Semantic Segmentation via Structured Patch Prediction, Context CRF and Guidance CRF

Falong Shen[1] Rui Gan[1] Shuicheng Yan[2,3] Gang Zeng[1]
[1]Peking University [2]360 AI Institute [3]National University of Singapore
{shenfalong,raygan,zeng}@pku.edu.cn, yanshuicheng@360.cn

Abstract

This paper describes a fast and accurate semantic image segmentation approach that encodes not only segmentation-specified features but also high-order context compatibilities and boundary guidance constraints. We introduce a structured patch prediction technique to make a trade-off between classification discriminability and boundary sensibility for features. Both label and feature contexts are embedded to ensure recognition accuracy and compatibility, while the complexity of the high order cliques is reduced by a distance-aware sampling and pooling strategy. The proposed joint model also employs a guidance CRF to further enhance the segmentation performance. The message passing step is augmented with the guided filtering which enables an efficient and joint training of the whole system in an end-to-end fashion. Our proposed joint model outperforms the state-of-art on Pascal VOC 2012 and Cityscapes, with mIoU(%) of 82.5 and 79.2 respectively. It also reaches a leading performance on ADE20K, which is the dataset of the scene parsing track in ILSVRC 2016. The code is available at `https://github.com/FalongShen/SegModel`.

1. Introduction

Semantic segmentation is a fundamental but difficult problem in computer vision. Compared with image classification, it provides a pixel-wise semantic understanding of the image, through which the scene is parsed in terms of object categories, locations and shapes. Deep networks have made a series of breakthroughs on the task of image classification [18, 14, 13]. Convolutional neural networks (CNN) controlled by varying depth and breadth provide powerful models, and the integrated multi-level hierarchical features and classifiers embed mostly correct prior knowledge about statistics and dependencies among pixels for preventing overfitting.

Recent advances in semantic segmentation mainly rely on fully convolutional networks (FCN) and conditional random fields (CRF) [4, 33, 1, 31, 7]. FCN transfers the recognition network in image classification by fine-tuning position-aware feature representations for semantic segmentation [24]. However, deeply learned features for image classification tend to tolerate the object translation and deformation by resolution-reducing pooling layers and sub-sampling layers in convolutional neural networks [13], which decreases the ability for locating and separating objects from neighboring contexts. In order to determine object positions and boundaries, a *bilinear up-sampling* operation is often adopted to retrieve a pixel-wise prediction in FCN, which leads to an interpretation ambiguity between the degrees of belief and its frequentist counterpart.

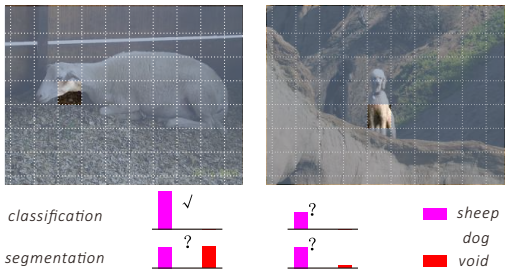


Figure 1: The belief-frequency ambiguity when transferring model from classification to segmentation. The right image is a hard example and both models produce a confusing prediction. The left image is an easy example, the segmentation model still produces a confusing prediction in order to make spatial prediction.

Interpretation of Local Prediction An analysis on the end-to-end training process of FCN in Sec. 3.1 shows that the *softmax* classifier produces a distribution to represent not only the degrees of belief about object categories, but also the frequentist of the category in the patch. As shown in Fig. 1, the interpolation routine in the up-sampling operation seems to treat the classification scores with both the belief and the frequentist interpretation. That says, the FCN classifier training uses a ambiguous criteria, with both image region statistics and training sample likelihood. This

double meanings interpretation is most obvious when interpolating classification scores across object boundaries and predicting the difficult samples. The ambiguous prediction prevents accurate and detailed object shapes being captured by the latter steps in segmentation. We propose to resolve this ambiguity by a structured patch prediction technique in Sec. 3.

Along the other direction of literature, probability graphic models have been widely used for structured prediction tasks. In particular, CRF has observed widespread success in semantic segmentation [19, 28, 17] thanks to their abilities in encoding high order conditional dependence among node labels given the appearances. However, learning CRF requires many repeated inference steps, and it is time consuming [35, 19]. Our work focuses on fusing the aforementioned discriminative features of FCN with the structured prediction capability of CRF, with emphases on both effective high order context constraints and scalable end-to-end joint training efficiency.

Context Compatibility Context clue represents the spatial-relationship between category labels and plays an important role in structured prediction tasks. It has been noted that context clue or high-order information is vital in object detection and semantic image segmentation [29, 19]. Through minimizing the Gibbs energy, CRF is widely adopted for harnessing the context clue to make structured prediction. However, these models are quite limited due to the time cost of graph inference for the derivation of partition function in each update of gradient descent [35]. Compared to the traditional CRF approach, auto-context [29] encoded the joint statistics by a series of classifiers based on the label context. For each classifier, the output of the last classifier is used as feature. Auto-context made an attempt to recursively select and fuse context label for structured prediction. Another probability of encoding context information is learning the messages based on feature context [20, 27]. The kind of feature context methods model the message estimator between each pair by stacking unary features, which is more similar to traditional CRF as they both rely on pair-wise message passing. We enforce prior structure knowledge with both label and feature contexts, and propose a distance-aware sampling and pooling strategy to reduce the complexity of high order cliques, as discussed in Sec. 4.

