Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 1616055, 16 pages
https://doi.org/10.1155/2023/1616055

Research Article

TriangleNet: Edge Prior Augmented Network for Semantic Segmentation through Cross-Task Consistency

Dan Zhang,[1] **Rui Zheng**,[1] **Gadeng Luosang**,[2] **and Pei Yang**[3]

[1] *Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China, Beijing, China*
[2] *Department of Information Science and Technology, Tibet University, Lhasa 850012, China*
[3] *Department of Computer Technology and Application, Qinghai University, Xining 810016, China*

Correspondence should be addressed to Rui Zheng; rzhengbj@163.com

Received 29 November 2022; Revised 1 August 2023; Accepted 24 August 2023; Published 16 September 2023

Academic Editor: Alexander Hošovský

This paper addresses the task of semantic segmentation in computer vision, aiming to achieve precise pixel-wise classification. We investigate the joint training of models for semantic edge detection and semantic segmentation, which has shown a promise. However, implicit cross-task consistency learning in multitask networks is limited. To address this, we propose a novel "decoupled cross-task consistency loss" that explicitly enhances cross-task consistency. Our semantic segmentation network, TriangleNet, achieves a substantial 2.88% improvement over the Baseline in mean Intersection over Union (mIoU) on the Cityscapes test set. Notably, TriangleNet operates at 77.4% mIoU/46.2 FPS on Cityscapes, showcasing real-time inference capabilities at full resolution. With multiscale inference, performance is further enhanced to 77.8%. Furthermore, TriangleNet consistently outperforms the Baseline on the FloodNet dataset, demonstrating its robust generalization capabilities. The proposed method underscores the significance of multitask learning and explicit cross-task consistency enhancement for advancing semantic segmentation and highlights the potential of multitasking in real-time semantic segmentation.

1. Introduction

The combination of image semantic segmentation and deep learning has gone through a long period of time, accumulating a large number of excellent works such as FCN [1], U-Net [2], FastFCN [3], Gated-SCNN [4], DeepLab Series [5–7], Mask R-CNN [8], and so on, as well as leaving unsolved problems. The main challenge is the fine-grained localization of pixel labels [9]. The prevailing structure of semantic segmentation networks mostly follows the encoder-decoder structure adopted by the FCN [1]. First, downsampling is used to expand the receptive field to extract high-level semantics, and then, upsampling is used to recover low-level details. The edge details lost by conventional downsampling operations in semantic segmentation networks are difficult to recover during upsampling. A compensatory solution is to introduce additional knowledge among which edge priors are intuitive and easily accessible. In order to inject edge priors into semantic segmentation networks, one way is to train a semantic edge detection model and a semantic segmentation model jointly. General practice is a two-stream framework that trains a semantic edge detection branch and a semantic segmentation branch in a hard parameter-sharing manner [10]. The predictions of the semantic edge detection branch on edge points may differ from those of the semantic segmentation branch, which implies the existence of cross-task inconsistency. Conventionally, a fusion module is introduced to cope with this conflict, such as the study in [11, 12] does, which intends to fuse features from the semantic edge detection branch to improve the semantic segmentation branch. However, the effects of these fusion modules are sometimes not as effective as expected. As the ablation experiments of the study in [11] points out, the improvement of the mean of class-wise Intersection-over-Union (mIoU) on the Cityscapes validation set mainly depends on duality loss (+1.44%) rather than semantic edge fusion (+0.22%) or pyramid context module (+0.62%). A considerable amount of segmentation errors along object boundaries still exist, which means the mutual consistency between the semantic segmentation branch and the semantic edge detection branch should be further studied to improve the quality of segmentation results.

We have observed that many semantic segmentation works can be loosely viewed as semantic edge detection tasks, since applying edge detectors to semantic segmentation outputs can yield semantic edge results. Their relationship can be modeled as shown in Figure 1. Logically, in order to conserve consistency among tasks, the results of inferring semantic edges from an input image should be the same regardless of the inference paths, that is, predicting semantic edges by first predicting semantic segmentation maps from an input image should achieve similar predictions as directly predicting semantic edges from the input image. This observation aligns with the concept of inference-path invariance, which serves as the guiding ideology in the work by Zamir et al. [13]. The concept emphasizes that predictions should remain consistent regardless of the specific inference paths. The input image domain, the semantic segmentation domain, and the semantic edge domain form an elementary consistency unit proposed by Zamir et al. [13], which is illustrated in Figure 1.

By imposing a cross-task consistency loss on the endpoint outputs of the two paths, the consistency between semantic segmentation and semantic edge detection can be explicitly learned. Based on these analyses, we propose a new framework to simultaneously train a semantic segmentation branch and a semantic edge detection branch, and the overall process is shown in Figure 2.

The highlights of this paper are as follows:

(1) Figure 3 illustrates the superior balance between speed and accuracy achieved by our framework on the Cityscapes dataset, distinguishing it as one of the few models capable of real-time inference at full resolution. Notably, our model operates at an impressive 77.4% mIoU while maintaining a fast frame rate of 46.2 FPS on Cityscapes.

(2) We introduce a novel approach, “decoupled cross-task consistency loss,” to explicitly enhance cross-task consistency between semantic edge detection and semantic segmentation, resulting in 1.83% improvement in mIoU on the Cityscapes test set. The decoupled loss effectively enforces consistency across tasks, facilitating the learning of shared representations and leading to improved overall performance.

(3) Our model demonstrates exceptional efficacy in categories characterized by distinct edges and boundaries, as evidenced by some categories achieving significant IoU improvements, with “train” nearly reaching an 18% increase in IoU on the Cityscapes test set. These results further reinforce the importance of incorporating edge information through our approach, highlighting its impact on enhancing segmentation performance.

