All-Higher-Stages-In Adaptive Context Aggregation for Semantic Edge Detection

Qihan Bo, Wei Ma, Yu-Kun Lai, and Hongbin Zha

***Abstract*—Convolutional Neural Networks (CNNs) can reveal local variation details and multi-scale spatial context in images via low-to-high stages of feature expression; effective fusion of these raw features is key to Semantic Edge Detection (SED). The methods available in the field generally fuse features across stages in a position-aligned mode, which cannot satisfy the requirements of diverse semantic context in categorizing different pixels. In this paper, we propose a deep framework for SED, the core of which is a new multi-stage feature fusion structure, called All-HiS-In ACA (All-Higher-Stages-In Adaptive Context Aggregation). All-HiS-In ACA can adaptively select semantic context from all higher-stages for detailed features via a cross-stage self-attention paradigm, and thus can obtain fused features with high-resolution details for edge localization and rich semantics for edge categorization. In addition, we develop a non-parametric Inter-layer Complementary Enhancement (ICE) module to supplement clues at each stage with their counterparts in adjacent stages. The ICE-enhanced multi-stage features are then fed into the All-HiS-In ACA module. We also construct an Object-level Semantic Integration (OSI) module to further refine the fused features by enforcing the consistency of the features within the same object. Extensive experiments demonstrate the superior performance of the proposed method over state-of-the-art works.**

***Index Terms*—semantic edge detection, multi-stage feature fusion, adaptive context aggregation, complementary feature enhancement, object-level semantic integration**

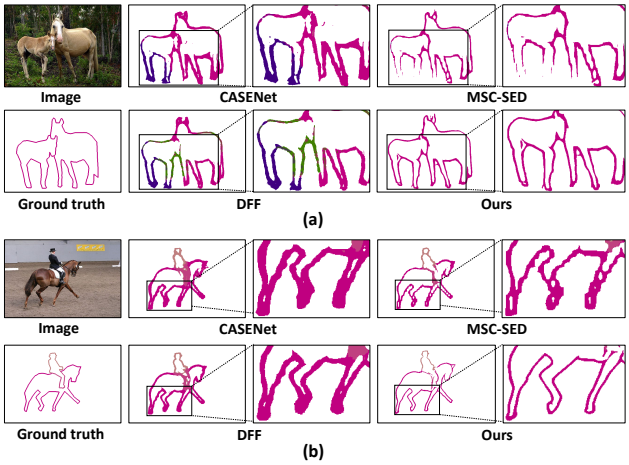


Fig. 1. Visual comparison of results obtained by CASENet [9], DFF [10], MSC-SED [11] and our method (best viewed in color). We choose CASENet, DFF and MSC-SED because they are representatives of state-of-the-art methods with source codes provided. The input images are from SBD dataset [15]. In (a), many samples at the small horse's legs are misclassified as non-horse by CASENet and DFF, and misclassified as non-edge by MSC-SED. Our method obtains much better semantic edges due to the adaptive context aggregation. In (b), the edges predicted by CASENet, DFF, and MSC-SED are coarse and topologically wrong. The proposed method obtains much thinner edges because the detail features are well preserved during feature fusions.

I. Introduction

SEMANTIC Edge Detection (SED) in images aims at jointly locating object boundaries and recognizing their semantic categories. SED benefits a wide variety of research topics and applications, including semantic visual SLAM [1], image-based localization [2], and 3D geometry estimation [3]. SED can be viewed as a dual-task of semantic segmentation. Differently, semantic segmentation decomposes an image into semantic regions by categorizing each pixel according to its spatial context [4]. The obtained semantic regions can be converted to semantic edges by keeping only boundary points and their semantical categories [5]. However, semantic segmentation is error-prone in categorizing pixels near boundaries, due to the complex context compositions around these pixels [6]–[8]. In contrast, focusing on locating and categorizing contour samples, SED has potential to obtain more accurate boundaries.

Convolutional Neural Networks (CNNs) have been widely used in various image processing and understanding tasks. CNN can extract multi-stage rich features from a single image, in which the lower-stage features reflect local variations and the higher-stage features contain more semantic clues. Effective fusion of the multi-stage features is key to SED. Existing CNN-based SED methods generally solve this issue via cross-stage concatenation or stage-by-stage gradual fusion in a position-aligned mode. For example, CASENet [9] fuses the highest stage feature maps with those of the lowest three stages via position-aligned shared concatenation. All the feature maps are rescaled to have the same size as the input image before the fusion. Subsequently, Hu et al. [10] extended CASENet to Dynamic Feature Fusion (DFF) for location-adaptive weights. More recently, considering that the lower-stage features are noisy, Ma et al. [11] proposed a Multi-scale Spatial Context based network for SED (MSC-SED), which gradually selects details and integrates them into higher-stage features at the same position.

Manuscript received ** **, 2021; revised ** **, **. This work was supported by the National Natural Science Foundation of China (Nos. 62176010, 61771026, 61632003). *(Corresponding author: Wei Ma.)*

Qihan Bo and Wei Ma are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: boqihan@emails.bjut.edu.cn; mawei@bjut.edu.cn).

Yu-Kun Lai is with the Cardiff University, Cardiff, UK (e-mail: laiy4@cardiff.ac.uk).

Hongbin Zha is with the Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: zha@cis.pku.edu.cn).

Although great progress has been made in multi-stage feature fusion for SED, existing methods have two limitations as follows.

1) Firstly, the above position-aligned fusion strategies cannot satisfy the diverse requirements of context clues at different edge points. For example, for edge samples located at the head of the big horse in Fig. 1(a), context clues aggregated by CASENet, DFF and MSC-SED are adequate for categorizing these samples. However, these methods cannot aggregate context required for correct categorization of the hard samples located at the little animal's legs. Correctly recognizing these hard samples probably requires context from the easy-to-recognize big horse, considering that a little animal next to a big horse is probably a horse as well. As shown in Fig. 1(a), CASENet, DFF and MSC-SED wrongly classify most of the edge samples from the little horse as non-horse or non-edge categories.
2) Secondly, it is hard for existing methods [9]–[11] to generate fused features with clear details. Specifically, the cross-stage concatenation [9], [10] upsamples multi-stage features to the same size and then linearly combines them together. The mixture of multi-scale features results in fused features with blurred details. The gradual fusion in [11] enhances abstract features in higher-stages with details. The fused feature maps inherit the low-resolution and abstract properties of the higher-stage features. Edges predicted based on such coarse features are generally thick and inaccurate, as it can be seen from the results of CASENet, DFF, and MSC-SED in Fig. 1(b). Non-Maximum Suppression (NMS) [12] has been widely applied as a post-processing strategy to thin the predicted edges [5], [13], [14]. However, NMS cannot eliminate the topological errors caused by the coarse features. For instance, the two sides of the horse legs obtained by CASENet, DFF and MSC-SED in Fig. 1(b) are wrongly adhered to each other, which cannot be corrected via NMS.

In this paper, we propose a new deep framework for SED. Instead of the position-aligned fusion, the proposed framework performs the multi-stage feature fusion in an adaptive way. Specifically, the proposed framework can adaptively aggregate required semantics for lower stage features from any position of any higher stage via cross-stage self-attention. We name this fusion as All-Higher-Stages-In (All-HiS-In) adaptive context aggregation. Due to the All-HiS-In adaptive context aggregation, our model can obtain fused features with clearer details and richer semantics. SED based on such features can correctly recognize hard samples located at the little horse contours in Fig. 1(a) and obtain thin boundaries directly, as shown in Fig. 1(b).

In addition, existing SED methods generally use ResNet as backbones for multi-stage feature extraction [9]–[11]. As features at adjacent stages are complementary in details or semantics, we propose an Inter-layer Complementary Enhancement (ICE) module, with no learnable parameters, to mutually supplement the feature representations of adjacent stages in groups. The ICE-enhanced features are then sent to All-HiS-In ACA. Besides, we propose an Object-level Semantic Integration (OSI) module to enrich the higher-stage features with object-level context. OSI helps regularize the representation within objects for consistent categorization of their edge points and suppression of inner edge points.

Our main contributions can be summarized as follows:

1) We propose a new All-HiS-In deep architecture for semantic edge detection. The core of the architecture is an All-HiS-In Adaptive Context Aggregation (All-HiS-In ACA) fusion method. Compared to existing fusion methods for SED, All-HiS-In ACA aggregates required semantic context for samples adaptively while preserving precise and delicate edges after fusion, and thus can obtain thinner and more accurate semantic edges.
2) We propose an ICE module, which can enhance features at each stage by referring to adjacent stages. ICE brings no burden to the training process since it contains no parameters to learn, while enriching the features and their inter-stage relevance.
3) We propose an OSI module to regularize the representation consistency of samples belonging to the same object. OSI further refines the fused features for edge localization and categorization.
4) All-HiS-In ACA, ICE and OSI are proved effective via rich ablation studies. And the proposed network achieves new state-of-the-art performance on both the SBD and Cityscapes datasets.

II. Related Work

A. Semantic Edge Detection

Early methods generally decompose the semantic edge detection task into separate subtasks and complete them one by one. For example, Hariharan et al. [15] proposed inverse detectors to detect semantic contours by combining information from a bottom-up contour detector and generic object detectors. Bertasius et al. [16] proposed a High-for-Low (HFL) scheme which produces category-agnostic edge points and then associates the points with semantic categories by referring to object-level features extracted by CNN. Maninis et al. [17] proposed Convolutional Oriented Boundaries (COB) to detect oriented contours and then assigned the contour points with class labels according to the masks obtained via semantic segmentation.

Recently, SED is generally treated as a multi-label classification task [9]–[11], so that edge detection and categorization are solved jointly with interaction in a unified framework. The core of these frameworks is the fusion of multi-stage CNN features. To fuse the multi-stage information for SED, existing methods perform cross-stage concatenation [9], [10] (as illustrated in Fig. 2(a)) or stage-by-stage gradual fusion [11] (as illustrated in Fig. 2(b)).