Boundary Guidance Low level features, such as image edges, texture and appearance homogeneity often help to obtain a clear and sharp boundaries around objects. Recently bilateral-filtering based CRF is popularly adopted for boundary localization. Combined with strong recognition capacity of convolutional neural network, bilateral CRF has shown remarkable success in addressing the task of sharp boundary around object [3, 25, 2]. Besides, Liu *et al.* [23] proposed a filter similar to the bilateral filter which can be processed on graphic process unit efficiently through the locally convolutional layer. We choose to augment the *message passing* with the guided filtering [12, 11], not only because of its edge-preserving property, but also due to its linear time complexity regardless of the kernel size. This leads to a fast training process with high performance as described in Sec. 5.

Theoretically our learning approach of context CRF resembles the error-correcting iterative decoding methods in [27, 29], since we use a series of classifiers to encode interactions between each node instead of the explicit global probability representation. From an alternative view for the *message passing* in the mean field algorithm, updating the marginal distribution is to collect messages from neighborhood regions. Thus, an effective message estimator can directly model region features consisting of information from estimated labels and deep convolutional features. This equivalent message view is the key of our efficient solver to the joint FCN and CRF model, and the details will be discussed the following sections.

The main contributions of this paper have four folds.

- We propose a joint objective to integrate segmentation-specified features, high order context and boundary guidance for accurate semantic segmentation. The proposed model reaches leading performances on three dominating segmentation benchmark datasets.
- A structured patch prediction technique is introduced for spacial filling. While keeping the feature abstraction at a relatively high level, it substitutes the over-smoothed interpolation operation and partially resolves the belief-frequency ambiguity.
- A distance-aware context is proposed to embed both label and feature compatibility while avoiding the price of high complexity. The corresponding context CRF can be efficiently optimized with little time costs while bringing large performance gains.
- We also introduce a guidance CRF to further enhance the segmentation accuracy. The *message passing* step is augmented with the guided filtering which allows efficient joint training of the whole system in an end-to-end fashion.

2. Our Proposed Method

Let $I \in \mathbf{I}$ denote one input image and $\mathbf{x} \in \mathcal{X}$ is its segmentation label assignment. Each pixel i in the label assignment $\mathbf{x} = \{x_i, i = 1, ..., N\}$ takes a value from a predefined label set $\mathcal{L} = \{1, ..., L\}$. The conditional likelihood function of a label assignment $\mathbf{x}$ for an image I is

$$P(\mathbf{x}|I;\theta) = \frac{1}{Z(I;\theta)} \exp[-E(\mathbf{x}, I;\theta)], \qquad (1)$$

where θ denotes the model parameters and $E(\mathbf{x}, I; \theta)$ is the Gibbs energy function. $Z(I;\theta) = \sum_{\mathbf{x}} \exp[-E(\mathbf{x}, I; \theta)]$ is the partition function conditioned on the image I. Our energy function takes the form

$$E(\mathbf{x},I;\theta) = E_{local}(\mathbf{x},I;\theta) + E_{context}(\mathbf{x}, I;\theta) + E_{edge}(\mathbf{x},I;\theta), \quad (2)$$

where $E_{local}(\mathbf{x}, I; \theta)$ denotes the unary score regarding to the appearance within the local neighborhood, $E_{context}(\mathbf{x}, I; \theta)$ encodes the context clue for structure prediction, and $E_{edge}(\mathbf{x}, I; \theta)$ encourages the concurrence between the segmentation boundaries and intensity edges.

The coarse segmentation feature map $f(\mathbf{x}|I)$ built by FCN has a much lower resolution than the original input image. Instead of up-sampling using a transposed convolutional layer with a large filter size (*e.g.*, 32 for $16\times$ model), we propose a multi-stage solution to resolution enhancement. Firstly we introduce a structured patch prediction technique (Sec. 3) for spacial filling at a certain interim resolution

$$f \mapsto E_{local}. \quad (3)$$

The context potential is also taken into consideration at this level to make structured prediction,

$$E_u(\mathbf{x}, I; \theta) = E_{local}(\mathbf{x}, I; \theta) + E_{context}(\mathbf{x}, I; \theta). \quad (4)$$

In order to perform guidance CRF and compute per-pixel entropy loss, we need to decouple each x_i in this step, *i.e.*, marginal potential with regard to each x_i. This task is solved in the context CRF component (Sec. 4).

Then we further up-sample the segmentation score map by transposed convolution with learnable parameters. Combined with the edge potential, the final total energy function is

$$E(\mathbf{x}, I; \theta) = E_u^{\uparrow}(\mathbf{x}, I; \theta) + E_{edge}(\mathbf{x}, I; \theta), \quad (5)$$

where $E_u^{\uparrow}(\mathbf{x}, I; \theta)$ is the decoupled score map from context CRF after up-sampling. Combined with the edge potential, the segmentation score map is refined with the guidance CRF (Sec. 5) and we can get a more accurate object boundary through an end-to-end joint training.

