n$ to a per-pixel label mask $\hat{y} \in \mathbb{R}^n$. We propose to train such a segmenter with another network, structure analyzer. The structure analyzer $A: \mathbb{R}^n \mapsto \mathbb{R}^k$ extracts $k$-dimensional multi-layer structure features from either ground truth masks, denoted as $A(y)$, or predictions, denoted as $A(S(x))$. We train the structure analyzer to maximize the distance between the structure features from either inputs, so that it learns to exaggerate structural mistakes made by the segmenter. On the contrary, we simultaneously train the segmenter to minimize the same distance. In other
words, segmenter $S$ and structure analyzer $A$ play the following two-player minimax game with value function $V(S, A)$:

$$
\min_S \max_A V(S, A) = \mathbb{E}_{x, y} \left[ \frac{1}{2} \| A(S(x)) - A(y) \|_2^2 \right].
$$

(1)

that is, we prefer the optimal segmenter as the one that learns to predict the true structures to satisfy structure analyzer. Note that the structure analyzer will bias its discriminative power towards similar but subtly different structures as they occur more frequently through the course of training.

One might relate this framework to GANs [13]. A critical distinction is that GANs try to minimize the data distributions between real and fake examples, and thus accept a set of solutions. Here, structured output prediction tasks require specific one-to-one mapping of each pixel between ground truth masks and predictions. Therefore, the discrimination of structures should take place for every patch between corresponding masks, hence the name adversarial structure matching loss (ASML).

### 3.2 Global Optimality of $S(x) = y$ and Convergence

We would like the segmenter to converge to a good mapping of $y$ given $x$, if given enough capacity and training time. To simplify the dynamic of convergence, we consider both segmenter $S$ and structure analyzer $A$ as models with infinite capacity in a non-parametric setting.

**Proposition 1.** For a fixed $S$, if $S(x) \neq y$, then $\| A^* (S(x)) - A^*(y) \|_2^2$ is infinitely large for an optimal $A$.

**Proof.** If $S(x) \neq y$, there exists an index $i$ such that $S(x)[i] - y[i] = \epsilon$, where $\epsilon \in \mathbb{R} \setminus \{0\}$. Without loss of generality, we assume $S(x)[j] = y[j]$ if $j \neq i$ and let $S(x)[i] = c + \frac{1}{2} \epsilon$ and $y[i] = c - \frac{1}{2} \epsilon$.

We consider a special case where $A_i$ on the $i$-th dimension of the input is a linear mapping, i.e., $D_i(x[i]) = w_i x[i]$. As $A$ is with infinite capacity, we know there exists $A$ such that

$$
\| A(S(x)) - A(y) \|_2^2 \geq \| A_i(S(x)) - A_i(y) \|_2^2 = \| w_i \left( c + \frac{1}{2} \epsilon \right) - w_i \left( c - \frac{1}{2} \epsilon \right) \|_2^2 = |w_i| \epsilon \|
$$

(2)

Note that $\| A_i(S(x)) - A_i(y) \|_2^2 \to \infty$ as $|w_i| \to \infty$. Thus $\| A^* (S(x)) - A^*(y) \|_2^2 \to \infty$. \hfill $\square$

In practice, parameters of $A$ are restricted within certain range under weight regularization so $\| A^* (S(x)) - A^*(y) \|_2^2$ would not go to infinity.

**Corollary 1.** For an optimal $A$, $S(x) = y$ if and only if $A^*(S(x)) = A^*(y)$.

**Proof.** $\Rightarrow$ If $S(x) = y$, $\| A(S(x)) - A(y) \|_2^2 = \| A(y) - A(y) \|_2^2 = 0$, for any $A$. Hence $\| A^*(S(x)) - A^*(y) \|_2^2 = 0$.