The representative work using the cross-stage concatenation-based fusion strategy is CASENet, proposed by Yu et al. [9] in 2017. Specifically, it enhances each channel of the side activations at the highest stage with low-level detail features via concatenation. Subsequently, Yu et al. [13]

argued that label noises caused by inevitable misalignment of edges during annotation can degrade edge learning quality and proposed a Simultaneous Edge Alignment and Learning (SEAL) framework based on CASENet. Liu et al. [18] adopted Diverse Deep Supervision (DDS) on all side activations of CASENet to boost the hierarchical feature representation. Hu et al. [10] improved CASENet with location-adaptive weights via a Dynamic Feature Fusion (DFF) strategy. Zhen et al. [5] developed an Pyramid Context Module (PCM) to aggregate global semantical context for lower-stage features via global pooling of higher stage features and then concatenating them with lower-stage features. They also utilized clues from semantic segmentation to suppress non-semantic edges for boundary detection. The above cross-stage concatenation methods perform fusion regardless of the noises in the lower level features. Ma et al. [11] proposed a bottom-up gradual fusion framework MSC-SED, which can suppress the noises in the lower-level feature according to semantic context during fusion and achieves higher performance.

Both the concatenation and gradual fusion methods perform position-aligned context aggregation across certain stages for all samples, which can not satisfy the adaptive semantic requirements of different edge points. In addition, the cross-stage concatenation and gradual fusion blur the detail features as we analyzed in Section I. Their fused maps for categorization are coarse at edge points and therefore the detected boundaries are thick. Post-processing algorithms, e.g., NMS [12], [19], can thin the boundaries but cannot correct the topological errors (as illustrated in Fig. 1) caused by the coarse fused maps. The proposed All-HiS-In fusion method (as illustrated in Fig. 2(c)) solves the above issues as we analyzed in Section I.

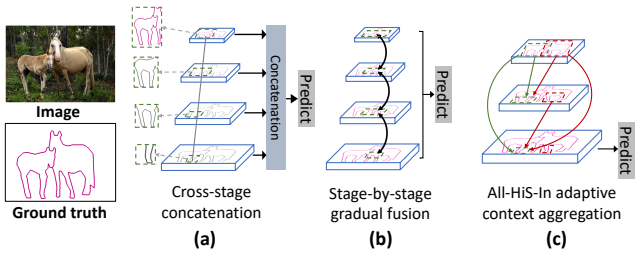


Fig. 2. Three typical multi-stage feature fusion strategies, cross-stage concatenation (a), stage-by-stage gradual fusion (b), and the proposed All-HiS-In fusion (c). The green dashed boxes denote receptive fields. The gray line in (a) indicates position-aligned cross-stage fusion. The black curves with arrow heads in (b) represent stage-by-stage gradual and position-aligned fusion in a bottom-up/top-down manner. The green and red curves with arrow heads in (c) indicate the adaptive context aggregation in stages and positions for the lower stages features.

B. Edge-Enhanced Semantic Segmentation

Semantic edge detection and semantic segmentation can be seen as dual tasks. Semantic segmentation generates semantic regions by describing and categorizing each pixel based on its spatial context. In contrast, semantic edge detection focuses on edge points and solves two subtasks, i.e., edge localization and categorization, at the same time. Semantic segmentation results can be conveniently converted to semantic edges. Therefore, we here briefly review works on semantic segmentation.

Various methods have been developed for semantic segmentation, e.g., U-Net [20], DeepLab series [21]–[23], and FastFCN [24], based on Fully Convolutional Networks (FCN) [25]. U-Net [20] adopts an encoder-decoder structure with skip-connections for richer types of features. DeepLab [21] employs dilated convolution operations to enlarge receptive fields for non-local semantical context. SFANet [4] alleviates the misalignment gap and seeks the balance between accuracy and inference speed. Tian et al. [26] dealt with semantic segmentation in the framework of unsupervised domain adaptation and proposed partial domain adaptation to avoid the negative transfer problem. Zhang et al. [27] extended proposal-based object segmentation beyond detected bounding boxes. DeepLabv3 [22] adopts Atrous Spatial Pyramid Pooling (ASPP) module to capture multi-scale contextual information and obtain image-level features that encode global context. The above methods are not effective in categorizing samples near boundaries, since the local variations of these samples are blurred during semantical context aggregation and the context of these samples is confusing for categorization.

Thereafter, many works try to improve the performance of semantic segmentation in expressing boundary samples. For example, GSCNN [6] introduces an extra gated shape CNN stream to extract boundary-related information, which is then merged with features from a regular semantic segmentation branch for semantic categorization. BNF [28] utilizes a global energy function to model the pairwise pixel affinities based on the boundary prediction. In [29], a Boundary-aware Feature Propagation module (BFP) is proposed to propagate local features within object regions, which are enclosed by boundaries learned using an extra branch of model. Chen et al. [23] introduced a decoder into DeepLabv3 [22] to refine segmentation results especially near object boundaries by gradually recovering spatial information. Zhen et al. [5] improved boundaries of semantic masks via boundary consistency constraint. Ji et al. [30] developed a cascaded CRFs and introduced it into the decoder of semantic segmentation to supplement information at boundaries. All the above edge-enhanced semantic segmentation methods try to compute boundary clues via extra computations and integrate them into the intermediate representation of semantic segmentation. The idea and consequence are similar to those of CASENet series. In this paper, we propose to aggregate context for boundary samples adaptively. We believe that the proposed method will also benefit the research on edge-enhanced semantic segmentation.

C. Contour Edge Detection

If the categorization subtask in SED is ignored, SED will degrade to contour edge detection. Recently, many CNN-based methods have been developed for contour edge detection. For example, Bertasius et al. [31] extracted candidate contour points via Canny edge detector and then scored each candidate as a contour point based on the connected multi-scale CNN features around the point. To suppress noise edges, DeepContour [32] partitioned contour data into subclasses as supervision to regularize CNN features. Xie et al. [33] adopted

an end-to-end fully convolutional neural network and trained it via deep supervision on side responses to solve the ambiguity in edge and object boundary detection. RCF [34] concatenated all layers of CNN features for edge detection in an image-to-image fashion. Wang et al. [35] proposed a decoder structure to obtain crisp object boundaries. LPCB [36] discussed the reason of blurry edges and proposed a method to directly predict crisp boundaries without post-processing. Kelm et al. [37] adopted a top-down paradigm for multi-stage feature fusion in their RefineContourNet. He et al. [14] developed a Bi-Directional Cascade Network (BDCN) and supervised it with labeled edges at each specific scale. Soria et al. [38] presented a Dense Extreme Inception Network for Edge Detection (DexiNed), in which two different strategies are tested for integration of multi-scale outputs, namely concatenation-then-fusion (termed as DexiNed-f) and averaging (termed as DexiNed-a). Deng et al. [39] proposed a dense connection structure to effectively utilize semantics and a novel loss for contour-structure similarity. PiDiNet [40] adopted a lightweight approach for edge detection by integrating the traditional edge detection operators into convolutional operations.

Compared to contour edge detection, semantic edge detection is more challenging and needs extra category-relevant semantics. Note that since contour edge detection is a sub-problem of SED, SED methods can be used for contour edge detection. We experimentally demonstrate that SED methods, including the proposed one, outperform existing contour edge detection methods in this task by large margins, due to the assistance of category-relevant semantics.

III. Approach

In this section, we describe the proposed network in detail. Firstly, we overview the overall architecture of the proposed network. Then we introduce the key components, including the Inter-layer Complementary Enhancement (ICE) module, All-HiS-In adaptive context aggregation for multi-stage fusion, and the Object-level Semantic Integration (OSI) module. Finally, we describe the total loss used to train our network.

A. Overview of the Proposed Architecture

In this paper, we adopt ResNet-101 with the dilated strategy [21], which has been widely acknowledged by existing SED methods, as the backbone. Following [9], we remove the original average pooling and fully connected layer, and change the stride of the first and fifth convolution blocks in ResNet-101 from 2 to 1 for better preservation of low-level edge information. Note that the size of the final map has a spatial resolution of 1/8 of the input image [10], [11]. For quick reference, we list the involved notations in Table I. As shown in Fig. 3, the last four stages of feature maps with different scales from ResNet-101 are fed into a 1×1 convolution layer to reduce channel number to 64. The outputs, denoted as R_1, R_2, R_3, R_4, are then divided into two groups, i.e., $\{R_1, R_2\}$ and $\{R_3, R_4\}$. Each group is sent to an ICE module, to enhance each stage's features according to the complementary information in the other stage's features in the same group.

TABLE I
Notation table

Notation	Description
$R_i, i \in \{1,2,3,4\}$	Four stages of features from the backbone.
$F_i, i \in \{1,2,3,4\}$	Four stages of features enhanced by the ICE module.
$T_i, i \in \{1,2,3,4\}$	Four layers of intermediate features in the All-HiS-In ACA module.
G	Global semantic features obtained by globally pooling the backbone's top-stage features.
$M_1, ..., M_K$	Intermediate semantic segmentation prediction. K is the number of categories.
T_{sum}	All-HiS-In ACA fused features.
O_{sum}	OSI refined features
$Trans()$	Operation of mapping relationship evaluation in the All-HiS-In ACA module, defined by equation 5.
$V(), Q(), K()$	1×1 convolution operations used for different elements in equation 5.
$W_{j,i}$	Similarity matrix of F_j and F_i
$Conv, Concat, Upsam$	Convolution, concatenation and upsampling operations

After that, we obtain four stages of richer features, denoted as F_1, F_2, F_3, F_4. The four sets of enhanced features and the global semantics G from the top layer of the backbone will be jointly delivered to the All-HiS-In ACA structure, where appropriate semantics are adaptively selected from all the higher-stages and then embedded in the lowest-stage features to obtain T_{sum} as the fused features for SED.

Note that, beside the SED supervision, we perform intermediate supervision after the ICE modules. Specifically, two prediction branches are built based on the two ICE modules for semantic segmentation and edge detection. Supervision on these two branches of predictions will drive the features at stages 3 and 4 to contain more semantics for categorization and features at stages 1 and 2 to have more edge details.

As intermediate results, the prediction masks, denoted as $M_1, ..., M_K$, in the semantic branch contains wealthy object-level semantic priors. We use these intermediate predictions in the OSI module to further unify the representations of contour samples belonging to the same object, to boost the completeness and semantic-correctness of contours in the final SED results.