3. Transfer Model via Structured Patch Prediction

FCN combined with the hole algorithm produces a coarse segmentation prediction, which is followed by a *bilinear up-sampling* operation to make high resolution prediction. This flowchart is widely adopted in previous semantic segmentation literature [24, 3]. However, it inevitably encounters the belief-frequency ambiguity depicted in Fig. 1. We give a theoretical explanation of this ambiguity and provide a solution in this section.

3.1. Theoretical Analysis on Up-sampling Operation

The coarse score map is up-sampled to the original input image size by a fixed *bilinear up-sampling* layer in most of previous works. The unary feature f_i from FCN for patch i is converted to a label score q_i to describe the probability for the existence of each category in this patch. The score q_i is bilinear up-sampled $16\times$ larger to compute pixel-wise entropy loss with the ground truth label. It means all the prediction results in the patch i are summarised and compressed in the vector q_i. Let p_i^j (one-hot vector) denote the ground truth label in the patch i for the jth position, and let w_j be the corresponding bilinear weight. The ground truth distribution for this patch is $\sum_j w_j p_i^j$ as shown in Fig. 2, and thus the training target is

$$D(q_i || \sum_j w_j p_i^j). \quad (6)$$

It is important to notice this fundamental difference between image classification and semantic image segmentation by FCN. The target distribution of segmentation is never a one-hot vector for FCN. Instead, it is a weighted sum of all the presented categories in this patch. The prediction score q_i describes not only the existence of objects in certain category but also the portion of pixels in this category. While for classification, q_i only represents the belief for the existence of a certain kind of object in the image.

As shown in Fig. 1, the two-fold of the probability q_i in FCN causes the ambiguity, especially on the border of the object and for difficult input image. While the segmentation model of FCN is expected to describe both the existence of the category and the portions of the category in the patch, it lacks the ability to tell the difference.

From the perspective of information flow, the bottleneck of $C \times 1 \times 1$ coarse score block is the bridge between the $D \times 1 \times 1$ feature block and the $C \times 16 \times 16$ dense score map. The information is heavily compressed as a C-dimensional vector where most of the spatial information has lost.

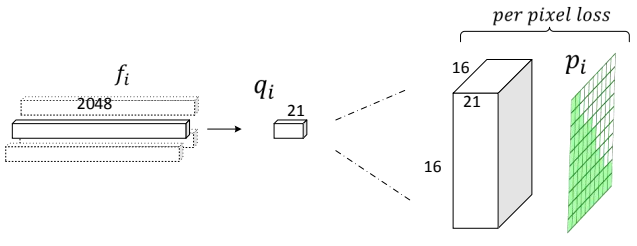


Figure 2: The 2048-D feature vector goes through a 21-D bottle neck before up-sampling to 16×16, which leads to heavily information loss.

3.2. Structured Patch Prediction

Instead of extracting as much information as possible in the patch through a $C \times 1 \times 1$ vector, we look forward a more effective bridge between the unary feature and the patch label score map. However, a direct connection from the long feature to the dense prediction needs enormous parameters and will be difficult for training. We need to make a trade-off between the number of parameters and representation abilities.

We propose to model the label score map by a structured patch prediction technique. The feature f_i is used to produce a $C \times n \times n$ score map[1], which is a transition to pixel-wise dense prediction. The information in the whole image patch is coarsely depicted in a small label patch. The belief-frequencist ambiguity is partially solved by the structured patch prediction. As recognition and localization are basically two tasks, we explicitly divide them by introducing more classifiers for each position of the patch.

Our intuition behind the structured patch prediction relies on the fact that the FCN feature models the spatial coherence of a local region. Previous works have also proved this idea [9, 26, 8]. The FCN feature can not only recoginize the category label in the patch, but also be aware of the context label structure. For example, R-CNN proposed to regress a bounding box to properly crop the object, and it also took the advantage of the spatial localization ability of CNN features [9]. In our experiments, we directly make use of the FCN features to assign label to each position in the patch.

4. Context Modeling with Conditional Random Field

Given an image I, the aforementioned structured patch prediction technique provides a segmentation score map. The Gibbs energy of the label assignment $\mathbf{x} \in \mathcal{L}^N$ is

$$E_u(\mathbf{x}, I; \theta) = \sum_i \phi_i(x_i, I_i; \theta) + \sum_c \psi_c(\mathbf{x}_c, I_c; \theta), \quad (7)$$

where $\phi_i(x_i, I_i; \theta)$ is the singleton node potential for assigning x_i to pixel i based on the local appearance descriptor by the structured patch prediction. ψ_c is defined on the high order clique c. I_i and I_c denote the local image regions around the position i and clique c respectively.