$\Leftarrow$ If $A^*(S(x)) = A^*(y)$ or $\| A^*(S(x)) - A^*(y) \|_2^2 = 0$, $S(x) \neq y$ contradicts Proposition 1. Hence $S(x) = y$. \hfill $\square$

**Theorem 1.** If $(S^*, A^*)$ is a Nash equilibrium of the system, then $S^*(x) = y$ and $V(S^*, A^*) = 0$

**Proof.** From Proposition 1, we proved $V(S, A^*) \to \infty$ if $S(x) \neq y$. From Corollary 1, we proved $V(S, A^*) = 0$ if and only if $S(x) = y$. Since $V(S, A) \geq 0$ for any $S$ and $A$, the Nash equilibrium only exists when $S^*(x) = y$, or $V(S^*, A^*) = 0$. \hfill $\square$

From the proofs, we recognize the imbalanced powers between segmenter and structure analyzer where structure analyzers can arbitrarily enlarge the value function if the segmenter is not optimal. In practice, we should limit the training of structure analyzers or apply weight regularization to prevent gradient exploding. Therefore, we train the structure analyzer only once per iteration with a learning rate that is equal to or less than the one for segmenter. Another trick is to binarize the predictions $S(x)$ (winner-take-all across channels for every pixel) before calculating ASML for structure analyzer. In this way, structure analyzers will focus on learning to distinguish the structures instead of the confidence levels of predictions.
Algorithm 1: Algorithm for training semantic segmentation networks using ASML.

for number of training iterations do
  /* Train structure analyzer and regularizer */
  Sample a minibatch with \( m \) images \( \{x^{(1)}, \ldots, x^{(m)}\} \) and segmentation masks \( \{y^{(1)}, \ldots, y^{(m)}\} \).
  (optional) Binarize the structured output predictions \( S(x^{(i)}) \).
  Update the structure analyzer \( A \) and regularizer \( R \) by ascending its stochastic gradient:
  \[
  \nabla_{\theta_A} \frac{1}{m} \sum_{i=1}^{m} \left( \frac{1}{2} \| A \left( S(x^{(i)}) \right) - A(y^{(i)}) \|_2^2 + \lambda y^{(i)} \cdot \log R \left( A_t(y^{(i)}) \right) \right)
  \]

  /* Train segmenter */
  Sample a minibatch with \( m \) images \( \{x^{(1)}, \ldots, x^{(m)}\} \) and segmentation masks \( \{y^{(1)}, \ldots, y^{(m)}\} \).
  Update the segmenter \( S \) by descending its stochastic gradient:
  \[
  \nabla_{\theta_S} \frac{1}{2m} \sum_{i=1}^{m} \left\| D \left( S(x^{(i)}) \right) - D(y^{(i)}) \right\|_2^2
  \]
end

The gradient-based updates can use any standard gradient-based learning rule. Structure analyzer \( A \) should use a learning rate that is equal to or less than segmenter \( S \).

3.3 Reconstructing \( y \) as Structure Regularization

Although theoretically structure analyzers would discover any structural difference between predictions and ground truth, randomly initialized structure analyzers suffer from missing certain structures in the early stage. For example, if filter responses for a sharp curve are initially very low, ASML for the sharp curve will be as small, resulting in inefficient learning. This problem will emerge when training both segmenters and structure analyzers from scratch. To alleviate this problem, we propose a regularization method to stabilize the learning of structure analyzers.

One way to ensure the filters in structure analyzer form a good structure basis is through reconstructing ground truth, which contains complete structures. If filters in structure analyzer fail to capture certain structures, the ground truth mask cannot be reconstructed. Hence, we append a structure regularizer on top of structure analyzer to compose a structure autoencoder. We denote the structure regularizer \( R : A_t(y) \mapsto y \), where \( A_t(\cdot) \) denotes features from the structure analyzer, which are not necessarily the same set as features for ASML; hence the reconstruction mapping: \( R(A_t(y)) \mapsto y \). As a result, the final objective function is as follows

\[
S^* = \arg\min_S \max_{A} \mathbb{E}_{x,y} \left[ \frac{1}{2} \| A(x) - A(y) \|_2^2 \right] + \min_{A,R} \lambda \mathbb{E}_y \left[ -y \cdot \log R(A_t(y)) \right].
\] (3)

Note that the structure regularization loss is independent to \( S \).

4 Experiment

We demonstrate the effectiveness of our proposed methods and compare the results on several popular semantic segmentation architectures trained using CEL or ASML. We first give an overview of the datasets, evaluation metrics, and implementation details used in these experiments. Then we present the main results and analyses on confusion and boundaries.