(4) The decoupled architecture we have designed allows for joint training of multiple tasks without the need for fusion modules during inference, thereby avoiding the introduction of extra inference overhead. This efficient and practical approach enables us to leverage the advantages of multitasking for real-time semantic segmentation without compromising on performance.

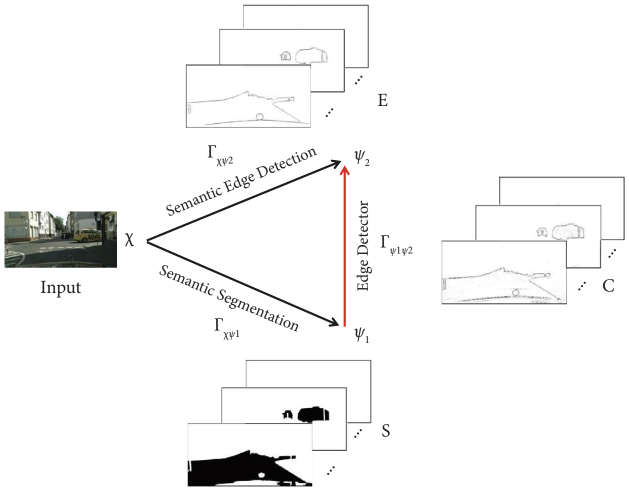


Figure 1: The multitask learning framework of semantic segmentation and semantic edge detection coincides with the elementary consistency unit theory where the prediction $\chi \longrightarrow \psi_1$ is enforced to be consistent with $\chi \longrightarrow \psi_2$ using a function that relates ψ_1 to ψ_2. S, E, and C, respectively, denote the outputs processed through $\Gamma_{\chi\psi_1}$, $\Gamma_{\chi\psi_2}$, and $\Gamma_{\psi_1\psi_2}$.

2. Related Work

2.1. Semantic Segmentation. Strengths, weaknesses, and major challenges of semantic segmentation are extensively discussed in the literature [9, 14–16]. There are currently two approaches to semantic segmentation: improving the object’s inner consistency or refining details along objects’ boundaries.

The inner inconsistency of the object is attributed to the limited receptive field, by which the longer range relationships of pixels in an image cannot be fully modeled. Consequently, the dilated convolution [17] or high-resolution network [18] is introduced to enlarge the receptive field. Furthermore, many attempts have been made to capture contextual information, such as recurrent networks [19, 20], pyramid pooling module [21], graph convolutional networks [22], CRF-related networks [5, 6, 23], nonlocal operator [24], and attention mechanism [25, 26].

The ambiguity along edges is caused by downsampling operations in the FCNs that result in blurred predictions. It is difficult to recover spatial information lost during downsampling through simple upsampling. Thus, previous papers have made efforts to add priors to guide the upsampling process, many of which focus on the use of edge

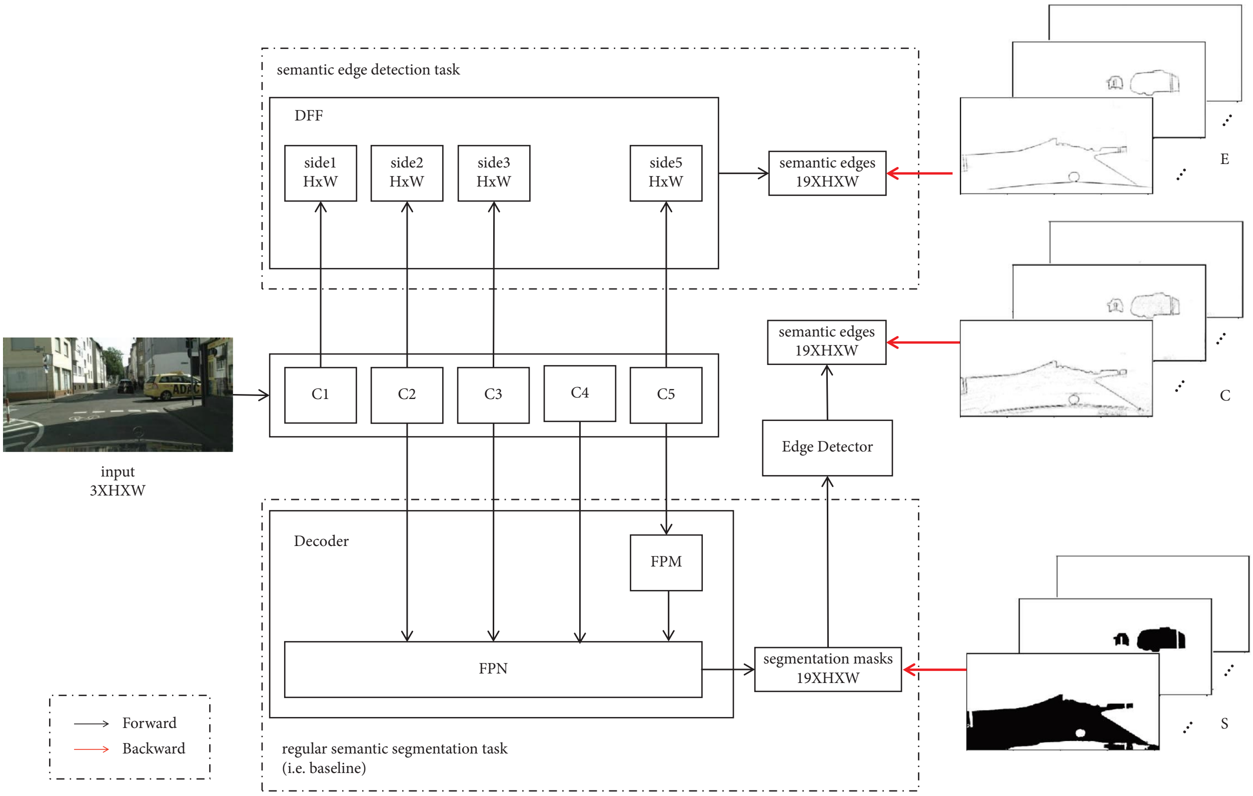


FIGURE 2: The overall pipeline of TriangleNet. The shared backbone network produces 5-layer features. The task-specific parts of the two branches are enclosed by dashed boxes.

priors. The general practice is a two-stream framework that trains an edge detection branch and a semantic segmentation branch jointly, which will be elaborated later.