Our goal is to estimate the marginal potentials to approximate $E_u(\mathbf{x}, I; \theta)$, which is

$$\sum_i \phi_i(x_i, I_i; \theta) + \sum_c \psi_c(\mathbf{x}_c, I_c; \theta) \approx \sum_i \phi_i^u(x_i, I_i; \theta). \quad (8)$$

[1]In our experiments $n = 2$. Therefore we still need the up-sampling operation.

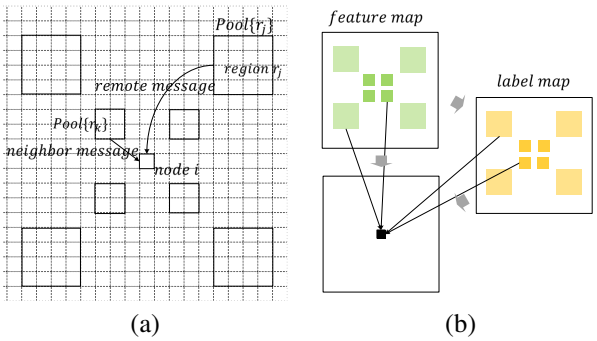


Figure 3: Illustration of context CRF. (a) We exploit a quite large field (28×28 on the feature map) to collect context information. The messages from neighbor regions and remote regions are pooled with different size in order to avoid over-fitting. (b) Both feature map and score map are exploited to produce messages.

In the following paragraphs we will introduce the construction of the high order context term $\psi_c(\mathbf{x}_c, I_c; \theta)$ in our formulation and how to implement Equation (8) efficiently.

4.1. Distance-aware High Order Context

The context term $\psi_c(\mathbf{x}_c, I_c; \theta)$ provides information surrounding the current patch, which is important for structured prediction. The natural images are highly spatial related, and the neighboring patches are more closely related than remote ones. We propose a distance-aware sampling strategy for context modeling as shown in Fig. 3(a). The context patches are divided in groups according to their distances to the centering patch. The remote patches are pooled in large areas to accumulate the weak evidences for a more robust representation of the correlation. We make use of the distance prior of context in order to avoid over-fitting in the training stage.

4.2. Message from High Order Term

Following the similar derivations of the mean field algorithm [16], we employ an iteration algorithm to approximate Equation (8)

$$\phi_i^u(x_i) = \phi_i(x_i, I_i; \theta) - \sum_c \mathbb{E}_{\hat{p}(\mathbf{x}_{c\setminus i})}[\psi_c(\mathbf{x}_{c\setminus i}, x_i, I_c; \theta)]. \quad (9)$$

The second term is the expectation of $\psi_c(\mathbf{x}_{c\setminus i}, x_i, I_c; \theta)$ over the estimated distribution of $\hat{p}(\mathbf{x}_{c\setminus i})$, which is about the messages passed from the high order clique c to the local node i. It is a C-dimensional vector encoding the information of label distribution, which is difficult to get a analytical solution. Lin *et al.* [19] has tried to learn potential functions for each two-nodes clique, but the inference is much slower and costs lots of memory, *e.g.*, it requires L^2 outputs for each pair-wise clique, and for a N-nodes graph there are up to N^2 pair-wise cliques. It is even much more

difficult to learn a potential function for high order clique with more than two nodes.

In order to model the high order clique, instead of calculating the marginalization with regard to $\hat{p}(\mathbf{x}_{c\setminus i})$, we propose to construct the convolutional neural networks and directly learn the messages. As show in Fig. 3(b), we place several convolutional layers on both the estimated probability map $\hat{p}(\mathbf{x}_c)$ and the context feature map f_c to capture the high order pattern

$$\mathbb{E}_{\hat{p}(\mathbf{x}_{c\setminus i})}[\psi_c(\mathbf{x}_{c\setminus i}, x_i, I_c; \theta)] = U[\hat{p}(\mathbf{x}_c), x_i, f_c; \theta], \quad (10)$$

where $U[\hat{p}(\mathbf{x}_c), x_i, f_c; \theta]$ is a scalar describing the compatibility of x_i in the high order clique assignment $\mathbf{x}_c$ based on context feature f_c. This message term can also be treated as a new classifier based on the estimated probability map from the previous iteration and the context image feature.

5. Boundary Guidance with Conditional Random Field

We have exploited the structured patch prediction technique to enhance the density of FCN features, and we have encoded the context information to enforce context compatibility. Both improvements can't be afforded in the high resolution due to the sensibility of patch-based features and the complexity of high-order potentials. To obtain the detailed object boundaries, we further refine the segments with a guidance CRF at high resolution. The fully connected CRF with low level image features, *e.g.*, color, coordinate, has been successfully used to enhance the object localization accuracy [3, 19].

Simply bilinear up-sampling the score map often leads to a misalignment between predicted object boundaries and color edges. The guided filtering is an edge-preserving technique with nice visual quality and fast speed [12]. We propose to combine pair-wise CRF with guided filtering and jointly learn the whole networks to align the segmentation with the color boundaries on images.