B. Inter-Layer Complementary Enhancement (ICE)

The idea of inter-layer complementary enhancement is inspired by the following observations. As revealed in [18] based on CASENet and agreed by most of the SED methods [9], [11], the five stages of features in ResNet-101 have their specialties. Features at the lowest three stages contain rich edge-related details, but lack semantic clues to distinguish contours from numerous edges and identify contours of a specific semantical class. In other words, the lower layers of the network tend to focus more on local gradient variations, but cannot provide context even for contour-or-not binary categorization. In the feature maps of the fourth and fifth stages, the contours of objects are highlighted and endowed

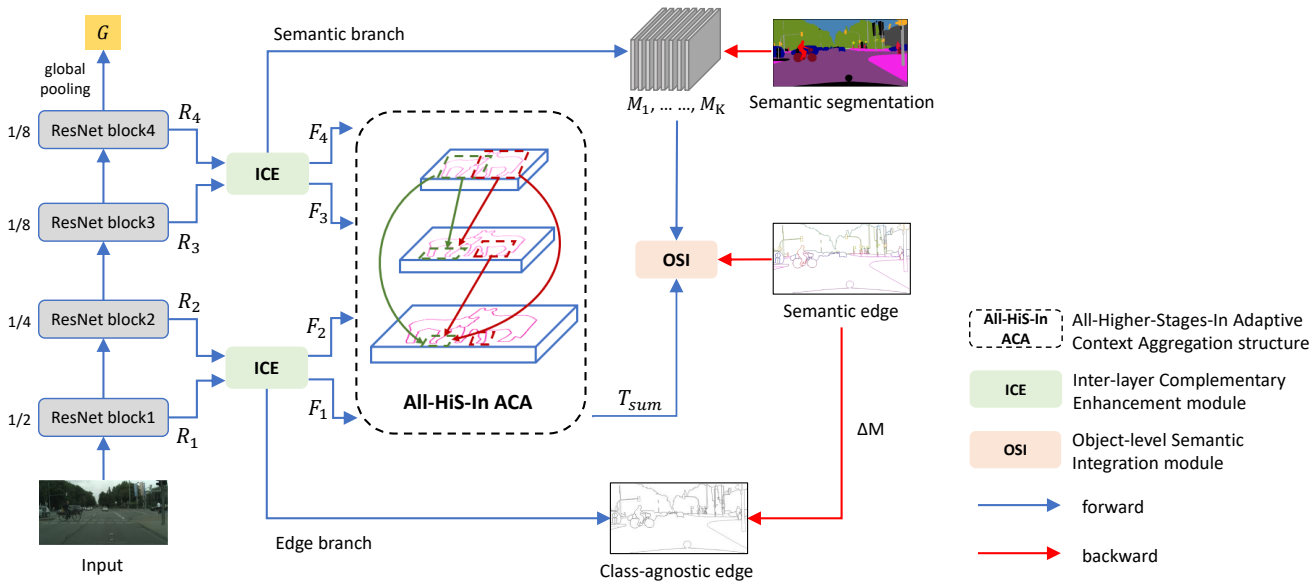


Fig. 3. Overview of the proposed network. The Inter-layer Complementary Enhancement (ICE) module is used to enhance feature representations by referring to an adjacent stage with similar properties. The enhanced features are then sent to the All-Higher-Stages-In Adaptive Context Aggregation (All-HiS-In ACA) module for multi-stage feature fusion. The fused features are refined in the Object-level Semantic Integration (OSI) module. There are three parts of supervision in all: edge detection of the edge branch, semantic segmentation of the semantic branch and the final semantic edge detection based on the fused features. ΔM is the operation of obtaining object contours along the semantic dimension, imposed on the ground truth of semantic edges.

with rich semantic clues for categorization; the internal edges and background textures can be suppressed as well. However, the contours are much blurred in positions due to the low resolution of the feature maps at these stages. Therefore, the features at the fourth and fifth stages of ResNet-101 are more suitable for semantic categorization than edge localization, as opposed to the other three stages of the network.

According to the above facts, we group the second to the fifth stages' features obtained by ResNet-101 into two groups according to their different specialties. Note that, the lowest stage of features contains too much edge noise, and therefore we will not use this stage in the following steps. As shown in Fig. 3, we design a two-branch structure, including edge branch and semantic branch. We enrich the two groups of the features according to their specialties by using two strategies. At first, although the two stages in each group are similar in specialties, they have clues complementary to each other due to stage-by-stage CNN encoding. Given features of a group of two stages, the ICE module compensates for the ignored details in the higher stage features due to down-sampling and dilated convolution and compensates for the restricted receptive field in the lower stage. These compensation operations enrich the multi-stage features and strengthen their inter-layer relevance, and thus are helpful to the following adaptive context aggregation based on cross-stage self-attention.

Concretely, given the side outputs of a group of stages, denoted as (R_1, R_2) or (R_3, R_4), we obtain an enhanced group of features, denoted as (F_1, F_2) or (F_3, F_4), by aggregating responses of the other layer in the same group via ICE. Taking the second group as an example, as shown in Fig. 4, R_3 and R_4 pass through the activation function σ ($Sigmoid$ function are used in experiments) , and become $\sigma(R_3)$ and $\sigma(R_4)$, respectively. We also compute $1-\sigma(R_3)$ and $1-\sigma(R_4)$,

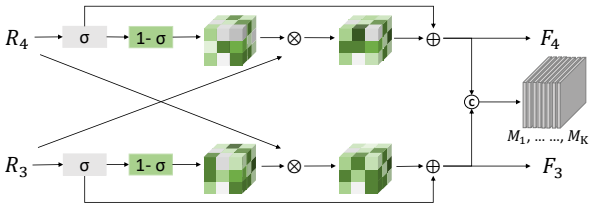


Fig. 4. The Inter-layer Complementary Enhancement (ICE) module located in the semantic branch. σ represents a $Sigmoid$ activation function. $\otimes$ and $\oplus$ represent element-wise multiplication and summation, respectively. © represents the concatenation operation.

to represent the potentially missing responses in $\sigma(R_3)$ and $\sigma(R_4)$, respectively. We use clues from the other stage of features for compensation of the potentially missing clues. Mathematically,

$$F_3 = \sigma(R_3) + R_4 \cdot (1 - \sigma(R_3)) \tag{1}$$

$$F_4 = \sigma(R_4) + R_3 \cdot (1 - \sigma(R_4)) \tag{2}$$

Note that in some positions, there might be no responses even when taking into account complementary information from both stages.The ICE module contains no learned parameters and therefore brings no burden to the network training.

The second strategy to enrich the two groups of features is to supervise their training with suitable ground truth data. To achieve this, besides the enhanced features, we output a set of masks $M_1, ..., M_K$ from ICE:

$$M_1, ..., M_K = Conv(Concat(F_3, F_4)) \tag{3}$$

They are object-level masks and K is the number of categories. In the edge branch, masks $M_1, ..., M_K$ are category-agnostic

binary contours and $K = 1$. We use the ground truth of semantic segmentation and that of contour detection as supervision of the semantic branch and edge branch, respectively.

C. All-Higher-Stages-In Adaptive Context Aggregation (All-HiS-In ACA)

We develop the All-HiS-In ACA structure to achieve adaptive context aggregation in positions and stages. Particularly, we enhance features at each stage with context adaptively chosen from all its higher stages via cross-stage self-attention. We then upsample the context-enhanced features at each stage to the resolution of the lowest stage and fuse them together. The fused features contain rich and adaptively selected context for categorization while keeping the high-resolution details in the lowest stage.

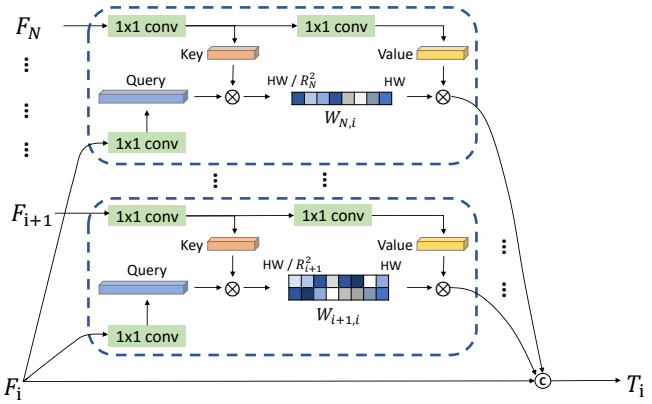


Fig. 5. The All-Higher-Stages-In Adaptive Context Aggregation (All-HiS-In ACA) structure for the i-th layer. All stages higher than the ith stage, including the global feature layer G, would be treated as context sources of the i-th layer. After aggregating context from every higher stage, all the context-enhanced results are concatenated as the final output T_i.

We depict the computation in aggregating context for the ith stage of feature maps, denoted as F_i, in Fig. 5. We use its all higher stages of feature maps $\{F_j, \mathrm{G}\}$ as the source of semantic context clues. Here, $j \in [i+1, N]$. $N = 4$, denoting the total number of stages. G represents the global semantic features obtained by globally pooling the top stage features of the backbone network. The adaptive context aggregation is the concatenation of all the context found from all the higher stages of features and F_i itself, which can be mathematically represented as:

$$T_i = Concat(\sum_{j=i+1}^{N} Trans(F_i, F_j), Trans(F_i, G), F_i) \quad (4)$$

where T_i has the same scale as F_i. We perform a $Trans$ operation on F_j and F_i to fuse cross layer features F_i and F_j:

$$Trans(F_i, F_j) = V(F_j) \cdot Sim(Q(F_i), K(F_j)) \quad (5)$$

The $Trans$ operation is inspired by the Transformer architecture [41] and evaluates the mapping relationship between the two stages of features, i.e., F_i and F_j, with different scales, as depicted in the blue dashed box of Fig. 5. In each $Trans$ operation, we transform F_i and F_j to Q_i and K_j via 1×1 convolutions as query and key, respectively. We then compute the similarity between elements of Q_i and K_j by using dot product and obtain $W_{j,i}$. The scale of $W_{j,i}$ is jointly determined by the sizes of F_i and F_j. As the depth of the network increases, the feature map F_j contains more semantics and has a larger perceptive field (with a lower spatial resolution), so the size of $W_{j,i}$ decreases as j increases. Finally, the similarity map $W_{j,i}$ is multiplied with $V(F_j)$, which is obtained by applying two 1×1 convolutions consecutively on F_j, to generate $Trans(F_i, F_j)$.