2.2. Multitask Learning. Driven by deep learning, many dense prediction tasks such as semantic segmentation and instance segmentation have achieved significant performance improvements. Typically, tasks are learned in isolation, i.e., each task is trained with a separate neural network. Recently, multitask learning (MTL) techniques that learn shared representations by jointly processing multiple tasks have shown promising results.

Almost all theories about MTL are based on the assumption that tasks learned together should be relevant or a phenomenon called negative transfer would occur. In practice, it is more dependent on expert experience to find relevant tasks. For example, [27–30] jointly train semantic segmentation and depth estimation to achieve better results, [31, 32] jointly train semantic segmentation and instance segmentation to increase accuracy, and [4, 11, 12, 33, 34] jointly train semantic segmentation and edge detection to improve metrics. Among these, the edge priors can be further subdivided into binary edge priors and semantic edge priors. For example, in the GSCNN [4], the binary edge is used as a gate to improve performance. In BFP [33], binary edge information is used to propagate local features within regions. The study by [34] adopts domain transform to perform edge-preserving filtering controlled by a binary edge map derived from a task-specific edge detection task. The study by [12] applies explicit semantic boundary supervision to learn semantic features and edge features in parallel and an attention-based feature fusion module to combine the high-resolution edge features with wide-receptive-field semantic features. The RPCNet [11] presents an interacting multitask learning framework for semantic segmentation and semantic boundary detection.

The most common multitask learning framework shares some layers in the feature extraction stage and designs independent layers for each specific task, which is called the hard parameter-sharing approach [10]. This approach makes it difficult to ensure that multiple tasks can work together. Although there have been some means of using uncertainty [35] to determine weights of tasks, the relationship between tasks is still not very clear, which drives the study of explicit consistency constraints between tasks.

2.3. Consistency Learning. It has been speculated that multitask networks may automatically produce cross-task consistent predictions since their representations are shared. Numerous studies [13, 36–38] have observed that this is not necessarily true, since consistent learning is not enforced directly during training, indicating the need for explicit enhancement of consistency during learning.

From the literature, two kinds of explicit consistency constraints can be summarized. One idea is formulated as the cross-task consistency theory based on inference path

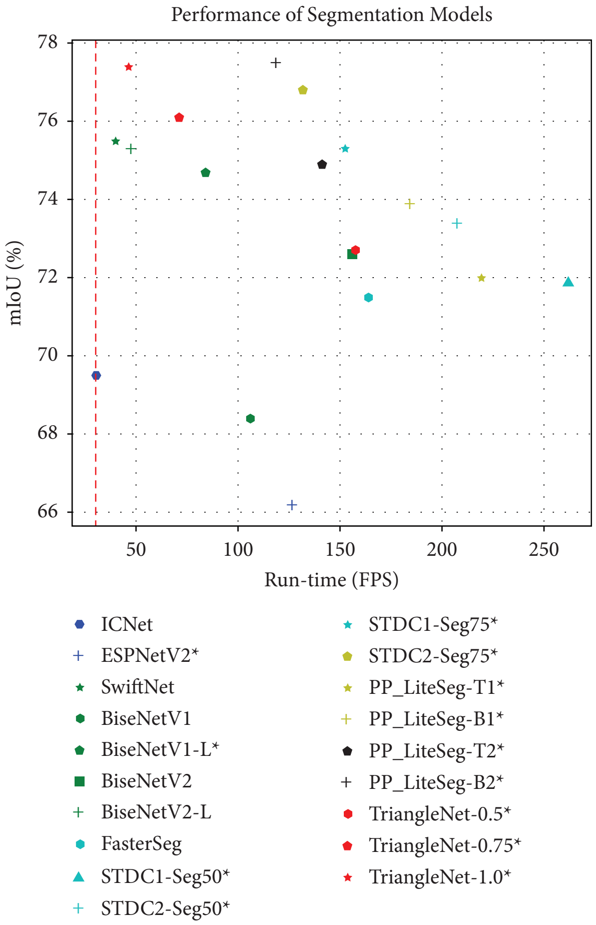


Figure 3: Run-time/accuracy trade-off comparison on the Cityscapes test set. Our models (in red) achieve an excellent run-time vs. accuracy trade-off among all previous real-time methods. FPS = 30 is the red line dividing real-time and non-real-time performance in the graph. The asterisk after the model name indicates that the inference speeds of these models were obtained using the same deep learning framework, PaddlePaddle, and the same hardware platform, A100 40G device.

invariance by the study in [13]. The authors in [13] first analyse the cross-task consistency theory of the triangular shown in Figure 1 and deduce the formula based on the l_1 norm assumption. Then, they generalize to cases where in the larger system of domains, consistency can be enforced using invariance along arbitrary paths, as long as their endpoints are same. The study by [39] conveys the same insight, which uses the predictions of one task as input to another network to predict the other task, obtaining task-transferred predictions. Explicit constraints are imposed between the transferred predictions and the prediction of the other task.