4.1 Experimental Setup

Tasks and datasets. We compare our proposed ASML against CEL on the Weizmann horse [17] and PASCAL VOC 2012 [7] datasets. The Weizmann horse is a relatively small dataset for figure-ground segmentation that contains 328 side-view horse images, which are split into 192 training and 136 validation images. The VOC dataset is a well-known benchmark for generic image segmentation.
which includes 20 object classes and a ‘background’ class, containing 10,582 and 1,449 images for training and validation, respectively.

**Architectures.** For all the structure autoencoders (i.e., structure analyzer and structure regularizer), we use U-Net [18] architectures with either 7 conv layers for instance segmentation or 5 conv layers for semantic segmentation. We conduct experiments on different segmentation CNN architectures to demonstrate the effectiveness of our proposed method. On horse dataset [17], we use U-Net [13] (with 7 convolutional layers) as our base architecture. On VOC [7] dataset, we carry out experiments and thorough analyses over 3 different architectures with ResNet101 [6] backbone, including FCN [3], DeepLab [15], and PSPNet [8], which is a highly competitive segmentation model. Aside from base architectures, neither extra weight parameters nor post processing are required at inference time.

**Implementation details on Weizmann horse.** We use the poly learning rate policy where the current learning rate equals the base one multiplied by $(1 - \frac{iter}{max_iter})^{0.9}$ with max iterations as 100 epochs. We set the base learning rate as 0.0005 with Adam optimizer for both $S$ and $A$. Momentum and weight decay are set to 0.9 and 0.00001, respectively. We set the batch size as 1 and use no data augmentation other than random mirroring. We set $\lambda = 2$ for structure regularization.

**Implementation details on VOC dataset.** Our implementation follows the implementation details depicted in [15]. We use the poly learning rate policy where the current learning rate equals the base one multiplied by $(1 - \frac{iter}{max_iter})^{0.9}$. We set the base learning rate with SGD optimizer as 0.001 for $S$ and 0.0005 for $A$. The training iterations for all experiments on all datasets are 30K while the performance can be further improved by increasing the iteration number. Momentum and weight decay are set to 0.9 and 0.0005, respectively. For data augmentation, we adopt random mirroring and random resizing between 0.5 and 2 for all datasets. We do not use random rotation and random Gaussian blur. We do not upscale the logits (prediction map) back to the input image resolution, instead, we follow [15]’s setting by downsampling the ground-truth labels for training ($output\_stride = 8$). The crop size is set to 336 × 336 and batch size is set to 8. We update BatchNorm parameters with $\text{decay} = 0.9997$ for ImageNet-pretrained layers and $\text{decay} = 0.99$ for untrained layers. For ASML, we set $\lambda = 10$ for structure regularization.

### 4.2 Main Result

We evaluate both figure-ground and semantic segmentation tasks via mean pixel-wise intersection-over-union (denoted as mIoU) [3]. We first conduct ablation study on both datasets to thoroughly analyze the effectiveness of using different layers of structure features in the structure analyzer. As shown in Table 1 using low- to mid-level features (from conv1 to conv2) of structure analyzers yields the highest performance (79.62% and 71.60% mIoU on Weizmann horse dataset and VOC dataset, respectively). We also report mIoU on VOC dataset using different base architectures as shown in Table 2. Our proposed method achieves consistent improvements across all three base architectures, boosting mIoU by 3.23% with FCN, 0.51% with DeepLab and 1.31% with PSPNet.

| Loss          | Horse mIoU (%) | VOC mIoU (%) |
|---------------|----------------|--------------|
| CEL           | 77.28          | 68.91        |
| ASML (Conv1)  | 78.14          | 70.00        |
| ASML (Conv2)  | 78.15          | 70.70        |
| ASML (Conv1-2)| 79.62          | 71.60        |
| ASML (Conv3)  | 77.79          | 70.85        |
| ASML (Conv1-3)| 78.11          | 69.81        |
| ASML w/o rec. | 77.83          | 72.14        |
| ASML w/o adv. | 76.70          | 71.26        |
| CEL+ASML      | 78.34          | 68.49        |