After enhancing all stages of feature maps via Equation 4, we up-sample the enhanced features by bilinear interpolation and sum them to obtain the final fused output T_{sum}:

$$T_{sum} = \sum_{i=1}^{N} Upsam(T_i) \quad (6)$$

T_{sum} has the same resolution as the first stage of feature maps, i.e., half of the original image size.

Due to the proposed All-HiS-In adaptive context aggregation structure, the category-agnostic edges can be directly attached with appropriate semantic cues, making the contours stand out from the general edges. In addition, the operation of $Trans$ can aggregate context from inter- and intra-object areas from all the higher-stages, which ensures sufficient context for semantic categorization.

D. Object-Level Semantic Integration (OSI)

After the ICE module and All-HiS-In ACA structure, we obtain features T_{sum} and the semantic masks prediction $M_1, ..., M_K$, where K is the number of categories. The fused features T_{sum} can be used for SED. Inspired by the object-level context for semantic segmentation [42], to further restrict the completeness of object contours in the final SED results, we feed the predicted mask $M_1, ..., M_K$ as semantic priors into the OSI module to refine T_{sum}. As illustrated in Fig. 6, there are K sets of features in the object-level representations, where different colors represent features of different types of objects. By regularizing the features in T_{sum} with object-level context, we can obtain more consistent representation in edge samples belonging to the same mask. In addition, the features within each mask are regularized as well, which benefits suppressing inner edges in the SED output. The pixel-object relation obtained by multiplication of the two represents the affiliation probability of a pixel belonging to a corresponding object category, and has the same resolution as T_{sum}. Finally, the regularized features are concatenated with the original feature T_{sum} as the output of the OSI module, denoted as O_{sum}. O_{sum} is then used for SED.

E. Loss Function

The total loss used for supervising the network training consists of three parts, i.e., semantic edge detection loss L_{sed}, semantic segmentation loss L_{seg}, and object contour detection loss L_{con}:

$$L_{total} = L_{sed} + \lambda_1 \cdot L_{seg} + \lambda_2 \cdot L_{con} \quad (7)$$

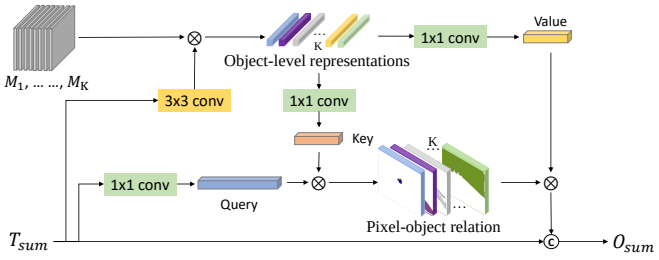


Fig. 6. The structure of Object-level Semantic Integration (OSI) module. This module utilizes the intermediate predictions of semantic segmentation as semantic priors to refine the feature representations of the contour samples belonging to the same object in T_{sum}.

The ground truth data of semantic segmentation is provided in the datasets, based on which we derive the ground truth data of SED, as done in [5], [10], [11], [13], [22]. The ground truth data of object contour detection is obtained by ignoring all the category properties in the SED ground truth. The supervision of contours and category masks are applied to the edge branch and semantic branch in section III-B, respectively. The predicted activations of the semantic branch are further used as the prior information for the subsequent modules. The semantic edges are the expected final predictions obtained based on the fused features O_{sum}.

We choose the auto-weighted loss proposed in [11] as L_{sed}. L_{seg} is the cross-entropy loss widely adopted in existing semantic segmentation works [6], [20], [22]. L_{con} takes the form of the loss function adopted in the mainstream edge detection works [14], [43]. λ_1 and λ_2 are weights of the segmentation loss and the contour detection loss. In our experiments, we empirically set $\lambda_1 = 0.025$ and $\lambda_2 = 0.015$.

IV. Experiments

A. Datasets

We evaluate our proposed method on two datasets popular in the research of semantic edge detection: SBD [15] and Cityscapes [44].

1) Semantic Boundary Dataset (SBD): It contains 11355 images from the train and val sets of PASCAL VOC2011, with 8498 images as the training set, and the remaining 2857 images as the test set. The dataset contains both category-level and instance-level segmentations and boundaries, with 20 object categories in all. Following existing methods [9], we use the training set to train our network and the test set for evaluation.

2) Cityscapes: It is a large-scale semantic segmentation dataset containing 5000 finely annotated images, which are divided into 2975 images for training, 500 images for validation, and 1525 images for testing. Each image is of 2048×1024 size and has high quality pixel-level labels of 19 semantic classes. However, the ground truth of the test set has not been published. Following [5], [9]–[11], [13], we use the training set for training and validation set for evaluation.

B. Implementation Details

1) Data Augmentations: We use random horizontal flip and random cropping on both Cityscapes and SBD datasets. For SBD, we augment the training data by resizing each training image with scaling factors of $\{0.5, 0.75, 1, 1.25, 1.5\}$ as [9]–[11]. For Cityscapes, each training image is resized once with a scaling factor randomly sampled from $\{0.5, 0.75, 1, 1.25, 1.5\}$.

2) Training Strategies: The proposed network is implemented with PyTorch on an NVIDIA GTX2080Ti (11GB) GPU. Following [9], [11], we adopt ResNet-101 pretrained on MS COCO [45] as backbone. We use stochastic gradient descent (SGD) to optimize the parameters of our network without using the gradient accumulation in [11]. The weight decay, momentum and batch size are set to 5e-4, 0.9 and 1, respectively. We set the base learning rate, iteration number and crop size to 3e-8/1e-8, 320k/900k and 352×352/512×512, respectively, for SBD/Cityscapes. We optimize the network by using the learning rate policy of 'poly', where the base learning rate is multiplied by $(1-\frac{iter_{cur}}{iter_{max}})^{power}$ with $power$= 0.9. Here, $iter_{cur}$ and $iter_{max}$ represent the current iteration number and the total iteration number, respectively.

C. Evaluation Metric

For performance evaluation, we follow the evaluation protocol proposed in [13], which is considered stricter than the one used in [9]. The Maximum F-measure (MF) at Optimal Dataset Scale (ODS) for each class is reported to evaluate semantic edges. An essential parameter in the evaluation is the matching distance tolerance, which is defined as the maximum slack allowed for boundary predictions to be considered as correct matches to the ground truth. We follow [11] and set the matching distance tolerance as 0.02 for SBD and 0.0035 for Cityscapes. We also test the matching distance tolerance of 0.00375 for Cityscapes for comparison with more related methods. Following [10], the ground truth maps are down-sampled to half of the original dimensions for Cityscapes, and contain instance-sensitive edges for both datasets.

D. Ablation Experiments

We perform ablation experiments on SBD to verify the effectiveness of the proposed modules, including All-HiS-In ACA, ICE and OSI. The verification is conducted in two modes, i.e., using the modules individually or adding them one-by-one, based on the backbone of ResNet-101. All the results are listed in Table II. The baseline method without adding any component utilizes only the top stage of features from ResNet-101 for semantic edge detection, as the Basic method in [9]. Note that we do not validate OSI alone, since it requires the semantic masks generated by ICE. ICE is validated in two ways: as a whole and a composition of the two parts, i.e., the ICE structure and the two branches of supervision.

1) Ablation Study on All-HiS-In ACA Fusion: We validate All-HiS-In ACA fusion on SBD dataset by adding this structure to the backbone of ResNet-101 and record the performance improvement that All-HiS-In ACA brings in. In this study, the inputs of All-HiS-In ACA are feature sets R_1, R_2, R_3, R_4, obtained from the last four stages of the backbone. The output T_{sum} is used as the fused features for

TABLE II
ABLATION STUDY ON THE CORE COMPONENTS OF THE PROPOSED MODEL BASED ON SBD DATASET. THE MEAN VALUE OF THE MF SCORES (%) OVER ALL CATEGORIES IS PRESENTED. Δ MF REPRESENTS THE INCREASE RELATIVE TO THE BACKBONE.

Network	All-HiS-In ACA	ICE		OSI	Mean MF	Δ MF
		Complementary Enhancement	Edge Loss & Segmentation Loss			
Backbone (ResNet-101)					73.6	
Our	√				76.1	+2.5
		√			75.6	+2.0
			√		75.3	+1.7
		√	√		75.9	+2.3
	√	√			76.5	+2.9
	√	√	√		76.9	+3.3
	√	√	√	√	77.3	+3.7

TABLE III
COMPARISON OF DIFFERENT MULTI-STAGE FEATURE FUSION STRUCTURES ON THE SBD DATASET. * REPRESENTS OUR IMPLEMENTATION. THE PARAMETERS, MEMORY, AND FLOPS ARE ESTIMATED WITH AN INPUT IMAGE OF 352×352 FROM THE SBD DATASET.

Backbone	Method	Mean MF	Parameters(M)	Memory(MB)	FLOPs(G)	Training time(H)
ResNet-101	Basic [9]	73.6	42.541	1716.84	96.99	8.95
	Shared concatenation [9]	74.4	42.542	1726.91	97.02	9.78
	Bottom-up fusion with LAM [11]	74.0	45.128	1835.94	103.66	11.9
	Dynamic feature fusion [10]	74.3	42.725	1831.48	99.71	11.03
	PCM * [5] (Our impl.)	74.5	43.161	1728.50	99.10	9.55
	Our All-HiS-In ACA	**76.1**	42.970	1778.14	98.79	11.56

semantic edge detection. As it can be observed from Table II, compared to SED based on the backbone, the proposed All-HiS-In ACA structure brings 2.5% increase in mean MF.