Another idea is that for a specific geometric feature, such as the boundary, the results extracted by different tasks should be consistent. For instance, the authors in [40] force the depth border to be consistent with the segmentation border through morphing. The authors in [41] penalize differences between the edges of the semantic heatmap and the edges of the depth map through a holistic consistency loss.

2.4. Real-Time Semantic Segmentation.

Real-time semantic segmentation is a challenging and essential task in computer vision, aiming to perform pixel-wise classification with high accuracy and rapid inference speed. The demand for real-time processing in applications such as autonomous vehicles, robotics, and augmented reality has driven extensive research efforts to develop efficient algorithms and architectures. Attempts to achieve a balance between speed and accuracy in real-time semantic segmentation include efficient architectures [42, 43], lightweight convolutions [44, 45], knowledge distillation [46–48], pruning and quantization [49, 50], and optimization techniques [51].

Multitask learning has shown a promise in various computer vision tasks, but it is relatively less popular in real-time semantic segmentation. One of the challenges in deploying multitask learning for real-time semantic segmentation is the potential increase in inference overhead due to fusion modules or additional processing steps. While multitask learning can be beneficial during training by leveraging shared representations and learning complementary features from related tasks, the goal is to incorporate this knowledge effectively without introducing extra inference time. To achieve this, researchers are exploring methods to learn shared representations without the need for explicit fusion modules during inference. Some approaches to address this concern include decoupled architectures [52], knowledge distillation, shared layers [53], and weight sharing [42, 43].

2.5. Edge Detection.

One notable method of applying deep neural networks to train and predict edges in an image-to-image fashion and end-to-end training is the holistically nested edge detection (HED) [54]. HED is a binary edge detection network, where the edge pixels are all set to 1 and 0 otherwise. Practically, edge pixels appear in contours or junctions belonging to two or more semantics, resulting in a challenging category-aware semantic edge detection problem. A pioneering approach is given by CASENet [55] that extends the work of HED. However, both HED and CASENet employ fixed weight fusion to merge side outputs, ignoring image-specific and location-specific information. To address this, DFF [56] designed an adaptive weight fusion module to assign different fusion weights for different input images and locations adaptively.

In essence, compared to binary edge detection, semantic edge detection is more coupled with semantic segmentation since it provides semantic information about edge pixels while locating edges.

3. Methods

In this section, we will first introduce the overview pipeline of our architecture illustrated in Figure 2 and then explain the components in detail.

3.1. Model Overview.

As shown in Figure 2, the overall network is a two-stream framework following a hard parameter-sharing manner [10]. It contains two branches:

the upper one is an implementation of DFF [56] responsible for semantic edge detection and the lower one is an FCN for semantic segmentation integrated with PPM [21] and FPN [57] as the decoder. The backbone of the FCN is replaceable, and the features extracted by the backbone are shared by the semantic edge detection branch and the semantic segmentation branch. Except for the shared backbone layers, the other layers of the two branches are task-specific and parallel. An edge detector is used to transfer the segmentation maps to semantic edges, which are enforced to be consistent with the output of the semantic edge detection branch by a consistency loss. We call this network TriangleNet because its underlying theory can be formulated by a triangular relation, as shown in Figure 1.

3.2. Edge Detector. There are various strategies to extract semantic edges from the results of semantic segmentation. An edge detection operator such as Canny [58], a spatial gradient solution [11], and a transfer network are optional solutions. To guarantee end-to-end training, the chosen scheme must be differentiable. For simplicity, we choose the spatial gradient solution, which uses adaptive pooling to derive a spatial gradient. The formulation is as follows:

$$\nabla S_k(p) = \left|S_k(p) - \text{pool}_w\left(S_k(p)\right)\right|, \tag{1}$$

where S represents the probability map of semantic segmentation, $S_k(p)$ indicates the predicted probability on the k-th semantic category at pixel p, and $|\cdot|$ remarks the absolute value function. pool_w is an adaptive average pooling operation with kernel size w. The same as in the study in [11], w is used to control the derived boundary width and is set to 3.

3.3. Task-Specific Elementary Consistency Unit. TriangleNet consists of three domains: the input image domain, the semantic segmentation domain, and the semantic edge domain. As illustrated in Figure 1, χ denotes the query domain (e.g., input RGB images) and $\psi = \{\psi_1, \psi_2\}$ is the set of two desired prediction domains. Specifically, ψ_1 represents the semantic segmentation domain and ψ_2 represents the semantic edge domain. The functions that map the query domain onto prediction domains are defined as $\Gamma_{\chi\psi_i}(i = 1, 2)$ which outputs ψ_i given χ. $\Gamma_{\psi_1\psi_2}$ denotes the cross-task function that maps the semantic segmentation domain to the semantic edge domain. According to the elementary consistency unit theory proposed by the authors in [13], predicting ψ_2 by first predicting ψ_1 from χ should achieve predictions similar to directly predicting ψ_2 from χ.

To enhance the comprehension of the consistent constraint across the three domains, as shown in Figure 1, we provided visual examples for each domain. S represents an instance from domain ψ_1, while E and C are the instances from domain ψ_2. Here, E corresponds to the output of the semantic edge detection model, while C is the output of the semantic segmentation model after undergoing the edge detector process. Notably, E and C exhibit a high degree of similarity, making them highly comparable and indicative of strong consistency between the two outputs.