The guided filtering in our guidance CRF takes two inputs: (1) the coarse segmentation score map $\phi^{u\uparrow}$ to be filtered and (2) the down-sampled[2] color image I. The filtering result is

$$g(x_i) = \sum_j w_{ij}(I)\phi^{u_j\uparrow}(x_j), \quad (11)$$

where $\phi^{u_j\uparrow}(x_j)$ is up-sampled from the output of context CRF. The weight w_{ij} depends on the input color image I, which is used as the guidance image. Following the similar derivations in [12], the expression of w_{ij} is

$$w_{ij} = \frac{1}{|\omega|^2}\sum_k \left[1 + (\Sigma_k + \epsilon U)^{-1}\sum_{c=1}^{3}(I_i^c - \mu_i^c)(I_j^c - \mu_j^c))\right] \quad (12)$$

where μ_k and Σ_k is the mean and 3×3 covariance matrix of image I in window ω_k, U is 3×3 identity matrix and $|\omega|$ is the number of pixels in ω_k. ϵ is a regularized parameter and we set it to 1 throughout our experiments.

Algorithm 1 Guidance CRF

Forward

input: Down-sampled Guidance image I, segmentation score map ϕ^u, compatibility matrix μ, weight parameter λ,maximum iteration k_{max}, $k=0$, $\phi^0 = \phi^u$.

while $k < k_{max}$

1. $q^k(x_i) = \frac{1}{Z_i}\exp[-\phi^k(x_i)]$. ▷ Softmax
2. $g^k(x_i) = \sum_j w_{ij}(I)q^k(x_j)$ ▷ Guided filtering
3. $m^k(x_i) = \sum \mu(x_i, x_j)g^k(x_j)$ ▷ Compatibility transform
4. $\phi_i^k(x_i) = \phi_i^u(x_i) - \lambda m^k(x_i)$ ▷ Local update
5. $k = k+1$

endwhile

output: marginal potential ϕ^b

Now we will introduce how to combine the pair-wise CRF with guided filtering. From the aforementioned sections, we have

$$E_u^{\uparrow}(\mathbf{x}, I; \theta) = \sum_i \phi^{u_i\uparrow}(x_i). \quad (13)$$

Substitute it in Equation (2), the energy of a label assignment $\mathbf{x}$ is given by

$$E(\mathbf{x}) = \sum_i \phi^{u_i\uparrow}(x_i) + \sum_{i<j}\psi_p(x_i, x_j, I_i, I_j), \quad (14)$$

where the unary potential $\phi^{u\uparrow}$ is the output of context CRF and up-sampled by structured patch prediction. The pair-wise potential ψ_p in the fully connected CRF has the form

$$\psi_p(x_i, x_j, I_i, I_j) = \mu(x_i, x_j)k(I_i, I_j) \quad (15)$$

where μ is the label compatibility function with the kernel $k(I_i, I_j) = w_{ij}$ defined in Equation (12). μ is initialized by Potts model. A mean-field algorithm is used to approximate the marginal distribution as shown in Algorithm 1.

The forward pass in the training stage performs a *softmax* layer, a *message passing* layer, a *compatibility transform* layer and a *local update* layer in each iteration. We run three iterations throughout our experiments by cross validation. As it is shown in Algorithm 1, all of these steps can

[2]We once experimented with original image but later found down-sampled ($4\times$) image leads to a faster training and more stable solution.

be described by CNN layers. The parameters of the guided filter depend on the spatial and appearance of the original image. Instead of direct computation by convolutional layers, the message passing step can be executed as one guided filtering, which can be computed very efficiently. Finally the marginal distribution from guidance CRF is bilinear upsampled to the original image resolution.

To back-propagate the segmentation error differentials *w.r.t* its input and network parameters in each layer, it is straightforward to perform back-propagation algorithm through the *local update* layer, the *compatibility transform* layer and the *softmax* layer. For the *message passing* layer, the gradient *w.r.t* its input is

$$\frac{\partial L}{\partial g}(x_i) = \sum_j w_{ij}(I)\frac{\partial L}{\partial q}(x_j), \tag{16}$$

which can also be calculated by performing guided filtering on the error differential map $\frac{\partial L}{\partial q}(x_j)$.

6. Optimization

Given a training set $\{(I, \mathbf{x}^*), I \in \mathbf{I}, \mathbf{x}^* \in \mathcal{X}\}$, the target of FCN and CRF optimization is to learn the parameters θ^* to maximize the posterior probability of the training data,

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_I \sum_i \log \hat{p}(x_i^*|I;\theta) + \frac{\lambda}{2}||\theta||_2^2. \tag{17}$$

Here I is the training image and x_i^* is the ground truth segmentation label for pixel i in this image; λ is the weight decay parameter. The program can be optimized efficiently by the standard stochastic gradient descent solver and the whole framework is shown in Fig. 4.