We also evaluate the effectiveness and complexity of the proposed All-HiS-In ACA structure by comparing it with other multi-stage feature fusion strategies in accuracy and complexity. The complexity is presented in terms of model parameters, memory cost, FLOPs and required training time. Results are given in Table III. The Basic structure is the baseline in [9], which is constructed by adding a classification head at the top of ResNet-101. The Shared concatenation is the fusion strategy adopted by CASENet [9]. The bottom-up fusion with LAM structure is derived from MSC-SED [11]. The Dynamic feature fusion [10] replaces the concatenation in CASENet with pixel-level weighted summation. All the above methods are implemented by using their original source codes. The PCM is our implementation of the fusion strategy in [5], which has no official open-source code. For fair comparison, all the fusion strategy methods adopt the same backbone, i.e., ResNet-101; all the fusion methods, except the Basic method, utilize the last four stages of features obtained by the backbone, as done in the proposed All-HiS-In ACA. In addition, we remove unnecessary plugins and keep only fusion-related structures. The ICE and OSI in our method are also removed and only the All-HiS-In fusion is kept.

As seen from Table III, the Basic model is the simplest in complexity but the lowest in mean MF. Compared to the Basic model, the Shared concatenation, Bottom-up fusion with LAM, Dynamic feature fusion and the PCM models combine multi-stage features for classification in various position-aligned fusion ways. Among them, the Bottom-up fusion with LAM is the most complex one in terms of parameters, memory cost, FLOPs and training time cost. In contrast, the proposed All-HiS-In fusion method obtains the highest accuracy with a medium level of parameters, memory cost, FLOPs and training time cost. This can be easily read from Fig. 7, in which we visualize the increased accuracy of all the fusion structures versus their increased complexity, relative to the Basic model. It needs to mention that we develop the All-HiS-In ACA structure based on the most primitive version of the self-attention paradigm, whose memory cost and FLOPs can be significantly reduced while maintaining the accuracy,

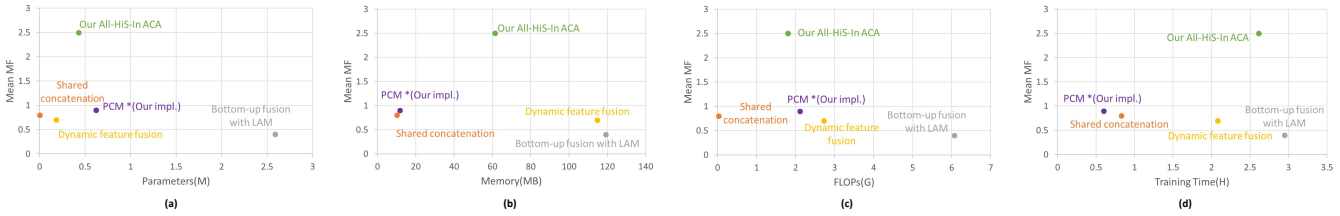


Fig. 7. Increased mean MF versus increased parameters(M), memory(MB), FLOPs(G) and training time(H) of all the methods relative to the Basic method. All the data are derived from Table III.

as proved in other tasks [46].

We also demonstrate the effectiveness of our All-HiS-In ACA structure in generating discriminative features for SED via visualization. Fig. 8 presents three input examples in the first column and their edge maps, which are predicted by using the edge branch in Fig. 3, in the second column. From the edge maps, it can be seen that the contour samples are highlighted but still contaminated by non-contour samples. All-HiS-In ACA can aggregate semantic context for non-contour and contour-class categorization. To show its effectiveness, we compute similarities between elements based on their aggregated context representations obtained by All-HiS-In ACA and visualize the similarities in Fig. 8. Specifically, we select one point (the red crosses in the first column of Fig. 8) on the contour and compute its similarities with the other elements based on their representations which encode all higher stages of semantic context with the lowest stage of details excluded. It can be observed that only the object-level contours are highlighted, while the inner edges and those from complex backgrounds are suppressed. Besides, we observe that our All-HiS-In ACA structure could capture semantic context dependencies across objects. For example, in the third column of Fig. 8, the hard samples on the edges of the small horse, which are wrongly classified by existing SED methods as seen from Fig. 1, share similar context as the easy-to-recognize samples on the edges of the large horse. Due to this ability, our method outperforms existing methods and obtains correct categorization on these hard samples, as seen from Fig. 1.

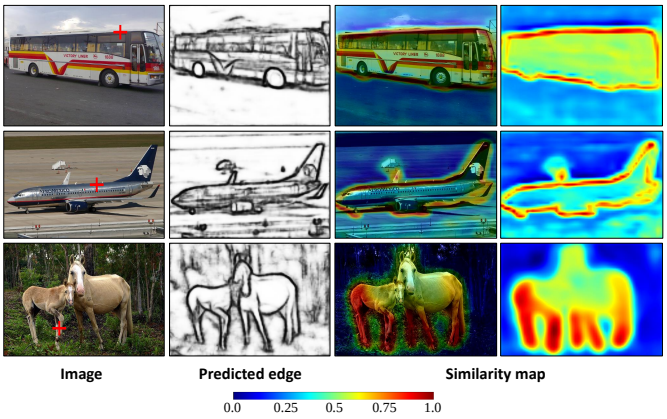


Fig. 8. Visualization of intermediate results. The first column shows original images with selected contour samples (indicated by the red crosses). The second column presents the predicted edges from the edge branch. We compute the similarities between the selected samples and the others in the same image, based on the aggregated context representations in All-HiS-In ACA, and form similarity maps. The third and fourth columns present the similarity maps overlaid on the input images and the maps themselves.

2) Ablation Study on ICE: We verify the two components in ICE, i.e., the complementary enhancement structure and the two branches of supervision, individually and together on SBD dataset, based on the backbone. As seen from Table II, the complementary enhancement and the two branches of supervision bring in improvements of 2.0% and 1.7%, respectively. The two components together, i.e. the whole ICE module, boost SED performance by 2.3%.

We also test the effectiveness of the combination of ICE and All-HiS-In ACA. From Table II, it can be seen that, the proposed All-HiS-In ACA plus the ICE module boosts the performance of SED to a mean MF (ODS) score of 76.9%, 3.3% higher than the baseline.

3) Ablation Study on OSI: The OSI module is designed based on ICE to refine the representations of samples belonging to the same object by referring to object-level semantics. Therefore, we cannot verify it alone as we did for the other modules. Here, we check the improvement brought in by adding the OSI module to the architecture of All-HiS-In ACA fusion plus ICE. From the last row of Table II, we obtain an extra performance gain of 0.4% by adding OSI.

E. Comparison with State-of-the-Art SED Methods

We quantitatively and qualitatively compare the proposed network with state-of-the-art SED methods, based on both the SBD dataset and Cityscapes dataset. All the quantitative results of the competitive methods are from their original papers. The competitive methods for the two datasets are not totally the same due to the availability of the quantitative results. In addition, since semantic segmentation results can be converted to semantic edges as we introduced in Section I, we compare the proposed SED with state-of-the-art semantic segmentation methods as well. Besides, SED methods can be used for contour detection as we analyzed in Section II-C. Therefore, we also test the proposed methods on contour edge detection.

1) Comparisons with SED methods on SBD: We compare the proposed network with methods in [9]–[11], [13], [18], [19], in MF (ODS) with matching distance tolerance of 0.02. All the methods adopt the same backbone of ResNet-101, and the same resolution of training and testing images. As it can be seen from Table IV, our network outperforms all the other networks and achieves a new state-of-the-art accuracy of 77.3% mean MF (ODS). In addition, in most classes, our network obtains higher accuracy than the other networks.

To better understand the superiority of the proposed method, we also visualize some prediction results in Fig. 1 and Fig. 9 for qualitative comparison. We chose CASENet [9], DFF [10], and MSC-SED [11], whose source codes are available for us, for comparison. From the figures, we can observe that the proposed network can predict thinner, cleaner, more complete and more semantically correct boundaries. Concretely, from the first and second rows of Fig. 9, it can be seen that the contour discontinuity problem is distinct in the results of the other methods but is alleviated much by our method. From the second and third rows of Fig. 9, it can be seen that the complex textures in the background are more effectively suppressed by our method, such as the pier and house in the second row and the fence in the third row. From the second and fourth rows of Fig. 9, we can see that the contours of the extremely small objects are more clearly structured and correctly classified by our method, some of which are not even annotated by humans in the ground truth (e.g., the small horse in the fourth row).

2) Comparisons with SED methods on Cityscapes: We compare the proposed network with several state-of-the-art SED networks in MF (ODS) with matching distance tolerance

TABLE IV
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE SBD TEST SET IN MF SCORES (%). THE MATCHING DISTANCE TOLERANCE IS 0.02.

Network	aer.	bike	bird	boat	bot.	bus	car	cat	cha.	cow	tab.	dog	hor.	mot.	per.	pot	she.	sofa	tra.	tv	Mean
CASENet [9]	83.6	75.3	82.3	63.1	70.5	83.5	76.5	82.6	56.8	76.3	47.5	80.8	80.9	75.6	80.7	54.1	77.7	52.3	77.9	68.0	72.3
SEAL [13]	84.5	76.5	83.7	64.9	71.7	83.8	78.1	85.0	58.8	76.6	50.9	82.4	82.2	77.1	83.0	55.1	78.4	54.4	79.3	69.6	73.8
STEAL [19]	84.5	77.3	84.0	65.9	71.1	85.3	77.5	83.8	59.2	76.4	50.0	81.9	82.2	77.3	81.7	55.7	79.5	52.3	79.2	69.8	73.8
DFF [10]	86.5	79.5	85.5	**69.0**	73.9	86.1	80.3	85.3	58.5	80.1	47.3	82.5	85.7	78.5	83.4	57.9	81.2	53.0	81.4	71.6	75.4
DDS [18]	86.5	78.4	84.4	67.0	**74.3**	85.8	80.2	85.9	60.4	80.8	**53.9**	83.0	84.4	78.8	**83.9**	58.7	81.9	56.0	82.1	73.0	76.0
MSC-SED [11]	86.1	78.8	85.4	68.2	74.2	87.7	80.6	86.1	60.9	83.6	50.4	85.2	86.1	78.8	82.7	**59.5**	84.0	56.9	**82.4**	70.7	76.4
Ours	**87.9**	**79.6**	**85.8**	68.8	73.7	**88.5**	**80.9**	**86.8**	**61.3**	**84.3**	51.8	**85.8**	**87.0**	**80.2**	**83.9**	59.1	**85.2**	**58.6**	**82.4**	**73.4**	**77.3**