3.3.1. Ohem Cross-Entropy Loss. In our framework, $\Gamma_{\chi\psi_i}$ are the neural networks. Through $\Gamma_{\chi\psi_1}$, we can obtain the semantic segmentation probability map $S \in R^{H\times W\times K}$, where K is the number of categories. A common way of training the neural network in $\Gamma_{\chi\psi_1}$ is to find parameters of $\Gamma_{\chi\psi_1}$ that minimize a loss called cross-entropy loss.

What we actually use is an improved version called the Ohem cross-entropy loss implemented by PaddleSeg [59]. It stands for "online hard example mining cross-entropy loss." Instead of considering the loss for all examples in a batch, it selects only the hard examples and uses those examples to update the model during training. This helps in dealing with class imbalance and emphasizing difficult examples that can lead to better generalization. Hard examples are considered with those examples with low probabilities of the relevant label. In other words, they are examples that the model finds challenging to classify correctly. We denote Ohem cross-entropy loss as L_s, which measures the difference between S and the semantic segmentation ground truth.

The formula for Ohem cross-entropy loss can be expressed as follows:

$$L_s = -\frac{1}{N_{\text{hard}}} \sum_{p\in \text{hard}_{\text{examples}}} Y(p)\log P(p), \tag{2}$$

where $Y(p)$ denotes the ground truth label at pixel p. $P(p)$ represents the probability of the corresponding label at pixel p. N_{hard} is the number of hard examples to be considered. It can be a fixed number or a percentage of the batch size, depending on implementation. In PaddleSeg [59], the number of hard examples is determined by the min_kept and thresh hyperparameters, where min_kept specifies the minimum number of hard examples to be kept and thresh sets the probability threshold below which examples are considered hard.

3.3.2. Semantic Edge Loss. Through $\Gamma_{\chi\psi_2}$, we can obtain the semantic edge probability map $E \in R^{H\times W\times K}$. While training the neural network in $\Gamma_{\chi\psi_2}$, we minimize a loss called multilabel loss, which is formulated as follows:

$$L_e = -\sum_k \sum_p G_k(p)\log E_k(p) + \left(1 - G_k(p)\right)\log\left(1 - E_k(p)\right), \tag{3}$$

where $G_k(p)$ denotes the ground truth edge label on the k-th semantic category at pixel p, and $E_k(p)$ indicates the predicted edge probability on the k-th semantic category at pixel p. L_e measures the difference between E and the semantic edge ground truth G.

3.3.3. Decomposed Cross-Task Consistency Loss. $\Gamma_{\psi_1\psi_2}$ is modeled as a spatial gradient operation formulated as equation (1). Taking S as the input of $\Gamma_{\psi_1\psi_2}$, another semantic edge probability map $C \in R^{H\times W\times K}$ can be obtained. C and E should be consistent. Instead of directly penalizing the difference between C and E, we penalize the difference between C and G, and E and G separately, thus indirectly forcing the alignment between C and E. The formulation is as follows:

$$L_c^d = \sum_k \sum_p W_k(p)\left(\left|C_k(p) - G_k(p)\right| + \left|E_k(p) - G_k(p)\right|\right). \tag{4}$$

We call L_c^d the decomposed cross-task consistency loss, in which

$$W_k(p) = \begin{cases} \beta^k, & G_k(p) = 1, \\ 1 - \beta^k, & G_k(p) = 0, \end{cases} \tag{5}$$

where $\beta^k = |Y_-^k|/|Y^k|$ and $1 - \beta^k = |Y_+^k|/|Y^k|$. $|Y_+^k|$ and $|Y_-^k|$ denote the edge and nonedge ground truth label sets of the k-th class semantic edge, respectively. Similar to $E_k(p)$, $C_k(p)$ denotes another predicted edge probability on the k-th semantic category at pixel p.

The right-hand side of equation (4) satisfies the following equation:

$$\begin{aligned} &\sum_k \sum_p W_k(p)\left(\left|C_k(p) - G_k(p)\right| + \left|E_k(p) - G_k(p)\right|\right) \\ &= \sum_k \sum_p W_k(p)\left|C_k(p) - G_k(p)\right| + \sum_k \sum_p W_k(p)\left|E_k(p) - G_k(p)\right|. \end{aligned} \tag{6}$$

For simplicity, we define the following equations.

$$L_{c1} = \sum_k \sum_p W_k(p)\left|C_k(p) - G_k(p)\right|, \tag{7}$$

$$L_{c2} = \sum_k \sum_p W_k(p)\left|E_k(p) - G_k(p)\right|. \tag{8}$$

Equations (7) and (8) are variants of the l_1 norm, and we call this kind of variant the boundary-aware l_1 norm. Substituting equations (7), (8), and (6) into (4), we derive

$$L_c^d = L_{c1} + L_{c2}. \tag{9}$$

3.3.4. Loss Function. We perform a weighted sum of the abovementioned three losses to obtain the loss to predict domain ψ_1 from χ while enforcing the consistency with domain ψ_2 as follows:

$$L = C_s L_s + C_e L_e + C_c L_c^d, \tag{10}$$

in which C_s, C_e, and C_c are the hyperparameters. As pointed out by the authors in [15], grid search is competitive or better compared to existing task balancing techniques in determining the weights of the loss functions. Therefore, in our experiments, C_s, C_e, and C_c are obtained by grid search.

First, we generate grids representing various coefficient values that we wish to explore and search over during our experiments. The loss function L_s primarily computes the loss for the majority of pixels in the image, leading to relatively larger loss values compared to the other two loss functions, which are specifically designed for focusing on object edges. However, we aim to prevent these two losses from being overshadowed due to their smaller values. To achieve this, we assign larger coefficients to the edge-related loss functions, prompting the model to pay closer attention to the edges during training. Specifically, we set $C_s \in \{1\}$ and $C_e, C_c \in \{5, 10, 20\}$. Then, we try all combinations of the hyperparameter values from the defined grids. Since we have only one value for C_s and three values for both C_e and C_c, we have a total of $1 \times 3 \times 3 = 9$ combinations to try.