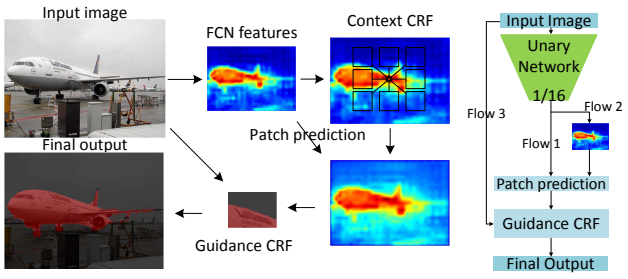


Figure 4: Schematic visualization of our model. The left figure is the pipeline of our proposed model. The context CRF is performed on both the coarse FCN feature map and the score map to encode context information and produce a structured patch prediction. At the fine level, we delineate the object boundary by guidance CRF. The right figure depicts the network structure. Each data stream is assigned a flow number in our library, which makes it memory-efficient.

7. Experiments

We evaluate the proposed model on three challenging segmentation benchmark datasets. We comparing our model with the state-of-the-art works. ADE20K is a new introduced dataset in ILSVRC 2016 and our model joined in the competition. Our models obtain leading performance on all the three datasets while having efficient running speed. The ablative study is done on Pascal VOC 2012 as it is the most widely used dataset in semantic image segmentation.

7.1. Datasets and Implementation

7.1.1 Datasets

Pascal VOC 2012 [6] dataset is a popular segmentation benchmark. It includes 20 categories plus background. The original *train* set has 1464 images with pixel-wise labels. We also use the annotations from [10], resulting in 10582 (augmented *train* set), 1449 (*val* set) and 1456 (*test* set) images. The accuracy is evaluated by mean IoU scores.

Cityscapes [5] dataset consists of 2975 training images and 500 validation images . Both have pixel-wise annotations. There are also another about 19,998 image with coarse annotation. There are 19 categories in this dataset and there is no background category. All the images are about street scene in some European cities and are taken by car-carried cameras. It should be noticed that the size of every image is 1024×2048 in this dataset.

ADE20K [36] datset is divided into 20,000 images for training, 2000 images for validation, and another batch of held-out images for testing. Every image in this dataset is annotated with pixel-wise label. There are totally 150 semantic categories included in the challenge for evaluation.

Auxilliary dataset. To compare with the-state-of-the-art, sometimes we further exploit the large scale dataset MS COCO [21] to pre-train the model, which includes 123,287 images in its *trainval* set with 80 categories and one background. Each image comes with pixel-wise label.

7.1.2 Implementation

We use the public Caffe [15] framework for deep learning but we have made lots of changes. We adopt a data-flow based memory management strategy. In the inference stage, the data blobs in the same flow share the same pieces of GPU memory. In the backward pass of the training stage, the gradient blobs in the same flow also share the GPU memory.

Training settings and parameters. We skip the subsampling operation in the $conv5\_1$ layer in Resnet-101[3] and modify the filters in the $conv5$ block by introducing zeros to increase the size, which is known as "hole algorithm"

[3]The base model is pubic available at `https://github.com/tornadomeet/ResNet`. We always use it as base model without specific notation.

Table 1: Results on Pascal VOC 2012 *test* set and Cityscapes *test* set. Measured by the mean IoU (%). Both of our submitted models are fine-tuned from Resnet-101 and exploit MS-COCO.

Method	PasVOC12	CityScapes
DPN[23]	77.5	66.8
Dilation10[33]	-	67.1
Adelaide_context[19]	77.8	71.6
Adelaide_VeryDeep[31]	79.1	-
LRR_4x[7]	79.3	71.8
DeepLab-v2[4]	79.7	70.4
CentraleSupelec Deep G-CRF[1]	80.2	-
SegModel	82.5	79.2

Table 2: Results on ADE20K *val* set and *test* set. Measured by the average of mean IoU and pixel accuracy (%). Our models are trained on ADE20K *train* set, without resorting to MS-COCO or Place365. The performance on the *val* set is evaluated by a single model.

Method	*val*	*test*
CRFasRNN[35]	-	47.0
ACRV-Adelaide[19]	-	53.3
Hikvision	60.4	53.4
CASIA_IVA	-	54.3
SegModel	61.2	54.5
360+MCG-ICT-CAS_SP	-	55.6
Adelaide[31]	-	56.7
SenseCUSceneParsing[34]	63.1	57.2
post competition		
SegModel	61.7	-

[3]. This operation yields a stride of 16 pixels and we name it a $16\times$ model in this paper. It should be noticed that the $16\times$ model is much faster than the $8\times$ model in both the training stage and inference stage. Weight decay parameter is set to 0.0001 and the momentum parameter is set to 0.9. The initial learning rate is set at 0.01 and "ploy" strategy is adopted [22, 4]. The mini-batch size is set to 16. Half of all pixels in each batch with larger loss are kept for loss computation [31, 32]. We run several epoches at the end of training stage to compute the batch normalization statistics. Scale jittering, color altering [30] and horizontal mirror images are adopted for data augmentation. For scale jittering in the training phase, every image is resized with randomly ration in range $[0.5, 2.0]$. We also scale the image with different aspect ratio in range [4/5,5/4].