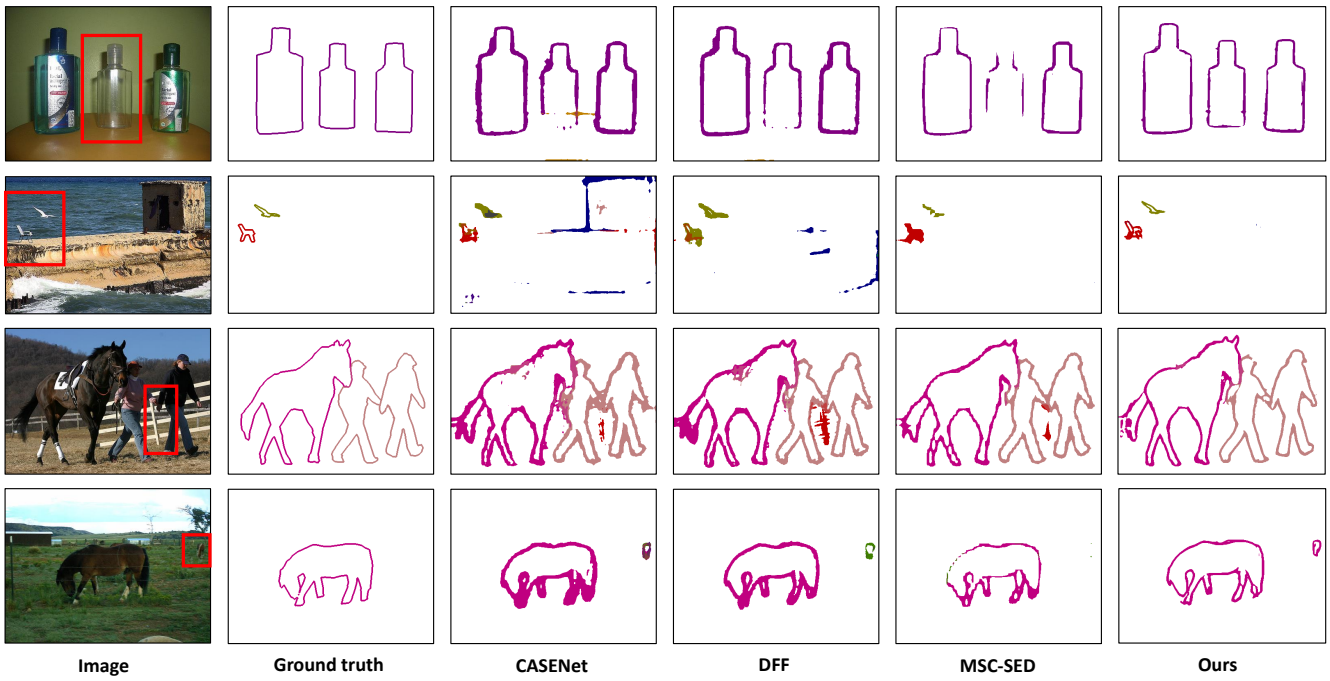


Fig. 9. Qualitative comparison of results obtained by CASENet [9], DFF [10], MSC-SED [11] and ours on SBD dataset. Best viewed in color. The red rectangular boxes indicate the areas where the advantages of our method can be seen more clearly.

of 0.0035. Quantitative results are given in the first part of Table V. From the table, it can be seen that our network outperforms all the other SED networks and achieves a new state-of-the-art performance of 76.0% on the val set. According to Table V, compared to DFF [10], our approach does not work better all the time. Our method outperforms DFF in categories with distinct contour shapes, such as person, bus, train, motorbike and bike. This is consistent with the design objectives of our adaptive context aggregation and object-level feature regularization. Even though, in those background-like categories, such as road, side, building and sky, which are easily obscured by foreground objects or have no specific structures, our method achieves performance comparable with DFF.

We also present the comparison in matching distance tolerance of 0.00375 in Table VI, with extra optional steps, including NMS, multi-scale test and random flipping [5], adopted by RPCNet [5]. From Table VI, it can be seen that in the case without using any additional operations, our network outperforms all the other networks by a large margin in mean MF (ODS). RPCNet with all the additional operations obtains the same mean MF score as ours. However, RPCNet adopts eight layers of PCM modules and therefore is much more complex than the proposed method.

For fair comparison with RPCNet, we also apply its additional operations on the proposed network. The parameters of the additional steps are all the same as RPCNet. The post-processing of NMS helps the refinement of the predicted semantic edges. Results in Table VI show that our network is superior to RPCNet by 0.3% in mean MF (ODS). Note that the improvement brought in by NMS to our network is only 0.3%, while for CASENet and SEAL, the improvements are all above 0.7%. This is a side evidence that our network itself has the ability to predict finer boundaries, rather than relying much on the post-processing of NMS, which cannot correct topological errors or recover high-frequency contour details.

We then qualitatively compare the proposed method with DFF [10] and MSC-SED [11], the two most recently proposed methods with source codes available, based on Cityscapes. Compared with SBD, Cityscapes dataset has more small objects and richer edges in each image. Therefore, thick predictions risk more topological errors and over-smooth contours, which cannot be corrected by NMS post-processing. As shown in Fig. 10, the prediction results of our network are thinner and

TABLE V
COMPARISON WITH STATE-OF-THE-ART SED AND SEMANTIC SEGMENTATION METHODS ON THE CITYSCAPES VAL SET IN MF SCORES (%). THE MATCHING DISTANCE TOLERANCE IS SET TO 0.0035.

Network	road	sid.	bui.	wall	fen.	pole	lig.	sign	veg.	ter.	sky	per.	rid.	car	tru.	bus	tra.	mot.	bike	Mean
CASENet [9]	86.2	74.9	74.5	47.6	46.5	72.8	70.0	73.3	79.3	57.0	86.5	80.4	66.8	88.3	49.3	64.6	47.8	55.8	71.9	68.1
SEAL [13]	87.6	77.5	75.9	47.6	46.3	75.5	71.2	75.4	80.9	60.1	87.4	81.5	68.9	88.9	50.2	67.8	44.1	52.7	73.0	69.1
STEAL [19]	87.8	77.2	76.4	49.5	49.2	74.9	73.2	76.3	80.8	58.9	86.8	80.2	69.0	83.2	52.1	67.7	53.2	55.8	72.8	69.7
DFF [10]	**89.4**	**80.1**	**79.6**	51.3	54.5	**81.3**	81.3	**81.2**	**83.6**	**62.9**	**89.0**	85.4	75.8	**91.6**	54.9	73.9	51.9	64.3	76.4	74.1
MSC-SED [11]	88.5	77.2	76.4	**57.2**	**57.0**	70.8	78.6	80.1	80.1	60.3	86.5	82.5	75.3	90.3	**66.5**	80.5	68.9	67.6	75.1	74.7
Ours	87.9	78.7	77.3	53.8	54.4	80.5	**81.5**	81.1	82.4	61.1	88.7	**85.5**	**77.7**	91.2	64.6	**82.6**	**69.2**	**68.2**	**77.7**	**76.0**
HyperSeg [47]	53.7	73.0	67.3	39.0	36.3	73.5	62.6	66.5	76.0	54.3	78.1	50.9	57.4	70.2	44.3	67.0	38.9	46.5	60.7	58.8
SegFormer [48]	63.4	78.5	77.0	54.7	53.7	82.1	81.1	77.8	82.2	60.2	89.0	79.3	74.9	81.8	62.7	81.4	66.8	64.4	73.7	72.9

TABLE VI
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE CITYSCAPES VAL SET IN MF SCORES (%). CASENET * IS THE REIMPLEMENTATION OF CASENET IN [19]. THE MATCHING DISTANCE TOLERANCE IS SET TO 0.00375. TEST NMS, MS_FLIP REPRESENT THE POST-PROCESSING NMS, MULTI-SCALE TEST AND RANDOM FLIPPING, RESPECTIVELY.

Network	Test NMS	MS_Flip	road	sid.	bui.	wall	fen.	pole	lig.	sign	veg.	ter.	sky	per.	rid.	car	tru.	bus	tra.	mot.	bike	Mean
CASENet [9]			87.1	76.0	75.7	46.9	47.7	73.2	72.7	75.7	80.4	57.8	86.7	81.0	67.9	89.1	45.9	68.1	49.6	54.2	73.7	68.9
CASENet *			87.2	76.1	75.7	47.9	47.6	73.7	71.8	75.2	80.6	58.4	86.8	81.0	68.2	89.3	49.0	67.8	50.8	55.3	74.2	69.3
CASENet *	√		88.1	76.5	76.8	48.7	48.6	74.2	74.5	76.4	81.3	59.0	87.3	81.9	69.1	90.3	50.9	68.4	52.1	56.2	75.7	70.3
STEAL [19]			88.1	77.6	77.1	50.0	49.6	75.5	74.0	76.7	81.5	59.4	87.2	81.9	69.9	89.5	52.2	67.8	53.6	55.9	75.2	70.7
STEAL [19]	√		88.9	78.2	77.8	50.6	50.4	75.5	76.3	77.5	82.3	60.2	88.0	82.5	70.2	90.4	53.3	68.5	53.4	57.0	76.1	71.4
RPCNet [5]	√	√	**90.9**	**82.3**	**82.1**	57.2	59.0	**84.5**	**83.3**	**82.3**	**84.9**	**64.2**	89.9	86.3	78.5	**92.6**	67.8	82.8	68.5	69.2	**80.1**	78.2
Ours			89.2	80.4	79.4	59.4	59.0	81.2	82.8	82.0	83.6	63.5	89.9	86.5	79.2	92.0	70.2	84.9	72.6	70.5	79.6	78.2
Ours	√	√	89.5	80.6	80.0	**59.6**	**59.1**	82.7	83.0	82.0	84.3	63.6	**90.2**	**86.5**	**79.5**	92.1	**70.2**	**85.1**	**72.9**	**70.6**	80.0	**78.5**

more accurate, e.g., the contours of the traffic signs or poles are clearer in structure and more consistent with the ground truth data.