Substituting equation (9) into (10), we obtain

$$L = C_s L_s + C_e L_e + C_c\left(L_{c1} + L_{c2}\right), \tag{11}$$

which is equivalent to the following equation:

$$L = \left(C_s L_s + C_c L_{c1}\right) + \left(C_e L_e + C_c L_{c2}\right), \tag{12}$$

where the first term is pertinent to the network $\Gamma_{\chi\psi_1}$, while the second term is pertinent to the network $\Gamma_{\chi\psi_2}$. These two terms are independent and can be dealt with in parallel for task-specific layers in networks $\Gamma_{\chi\psi_1}$ and $\Gamma_{\chi\psi_2}$, which is exactly the original intention of our definition of L_c^d as two independent parts.

4. Experiments

We first conducted experiments on Cityscapes [60] which is a popular computer vision dataset for semantic urban scene understanding. It contains 5,000 annotated images with fine annotations collected from 50 cities in different seasons. The images were divided into sets numbered 2,975, 500, and 1,525 for training, validation, and testing. Conventionally, only 19 categories are used to assess the accuracy of category segmentation. Although it also provides coarsely annotated images, we only use finely annotated images. In addition, experiments on the FloodNet [61] were also performed to further confirm the generalization and application values of our method. Code and models are available at https://github.com/nailperry-zd/PaddleSeg-TriangleNet.

4.1. Experiments on Cityscapes

4.1.1. Baseline. We append the PPM [21] and FPN [57] as the decoder to naive FCN as the Baseline, where ResNet-18 [62] serves as the backbone, that is, training the semantic segmentation branch shown in Figure 2 independently.

4.1.2. Implementation Details. We use the 2.3.0 version of the PaddlePaddle [63] framework to carry out the following experiments. The hardware platform adopts a single V100 GPU with a video memory of 32G. All networks with ResNet-18 [62] as the backbone share some settings, where stochastic gradient descent (SGD) with a batch size of 4 is used as the optimizer, with a momentum of 0.9 and weight decay of $5e-4$. All these ResNet-18 variants are trained for 300K batch iterations with an initial learning rate of 0.01. Data augmentation contains normalization, random distortion, random horizontal flip, random resizing with a scale range of [0.5, 2.0], and random cropping with a crop size of 1024×1024. During inference, we use the whole picture as input. In terms of loss weights, C_s, C_e, and C_c are set to 1, 10, and 20, respectively. For quantitative evaluation, mIoU is used for accuracy comparison.

4.1.3. Comparison against State-of-the-Art Methods. We present a comprehensive comparison of our method with both real-time and non-real-time semantic segmentation algorithms in Tables 1 and 2, respectively.

In Table 1, it is important to highlight that some of the models listed have been officially integrated into PaddlePaddle and are available in the PaddleSeg [59] open source library. This integration facilitates a rigorous evaluation of the inference speed for these PaddlePaddle-integrated models, as well as our model, TriangleNet. To measure the speed accurately, we utilize the PaddleInference API from the PaddleSeg [59] library on an A100 GPU with 40GB memory, using the f32 accuracy parameter. However, in cases where certain models do not have a PaddlePaddle implementation or when our direct measurements are not available, we provide FPS data from the original papers or third-party sources within brackets in the table for reference. This meticulous approach ensures a comprehensive and fair comparison, allowing us to draw reliable conclusions regarding TriangleNet's performance in real-time semantic segmentation in comparison to other state-of-the-art models.

For the real-time comparison, we ensure a fair assessment by utilizing our best model based on ResNet-18 with varying inference sizes: 512×1024, 768×1536, and 1024×2048, represented by TriangleNet^1-0.5, TriangleNet^1-0.75, and TriangleNet^1-1.0, respectively. For the non-real--time comparison, we adopt multiscale inference, denoted by TriangleNet^1-MS, incorporating scales of 0.75, 1.0, and 1.25.

As shown in Table 1, at a resolution of 512×1024, our model not only surpasses ESPNetV2 in both speed and accuracy but also outperforms STDC1-Seg50 and PP-LiteSeg-T1 in accuracy, despite achieving approximately 60% and 70% of their respective speeds. Similarly, at 768×1536 resolution, while our model maintains 85% of BiSeNetV1-L's speed, it exhibits a 1.4% increase in accuracy compared to it. In addition, our model's speed, at around 50% of STDC1-Seg75 and PP-LiteSeg-T2, is compensated by approximately 1% higher accuracy over them. Under a resolution of 1024×2048, our model exhibits significantly higher accuracy than ICNet, SwiftNet, and FasterSeg.

Our method demonstrates a remarkable speed/accuracy trade-off across various resolutions when compared to real-time counterparts. Notably, our model achieves impressive accuracy without compromising on speed, enabling real-time inference even at full resolution. Notably, in Table 2, our model demonstrates competitive performance even compared to non-real-time models based on ResNet-101 [62], achieving similar mIoU scores while utilizing only one-fifth of the parameters. This highlights the efficiency and effectiveness of our approach across diverse scenarios.

4.1.4. Ablation Studies. Our approach involves several elements compared to the Baseline. Each element may contribute to the improvement of mIoU. To verify the necessity of each element, we performed the following ablation studies.