7.2. Comparisons with State-of-the-art

We quantitatively compare our proposed model with state-of-art models on these three datasets and our model is named **SegModel**. The segmentation results on the *test* set of Pascal VOC 2012 and Cityscapes is measured by mIoU (%). For ADE20K, the performance is measured by the average of mIoU(%) and pixel accuracy(%). We do not jointly train guidance CRF for ADE20K as there are too many (150) categories in this dataset. But guidance CRF is added in the inference stage.

In comparison, Deeplab-v2 [4] is trained on MS COCO *trainval* set and Pascal VOC 2012 augmented *train* set fine-tuning from Resnet-101. It ensembles three $8\times$ models both in the training and testing stage and adopts bilateral CRF as a post processing step. Our submission to Pascal VOC 2012 ensembles two $16\times$ models. Our model shows much higher performance than Deeplab-v2 on the *test* set of Pascal VOC 2012. For cityscapes, our two $16\times$ models are ensembled to reach the state-of-the-art. The 1024×2048 images can be easily feed into the network and segmented in a single run for one scale in our library. For detailed results on the two datasets please refer to Table 1. As shown in Table 2, our proposed model also has leading performance on ADE20K, which is the dataset of ILSVC2016 scene parsing track. We fine-tune the $16\times$ model from Resnet-152 during the competition but we find Resnet-101 gives a similar performance.

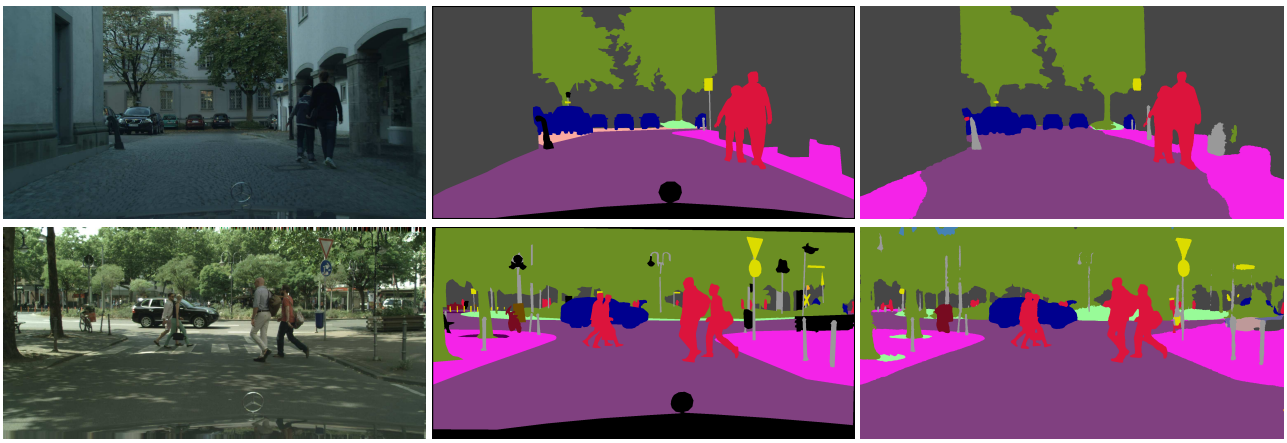
7.3. Ablative Studies

We conduct the evaluations of each components in our model on the Pascal VOC 2012 *val* set (1449 images), training on the augmented *train* set (10582 images). Each of our proposed parts is gradually added to model to do a ablation learning. We train for up to 36 epochs on Pascal VOC 2012 augmented *train* set and the training curves are shown in Fig. 7. To fairly show the effectiveness of each component, all these models are trained from the same base model for same training epoches. The whole training costs about 12 hours for **Guide** on two modern GPU cards. The final performance in Table 3 verifies the effectiveness of each component in our model and Table 4 displays the inference time of each model.

To classify the center pixel in a patch, feature context and label context provide a high-level understanding of a large region in the image and promote a smoothness between labels. **Context** brings a improvement of 6.8 percent on mean IoU at the cost of 14% more times to inference comparing to **Unary**. Adding the structured patch prediction part in context CRF further brings up the mean IoU while have little more time cost. **Patch** has also improved the boundary quality as shown in Fig. 6. After integrated with guidance

Table 3: Results on Pascal VOC 2012 *val* set. **Context**: Employing context CRF. **Patch**: Replace unary prediction in context CRF with structured patch prediction. **Guidance**: Add guidance CRF part to align the results. **Joint**: Jointly trained with MS-COCO. **MS**: multi-scale testing images.

	Unary	**Context**	**Patch**	**Guide**	**Joint**	**MS**
mIoU(%)	69.5	76.3	76.8	77.7	79.5	80.9



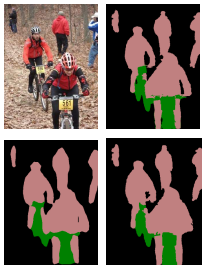
(a) Input (b) Truth (c) Prediction

Figure 5: Some visual results of Cityscapes *val* set. It costs about $0.5s$ for a 2048×1024 color image.