Table 1: Ablation study of ASML on the Weizmann horse dataset with U-Net and on the PASCAL VOC dataset with FCN. Generally, using low- to mid-level features (from conv1 and conv2) of structure analyzers yield the best performance. It also shows that reconstruction is not always needed if the base network is pre-trained.

| Base / Loss   | mIoU (%) |
|---------------|----------|
| FCN / CEL     | 68.91    |
| FCN / ASML    | 72.14    |
| DeepLab / CEL | 77.54    |
| DeepLab / ASML| 78.05    |
| PSPNet / CEL  | 80.12    |
| PSPNet / GAN  | 80.74    |
| PSPNet / ASML | 81.43    |

Table 2: Experimental results on PASVAL VOC with several base models, FCN [3], DeepLab [15], and PSPNet [8]. The improvements by replacing CEL with ASML are consistent across different base models.
Figure 3: Percentage of each category confused by ‘background’ (the lower the better). The categories with much confusion reduction are ‘chair’, ‘plant’, ‘sofa’, and ‘tv’.

Table 3: Per-category boundary benchmark results of PSPNet [8] trained using various loss functions on PASCAL VOC 2012 [7] validation set. Measurements grouped from top to bottom: precision, recall, and f-measure. ASML performs the best overall on all boundary measurements.

| Loss / Measure | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mean |
|----------------|------|------|------|------|--------|-----|-----|-----|-------|-----|-------|-----|-------|-------|--------|-------|-------|------|-------|----|------|
| **CEL / precision** | 89.37 | 82.62 | 86.87 | 69.82 | 69.29 | 78.59 | 73.84 | 80.36 | 55.80 | 85.85 | 39.06 | 75.58 | 83.18 | 74.63 | 74.85 | 62.86 | 84.72 | 40.79 | 68.68 | 60.68 | 71.99 |
| **GAN / precision** | 88.24 | 84.34 | 91.17 | 67.87 | 65.71 | 83.09 | 76.84 | 60.30 | 71.99 | 65.86 | 84.72 | 40.99 | 68.66 | 83.18 | 74.63 | 74.85 | 62.86 | 84.72 | 40.79 | 68.68 | 60.68 | 71.99 |
| **ASML / precision** | 90.92 | 89.30 | 90.60 | 72.22 | 70.45 | 91.17 | 83.09 | 76.84 | 60.30 | 71.99 | 65.86 | 84.72 | 40.99 | 68.66 | 83.18 | 74.63 | 74.85 | 62.86 | 84.72 | 40.79 | 68.68 | 60.68 | 71.99 |
| **CEL / recall** | 69.41 | 42.05 | 68.85 | 41.67 | 62.37 | 62.59 | 58.16 | 66.72 | 29.72 | 58.74 | 27.87 | 68.84 | 60.78 | 51.26 | 55.80 | 41.74 | 54.73 | 41.74 | 55.08 | 51.26 | 55.08 |
| **GAN / recall** | 67.49 | 40.26 | 62.95 | 37.43 | 58.89 | 61.12 | 55.01 | 63.24 | 28.60 | 56.07 | 22.34 | 65.58 | 59.39 | 49.45 | 53.64 | 42.34 | 53.12 | 41.73 | 53.92 | 42.20 | 60.20 |
| **ASML / recall** | 69.75 | 48.62 | 63.56 | 40.26 | 62.58 | 63.73 | 57.09 | 65.93 | 38.07 | 59.10 | 29.39 | 66.71 | 62.19 | 52.37 | 57.33 | 42.90 | 54.77 | 49.69 | 55.24 | 55.84 | 63.96 |
| **CEL / f-measure** | 78.13 | 55.73 | 74.91 | 52.39 | 65.65 | 69.68 | 63.00 | 72.91 | 38.78 | 69.78 | 32.52 | 70.83 | 70.24 | 60.78 | 63.93 | 54.58 | 66.50 | 41.14 | 61.12 | 55.84 | 59.91 |
| **GAN / f-measure** | 76.48 | 54.30 | 74.48 | 48.25 | 62.11 | 70.45 | 64.10 | 70.12 | 37.53 | 68.32 | 32.89 | 70.45 | 70.45 | 60.11 | 63.20 | 56.37 | 60.74 | 61.01 | 52.24 | 58.74 |
| **ASML / f-measure** | 78.94 | 63.01 | 75.33 | 57.71 | 66.24 | 71.68 | 67.13 | 72.41 | 46.26 | 70.45 | 37.80 | 69.52 | 72.77 | 62.14 | 66.85 | 57.01 | 67.04 | 46.96 | 63.22 | 61.80 | 62.35 |

ASML is also 0.71% higher than GANs (together with CEL) on VOC dataset. We demonstrate some visual results in Fig. 4.