(1) Ablation Study on Joint Framework. From a multitask perspective, joint training benefits from higher task correlation. To explore this idea, we conducted experiments combining different tasks. In Table 3, the second row demonstrates joint training of Baseline with HED [54], a classic binary edge detection model, resulting in a slight improvement in mIoU on the Cityscapes test set. Subsequently, we replaced HED with DFF [56], a superior semantic edge detection model, in the third row. This change led to a 0.59% improvement against the Baseline in mIoU on the Cityscapes test set. The results suggest a stronger correlation between semantic segmentation and semantic edge detection tasks. This correlation arises from the accurate extraction of semantic edges under the constraints of semantic segmentation, as semantic segmentation can suppress nonedge pixels, and in turn, relies on semantic edges to distinguish between objects and background. The two tasks mutually complement each other, enhancing the overall performance of the model.

During this process, we adopted the poly learning rate policy, which is widely used and proven effective, as depicted in Figure 4(a).

(2) Ablation Study on L_c^d. We introduced the L_c^d during the last 50% iterations to verify its effect. This idea draws on the study in [74], where a loss called ABL is added at the last 20% epochs, since the gradient of ABL is not useful when the semantic edges output by the network are far from the semantic edge ground truth at the beginning of the training, much similar to our case.

During this process, we employed a custom learning rate policy named "2-cycle-SGDR poly," which can be seen as a variant of the cosine annealing policy [75]. The visualizations of the cosine annealing and 2-cycle-SGDR poly policies are shown in Figures 4(b) and 4(c), respectively. In the 2-cycle-SGDR poly policy, the learning rate periodically increases. This phenomenon of decreasing the rate to the minimum and then increasing the rate is called "restarts" in SGDR [75]. The underlying idea is to encourage the model to

Table 1: Accuracy comparison of our best models based on ResNet-18 against real-time models on the Cityscapes test set.

Model	Backbone	mIoU	FPS		
			1024 × 2048	768 × 1536	512 × 1024
ICNet [66]	PSPNet50	69.5	−(30.3)		
ESPNetV2 [67]	—	66.2			126.5 (114.7+)
SwiftNet [68]	ResNet18	75.5	−(39.9)		
BiSeNetV1 [69]	Xception39	68.4		—(105.8)	
BiSeNetV1-L [69]	ResNet18	74.7		83.9 (65.5)	
BiSeNetV2 [70]	—	72.6			−(156)
BiSeNetV2-L [70]	—	75.3			−(47.3)
FasterSeg [71]	—	71.5	−(163.9)		
STDC1-Seg50 [72]	STDC1	71.9			262.1 (250.4)
STDC2-Seg50 [72]	STDC2	73.4			207.4 (188.6)
STDC1-Seg75 [72]	STDC1	75.3		152.7 (126.7)	
STDC2-Seg75 [72]	STDC2	76.8		131.5 (97.0)	
PP-LiteSeg-T1 [73]	STDC1	72.0			219.4 (273.6)
PP-LiteSeg-B1 [73]	STDC2	73.9			184.3 (195.3)
PP-LiteSeg-T2 [73]	STDC1	74.9		141.2 (143.6)	
PP-LiteSeg-B2 [73]	STDC2	77.5		118.4 (102.6)	
TriangleNet^{1}-0.5	ResNet18	72.7			157.4
TriangleNet^{1}-0.75	ResNet18	76.1		71.0	
TriangleNet^{1}-1.0	ResNet18	77.4	46.2		

"–" indicates that the corresponding data are not given. FPS, frames per second. TriangleNet^{1} is an instance of the framework shown in Figure 2. In the three columns of FPS, the values outside the brackets are measured by our team, whereas the values within the brackets are either sourced from the original papers or from third-party papers. "+" denotes the value is sourced from [64].

Table 2: Accuracy comparison of our best model based on ResNet-18 against non-real-time models on the Cityscapes test set.

Model	Backbone	Params (M)	mIoU (%)
DeepLab [5]	ResNet-101 [62]	59	63.1
DepthSeg [65]	ResNet-101	58	78.2
PSPNet [21]	ResNet-101	65	78.4
TriangleNet^{1}-MS	ResNet-18	13	77.8

Table 3: Ablation study on joint framework.

Model	LR policy	mIoU (%)	
		Val	Test
Baseline	Poly	76.77	74.48
Baseline + HED [54]	Poly	77.33	74.66
Baseline + DFF [56]	Poly	**78.13**	**75.07**

The bold values indicate the maximum value in each column. In this context, the bold values specifically indicate that the model in the third row demonstrates superior performance on both the validation and test sets.

traverse from one local minimum to another, particularly if it is trapped in a steep trough.

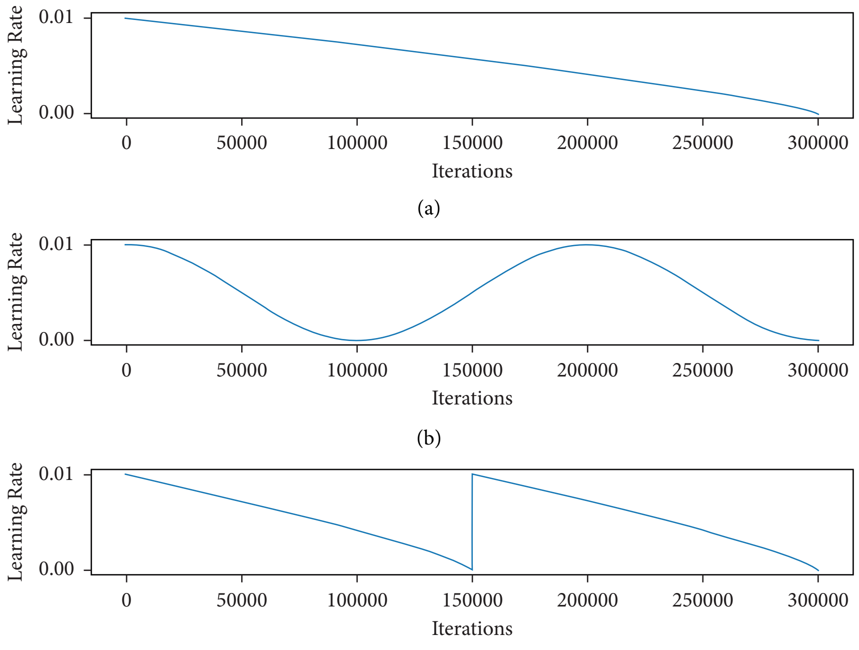
After comparing the first and second rows in Table 4, we observed that employing the 2-cycle-SGDR poly policy alone resulted in a 0.46% increase in mIoU on the Cityscapes test set. Subsequently, with the introduction of L_c^d in the third row, the mIoU exhibited a consistent growth trend on both the validation and test sets, with a more significant improvement observed on the test set. The inclusion of L_c^d further boosted the mIoU by 1.83% on the test set, indicating its positive impact on the overall performance of our model.