F. Comparison with Semantic Segmentation Methods

We compare the proposed method with state-of-the-art semantic segmentation algorithms, including HyperSeg [47] and SegFormer [48], in semantic edge detection by converting their resulted masks into semantic boudaries. HyperSeg and SegFormer are chosen as representatives of CNN-based and vision transformer-based semantic segmentation algorithms, respectively. Both of them have open-source codes. Results on Cityscapes are shown in Table V. According to the table, semantic segmentation methods have no advantages in locating and categorizing semantic edges. As we analyzed in Section I, semantic segmentation is error-prone in classifying boundaries pixels due to the complex context around the points. The proposed SED method achieves the best performance. The CNN-based DFF and MSC-SED also outperform the two semantic segmentation methods.

G. Applying Our Method to Contour Edge Detection

We verify the effectiveness of the proposed method on contour edge detection. Following [11], we conduct experiments on the SBD dataset. The experimental results are shown in Table VII, where the values of HED [33] , RCF [34] , BDCN [14] and MSC-SED [11] are from [11]. DexiNed-f and DexiNed-a [38] and RefineContourNet [37] are retrained based on the SBD dataset without semantics with the provided source codes. We adopt the same metrics of ODS, OIS and AP under two matching distance tolerances as [11] for evaluation on contour detection. As seen from Table VII, the SED methods, i.e., MSC-SED and our method, outperform non-SED contour detection methods by large margins. This is probably because the two methods are originally designed for semantic edge detection in favor of capturing category-relevant clues which contour detection requires. And the proposed model even utilizes categories as supervision to learn category-relevant features.

TABLE VII
COMPARISON WITH CONTOUR EDGE DETECTION METHODS BASED ON THE SBD DATASET IN F-MEASURE AT OPTIMAL DATASET SCALE (ODS), OPTIMAL IMAGE SCALE (OIS) AND AVERAGE PRECISION (AP) UNDER TWO MATCHING DISTANCE TOLERANCES.

	Matching distance tolerance = 0.02			Matching distance tolerance = 0.0075		
	ODS	OIS	AP	ODS	OIS	AP
HED [33]	71.9	74.9	75.2	62.6	64.2	61.5
RCF [34]	73.5	76.3	76.5	64.4	66.0	62.5
BDCN [14]	76.8	79.3	72.8	68.1	69.5	59.6
RefineContourNet [37]	67.1	70.1	68.8	57.5	59.4	52.6
DexiNed-f [38]	71.1	74.1	68.8	61.9	63.8	53.9
DexiNed-a [38]	70.2	73.0	70.7	60.6	62.4	55.4
MSC-SED [11]	82.5	84.7	84.1	72.5	73.6	69.1
Ours	**84.3**	**86.4**	**84.6**	**74.9**	**76.3**	**69.7**

Qualitative results are given in Fig. 11. From the figure, it can be observed that beside of ours and MSC-SED [11], the other methods generate noisy edges inside or outside objects. To show the advantages of our method over MSC-SED, we present several close-up views of their results in Fig. 12. According to the figure, the proposed method obtains much smoother edges than MSC-SED, which means pixels near edges can be more correctly classified as edges or not. Methodologically, compared to the bottom-up fusion in MSC-

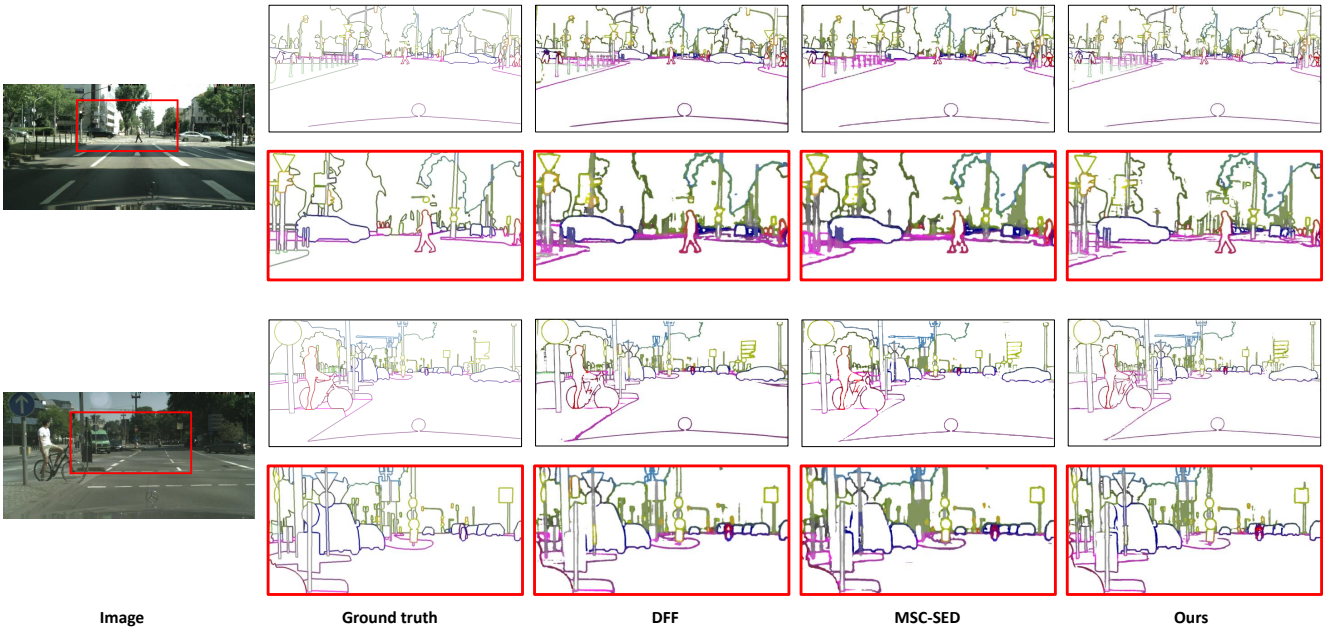


Fig. 10. Qualitative comparison of the results obtained by DFF [10], MSC-SED [11] and our method on Cityscapes. Best viewed in color. The red rectangular boxes indicate the enlarged areas.

SED, our adaptive context aggregation can better keep the detail structures while aggregating rich semantics in the fused features for classification.

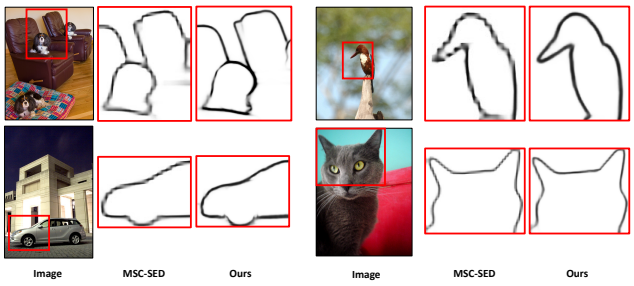


Fig. 12. Close-up views of the contour detection results obtained by MSC-SED [11] and ours. The red rectangular boxes indicate the enlarged areas.

V. Conclusion

The paper presented an effective multi-stage CNN feature fusion strategy, called All-HiS-In ACA, for SED. It is an effective complement to existing multi-stage CNN feature fusion strategies in the field of SED. Based on this fusion strategy, we developed a deep network which can adaptively aggregate clues from all higher-stage features and fuse them with low-stage features. The fused features keep the high-resolution detail features for accurate edge localization and contain rich semantic context for semantic categorization. In addition, we proposed a non-parametric ICE module, to enhance features according to inter-layer complementary clues. We also presented an OSI module, which can refine the fused features according to the object-level semantics. Due to the high-quality fused features, the proposed network outperforms state-of-the-art SED and semantic segmentation methods in edge localization and categorization. The proposed method was also applied to contour edge detection and showed superior performance to state-of-the-art contour detection methods. We believe that the proposed SED method and its ideas in fusion and non-parametric inter-layer complementary enhancement will benefit both SED and other related fields, e.g., edge-enhanced semantic segmentation and SED-based applications.

The proposed method still has limitations. Firstly, as we can see from Fig. 10, our method and the other methods cannot well handle partially-occluded targets, which are common in indoor and street scenes. In the future, multi-view SED will be considered, to better deal with the occlusion problem by exploiting complementary information from multiple views.

Secondly, we try to test both the proposed method and MSC-SED [11], the two performing the best in the experiments in Section IV, on the much more challenging dataset ADE20K [49]. ADE20K contains various scenes with stuff/things of 150 categories, which means each edge point has a solution space much larger than those of the SBD and Cityscapes datasets, in the multi-label learning framework [9] widely adopted by SED algorithms including ours. Besides, the samples among different categories in ADE20K are extremely imbalanced. 50% samples belong to only six of the 150 categories. Due to the above challenges, the mean MF score of the proposed method trained with the raw ADE20K data (no data augmentation) over all classes and that over the 6 categories are only 21% and 33.8%. Those of MSC-SED are 20.2% and 31.9%, respectively. In our future research, we will take the above challenges into consideration.