### 4.3 Confusing Context Improvement

We next demonstrate the robustness of our proposed method under confusing context. We first calculate the confusion matrix of pixel accuracy on PASCAL VOC 2012 [7] validation set. We identify that ‘background’ is biggest confuser for most of the categories and hence we summarize the percentage of confusion in Fig. 3 (i.e., the ‘background’ column from the confusion matrix). ASML reduces the overall confusion caused by ‘background’ from 19.4% to 14.45% on FCN and from 12.5% to 11.5% on PSPNet with 8.7% relative error reduction. Large improvements come from resolving confusion of ‘chair’, ‘plant’, ‘sofa’, and ‘tv’.

### 4.4 Boundary Localization Improvement

We argue that our proposed method is more sensitive to complex shapes of objects. We evaluate boundary localization using standard contour detection metrics [32]. The contour detection metrics compute the correspondences between prediction boundaries and ground-truth boundaries, and summarize the results with precision, recall, and f-measure. We compare the results with different loss functions: CEL, GAN and ASML on VOC validation set. Shown in Table 3, ASML outperforms both CEL and GAN among most categories and overall. Also note that the boundaries of thin-structure objects are much better captured by ASML, such as ‘bike’ and ‘chair’.
Figure 4: Visual results on VOC PASCAL 2012 validation set. Left to right: Images, Predictions from PSPNet trained using cross-entropy loss (CEL), GAN, ASML, and ground truth masks.
References

[1] Bertasius, G., Shi, J., Torresani, L.: Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In: CVPR. (2015)

[2] Hwang, J.J., Liu, T.L.: Pixel-wise deep learning for contour detection. In: ICLR Workshop. (2015)

[3] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)

[4] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: CVPR. (2017)

[5] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)

[7] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)

[8] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. (2017)

[9] Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS. (2011)

[10] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: ICCV. (2015)

[11] Xie, S., Huang, X., Tu, Z.: Top-down learning for structured labeling with convolutional pseudoprior. In: ECCV. (2016)

[12] Gygli, M., Norouzi, M., Angelova, A.: Deep value networks learn to evaluate and iteratively refine structured outputs. In: ICML. (2017)

[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)

[14] Luc, P., Couprie, C., Chintala, S., Verbeek, J.: Semantic segmentation using adversarial networks. NIPS Workshop (2016)

[15] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915 (2016)

[16] Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: CVPR. (2015)

[17] Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: ECCV. (2002)

[18] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. (2015)

[19] Li, X., Liu, Z., Luo, P., Loy, C.C., Tang, X.: Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In: CVPR. (2017)

[20] Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: CVPR. (2015)

[21] Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR. (2016)

[22] Chen, L.C., Schwing, A., Yuille, A., Urtasun, R.: Learning deep structured models. In: ICML. (2015)

[23] Lin, G., Shen, C., van den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: CVPR. (2016)

[24] Bertasius, G., Torresani, L., Yu, S.X., Shi, J.: Convolutional random walk networks for semantic image segmentation. In: CVPR. (2017)

[25] Ke, T.W., Hwang, J.J., Liu, Z., Yu, S.X.: Adaptive affinity field for semantic segmentation. arXiv preprint arXiv:1803.10335 (2018)

[26] Shi, J., Malik, J.: Normalized cuts and image segmentation. TPAMI (2000)
[27] Maire, M., Narihira, T., Yu, S.X.: Affinity cnn: Learning pixel-centric pairwise relations for figure/ground embedding. In: CVPR. (2016)

[28] Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: NIPS. (2017)

[29] Bertasius, G., Shi, J., Torresani, L.: Semantic segmentation with boundary neural fields. In: CVPR. (2016)

[30] Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: CVPR. (2016)

[31] Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)

[32] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. TPAMI (2011)