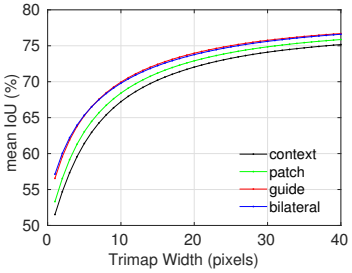
CRF, our full model **Guide** reaches a mean IoU of 77.7% on Pascal VOC 2012 *val* set. Bilateral CRF is widely adopted to delineate the object boundary in most previous works, guidance CRF has similar performance both visually and quantitatively as shown in Fig. 6 but it needs at much less time cost in the training stage and test stage. Finally, further exploiting MS-COCO and multi-scale testing, our model reaches 80.9% on the Pascal VOC 2012 *val* set.

Turning to implementation aspects, context CRF can be efficiently performed by box-filtering and hole algorithm. We use box filters with different kernel size on feature map and label map to average the context information in different size of regions. These averaged context information are put together via hole algorithm. For structured patch prediction, the convolutional layer is adopted to produce a long dimensional vector and re-arrange it in spatial dimensions. Both parts can be executed efficiently in CUDA.

Time complexity. All the code is optimized by CUDA and the time cost is measured on one GTX TITAN X. For a typical 300×500 color image, as it is shown in Table 4, it costs about $54.4ms$ in total to compute the segmentation score map on models fine-tuned from the Resnet-101, while the unary layers cost $43.7ms$. Our proposed context CRF and structured patch prediction costs little more time while bringing large performance gains. The bilateral-filtering based fully connected CRF is widely used for sharp object boundary in previous works [3, 4]. The bilateral CRF with a recently optimized implementation of fast bilateral filtering [17] takes about one second for 10 mean field iterations on CPU. As shown in Table 4 and Fig. 6, the guidance CRF layer costs only $4.4ms$ on one modern GPU card while having similar performance alongside the object boundary comparing to bilateral CRF. Some visual results are shown in Fig. 5.



(a)



(b)

Figure 6: (a)top-right: bilateral CRF on **Patch**. bottom-left: **Context**. bottom-right: **Guide**. (b) Pixel mean IoU around the object boundaries. The x-axis is the band width of the trimap.

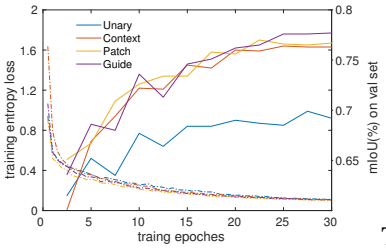


Figure 7: Training curves.

part	time(ms)
Unary	43.7
Context	49.8
Patch	50.0
Guide	54.4

Table 4: Inference time for a 500×300 color image.

8. Conclusion

In this paper, we have proposed a deep coarse-to-fine model with structured patch prediction, high order context and guided filtering for semantic image segmentation. Experiments on the Pascal VOC 2012, cityscapes and ADE20K show that our model achieves the state-of-the-art performance with appealing running speed.

ACKNOWLEDGEMENT

This work is supported by National Natural Science Foundation of China (NSFC) 61375022 and 61403005.

References

[1] S. Chandra and I. Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. *arXiv preprint arXiv:1603.08358*, 2016.

[2] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. *arXiv preprint arXiv:1511.03328*, 2015.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[4] L.-C. 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. *arXiv preprint arXiv:1606.00915*, 2016.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015.

[7] G. Ghiasi and C. Fowlkes. Laplacian reconstruction and refinement for semantic segmentation. *arXiv preprint arXiv:1605.02264*, 2016.

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[10] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 991–998. IEEE, 2011.

[11] K. He and J. Sun. Fast guided filter. *arXiv preprint arXiv:1505.00996*, 2015.

[12] K. He, J. Sun, and X. Tang. Guided image filtering. In *Computer Vision–ECCV 2010*, pages 1–14. Springer, 2010.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] 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 arXiv:1408.5093*, 2014.

[16] D. Koller and N. Friedman. Probabilistic graphical models, massachusetts, 2009.

[17] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *arXiv preprint arXiv:1210.5644*, 2012.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015.

[20] G. Lin, C. Shen, I. Reid, and A. van den Hengel. Deeply learning the messages in message passing inference. In *Advances in Neural Information Processing Systems*, pages 361–369, 2015.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[22] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[23] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1385, 2015.

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[25] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015.

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[27] S. Ross, D. Munoz, M. Hebert, and J. A. Bagnell. Learning message-passing inference machines for structured prediction. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2737–2744. IEEE, 2011.

[28] C. Russell, P. Kohli, P. H. Torr, et al. Associative hierarchical crfs for object class image segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 739–746. IEEE, 2009.

[29] Z. Tu. Auto-context and its application to high-level vision tasks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[30] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 2015.

[31] Z. Wu, C. Shen, and A. v. d. Hengel. Bridging category-level and instance-level semantic image segmentation. *arXiv preprint arXiv:1605.06885*, 2016.

[32] Z. Wu, C. Shen, and A. v. d. Hengel. High-performance semantic segmentation using very deep fully convolutional networks. *arXiv preprint arXiv:1604.04339*, 2016.

[33] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[34] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016.

[35] 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. *arXiv preprint arXiv:1502.03240*, 2015.

[36] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016.