In addition, SED evaluation usually uses MF scores, which are not enough to reflect the true quality of semantic contours. In the future, we will propose supplementary metrics for SED, in aspects of the thickness, completeness and topological

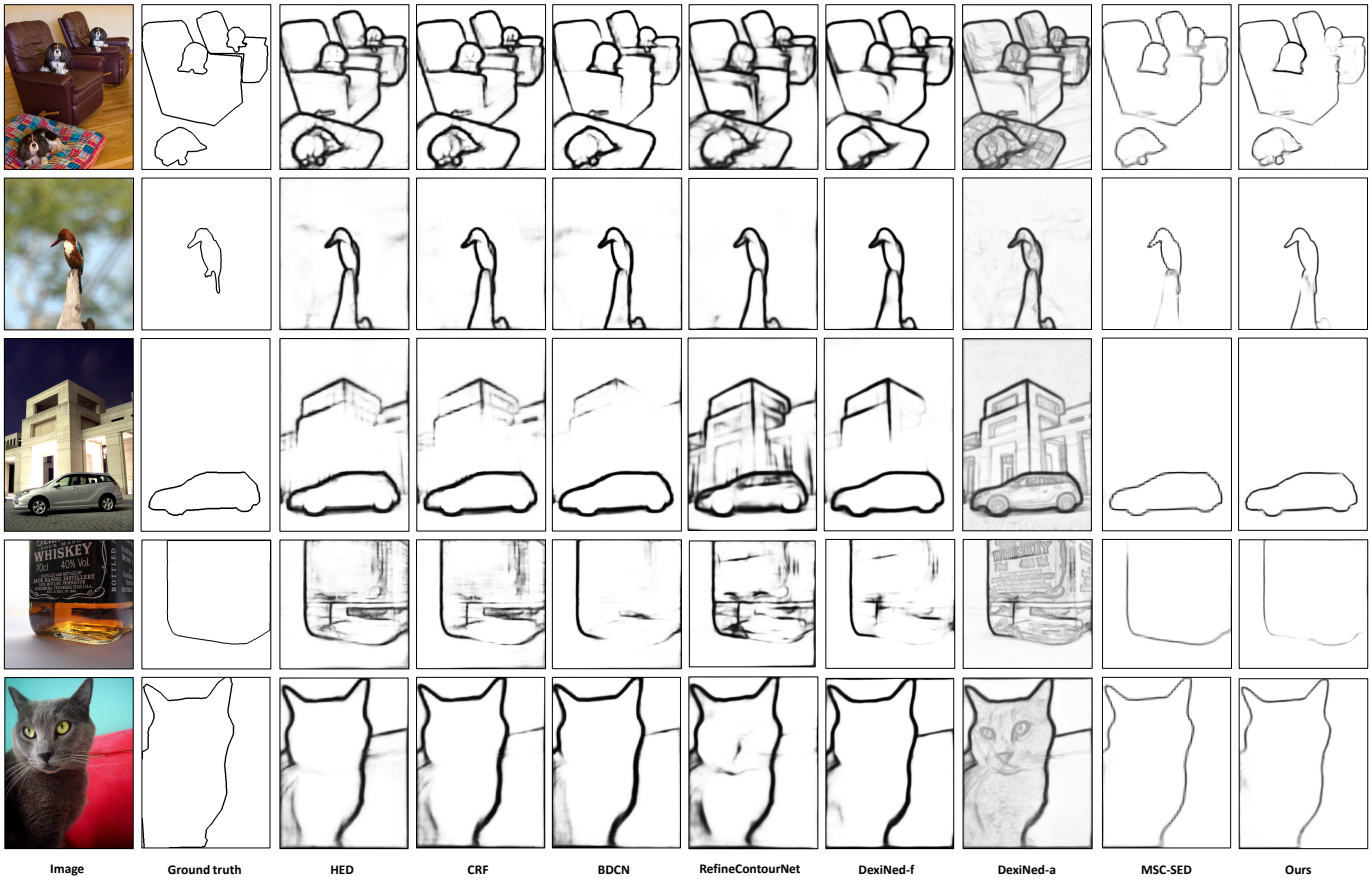


Fig. 11. Qualitative comparison of contour detection results obtained by HED [33], RCF [34], BDCN [14], RefineContourNet [37], DexiNed [38], MSC-SED [11] and ours. The test images are from the SBD dataset.

correctness of semantic edges.

REFERENCES

[1] T. Qin, T. Chen, Y. Chen, Q. Su, "AVP-SLAM: Semantic visual mapping and localization for autonomous vehicles in the parking lot," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2020, pp.5935-5945.

[2] S. Ramalingam, S. Bouaziz, P. F. Sturm, and M. Brand, "SKYLINE2GPS: localization in urban canyons using omni-skylines," in *Proc. IEEE/RS Int. Conf. Intell. Roboots Syst. (IROS)*, Oct. 2010, pp. 3816-3823.

[3] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 654-661.

[4] X. Weng, Y. Yan, S. Chen, J.-H. Xue, and H. Wang, "Stage-aware feature alignment network for real-time semantic segmentation of street scenes," *IEEE Trans. Circuits Syst. Video Technol.*, early access, 2021, DOI: 10.1109/TCSVT.2021.3121680.

[5] M. Zhen, J. Wang, L. Zhou, S. Li, Q. Long, "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13663-13672.

[6] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5228-5237.

[7] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang, "Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6459-6468.

[8] C. Tang, H. Chen, X. Li, J. Li, Z. Zhang, and X. Hu, "Look closer to segment better: Boundary patch refinement for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13921-13930.

[9] Z. Yu, C. Feng, M. Liu, and S. Ramalingam, "CASENet: Deep category-aware semantic edge detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1761-1770.

[10] Y. Hu, Y. Chen, X. Li, and J. Feng, "Dynamic feature fusion for semantic edge detection," in *Proc. of the 28th Int. Joint Conf. on Art. Int.*, Aug. 2019, pp. 782-788.

[11] W. Ma, C. Gong, S. Xu, and X. Zhang, "Multi-scale spatial context-based semantic edge detection," *Inf. Fusion*, vol. 64, pp. 238-251, Aug. 2020.

[12] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679-698, Nov. 1986.

[13] Z. Yu, W. Liu, Y. Zou, C. Feng, S. Ramalingam, B. V. K. V. Kumar, and J. Kautz, "Simultaneous edge alignment and learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 400-417.

[14] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bi-directional cascade network for perceptual edge detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3828-3837.

[15] B. Hariharan, P. Arbelaez, L. D. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 991-998.

[16] G. Bertasius, J. Shi, and L. Torresani, "High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 504-512.

[17] K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. V. Gool, "Convolutional oriented boundaries: From image segmentation to high-level tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 819-833, Apr. 2018.

[18] Y. Liu, M. -M. Cheng, D. -P. Fan, L. Zhang, J. Bian, and D. Tao, "Semantic edge detection with diverse deep supervision," *Int. J. Comput. Vis.*, vol. 130, no. 1, pp. 179-198, Nov. 2022.

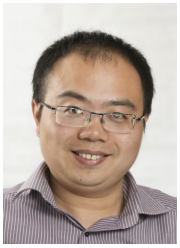
[19] D. Acuna, A. Kar, and S. Fidler, "Devil is in the edges: Learning semantic boundaries from noisy annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11075-11083.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional net-

works for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.(MICCAI)*, Springer, Oct. 2015, pp. 234-241.
[21] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834-848, Apr. 2018.
[22] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: https://arxiv.org/abs/1706.05587.
[23] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 833-851.
[24] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation," 2019, *arXiv:1903.11816*. [Online]. Available: https://arxiv.org/abs/1903.11816.
[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431-3440.
[26] Y. Tian and S. Zhu, "Partial domain adaptation on semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, early access, 2021, DOI: 10.1109/TCSVT.2021.3116210.
[27] X. Zhang, H. Li, F. Meng, Z. Song, and L. Xu, "Segmenting beyond the bounding box for instance segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 704-714, Feb. 2022.
[28] G. Bertasius, J. Shi, and L. Torresani, "Semantic segmentation with boundary neural fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3602-3610.
[29] H. Ding, X. Jiang, A. Q. Liu, N. Magnenat-Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6818-6828.
[30] J. Ji, R. Shi, S. Li, P. Chen, and Q. Miao, "Encoder-decoder with cascaded CRFs for semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 5, pp. 1926-1938, May 2021.
[31] G. Bertasius, J. Shi, and L.Torresani, "DeepEdge: A multi-scale bifurcated deep network for top-down contour detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4380-4389.
[32] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3982-3991.
[33] S. Xie, and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395-1403.
[34] Y. Liu, M. -M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5872-5881.
[35] Y. Wang, X. Zhao, and K. Huang, "Deep crisp boundaries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1724-1732.
[36] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 562-578.
[37] A. P. Kelm, V. S. Rao, and U. Zlzer, "Object contour and edge detection with RefineContourNet," in *Proc. Int. Conf. Comput. Anal. Images and Patterns. (CAIP)*, Sep. 2019, pp. 246-258.
[38] X. Soria, E. Riba, and A. Sappa, "Dense extreme inception network: Towards a robust CNN model for edge detection," in *Proc. IEEE Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1912-1921.
[39] R. Deng, and S. Liu, "Deep structural contour detection," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, Oct. 2020, pp. 304-312.
[40] Z. Su, W. Liu, Z. Yu, D. Hu, Q. Liao, Q. Tian, M. Pietikäinen, and L. Liu, "Pixel difference networks for efficient edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5117-5127.
[41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, May. 2021.
[42] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 173-190.
[43] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 570-586.
[44] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213-3223.
[45] T. Lin, M, Maire, S. Belongie, J. Hays, C. Zitnick, "Microsoft COCO: common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 740-755.
[46] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. Huang, "CCNet: Criss-cross attention for semantic segmentation," *IEEE Trans. on Pattern Anal. Mach. Intell.*, early access, 2020, DOI: 10.1109/TPAMI.2020.3007032.
[47] Y. Nirkin, L. Wolf, and T. Hassner, "HyperSeg: Patch-wise hypernetwork for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4060-4069.
[48] E. Xie, E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Inf. Proc. Sys. (NeurIPS)*, vol. 34, Dec. 2021.
[49] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K Dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nov. 2017, pp. 5122-5130.

Qihan Bo was born in 1997. He received the B.S. degree in Electronic and Information Engineering from Beijing University of Technology, Beijing, China in 2019. He is currently pursuing the M.S. degree in Beijing University of Technology. His research interests include Computer Vision and Deep Learning.

Wei Ma received her Ph.D. degree in Computer Science from Peking University, in 2009. She is currently an Associate Professor at Faculty of Information Technology, Beijing University of Technology. Her current research interests include image/video repairing, image/video semantic understanding and 3D vision.



Yu-Kun Lai is a Professor at School of Computer Science and Informatics, Cardiff University, UK. He received his B.S and PhD degrees in Computer Science from Tsinghua University, in 2003 and 2008 respectively. His research interests include computer graphics, computer vision, geometric modeling and image processing. For more information, visit https://users.cs.cf.ac.uk/Yukun.Lai/

Hongbin Zha received the B.E. degree from the Hefei University of Technology, China, in 1983 and the M.S. and Ph.D. degrees from Kyushu University, Japan, in 1987 and 1990, respectively. After working as a Research Associate in the Kyushu Institute of Technology, Japan, he joined Kyushu University, Japan, in 1991 as an Associate Professor. Since 2000, he has been a Professor at the Key Laboratory of Machine Perception (Ministry of Education), Peking University, China. His research interests include computer vision, digital geometry processing, and robotics. He has published more than 350 technical publications in journals, books, and international conference proceedings.