Furthermore, upon comparing the third and fourth rows in Table 4, we found that both cosine annealing and 2-cycle-SGDR poly policies can improve the model to some extent, confirming the effectiveness of the "restarts" in SGDR. Notably, 2-cycle-SGDR poly is more suitable for our model. Therefore, for all subsequent TriangleNet variants, we adopted the 2-cycle-SGDR poly learning rate schedule to ensure consistent and superior optimization of our model.

(3) Ablation Study on Different Semantic Edge Detection Strategies. In our exploration of state-of-the-art strategies for semantic edge detection, we conducted a comparison between DFF [56] and CASENet [55]. The two rows in Table 5 demonstrate that both DFF and CASENet, when employed as semantic edge detection branches, lead to improved segmentation accuracy. This finding highlights the positive role of injecting semantic edges into the semantic segmentation process. Notably, as mentioned in the study in [56], DFF surpasses CASENet in standalone semantic edge extraction. Moreover, even after integrating with semantic segmentation, DFF continues to deliver superior accuracy improvements, reaffirming its effectiveness in enhancing the overall performance of the model.

(4) Ablation Study on Different Semantic Segmentation Models. Our evaluation extends to various semantic segmentation models, and the results presented in Table 6 reveal that the integration of U-Net [2] and Baseline into our framework for joint training yields remarkable improvements in semantic segmentation accuracy, showcasing the effectiveness and benefits of our approach in enhancing semantic segmentation performance.

4.1.5. Analyses. The results from the ablation studies demonstrate that the semantic edge detection task exhibits a stronger correlation with the semantic



(a)

(b)

(c)

Figure 4: Visualization of different learning rate policies: (a) poly, (b) cosine annealing, and (c) 2-cycle SGDR poly.

Table 4: Ablation on L_c^d. All these models are trained for 300K iterations.

Model	LR policy	L_c^d involved	mIoU (%)	
			Val	Test
Baseline + DFF	Poly	×	78.13	75.07
Baseline + DFF	2-cycl-SGDR poly	×	78.41 (0.28 ↑)	75.53 (0.46 ↑)
TriangleNet[1]	2-cycl-SGDR poly	√	**78.96 (0.55↑)**	**77.36 (1.83↑)**
TriangleNet[1]	Cosine annealing	√	78.65	76.93

"×" means that L_c^d is not involved in all iterations. TriangleNet[1] is an instance of the framework shown in Figure 2. In this situation, we get all the results by single-scale inference. The bold values indicate the maximum value in each column. In this context, the bold values specifically indicate that the model corresponding to the configuration detailed in the third row demonstrates superior performance on both the validation and test sets.

Table 5: Experiments on different semantic edge detection (SED) strategies.

Model	SED strategy	mIoU (%)	
		Val	Test
TriangleNet[1]	DFF [56]	**78.96**	**77.36**
TriangleNet[2]	CASENet [55]	78.87	77.11

TriangleNet[2] is the same as TriangleNet[1] except that TriangleNet[2] uses CASENet [55] as the semantic edge detection. In this situation, we get all the results by single-scale inference. The bold values indicate the maximum value in each column. In this context, the bold values specifically indicate that the model corresponding to the configuration detailed in the first row demonstrates superior performance on both the validation and test sets.

Table 6: Experiments on different semantic segmentation (SEM) strategies.

Model	mIoU (%) on val
U-Net [2]	66.34
TriangleNet[3] (SEM: U-Net)	69.12 (2.78 ↑)
Baseline	76.77
TriangleNet[1] (SEM: Baseline)	78.96 (2.19 ↑)

TriangleNet[3] is the same as TriangleNet[1] except that TriangleNet[3] uses U-Net [2] as the semantic segmentation branch.

segmentation task compared to the binary edge detection task. When we jointly train both tasks, we observe a 0.59% improvement in mIoU on the Cityscapes test set. In addition, the adoption of the 2-cycle-SGDR poly learning rate policy leads to a slight yet meaningful improvement of 0.46%.

Table 7: Intersection over Union (IoU) growth per category on the Cityscapes test set.

Model	Road	Sidewalk	Building	Wall	Fence	Pole	Trafficlight	Trafficsign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mIoU
Baseline	98.05	82.70	92.99	51.81	55.34	60.97	70.13	73.64	92.93	72.29	95.37	83.09	64.19	94.84	61.62	70.45	62.96	60.81	72.03	74.48
Baseline + DFF	98.32	84.39	92.17	48.92	57.15	62.34	71.34	75.48	93.12	73.10	95.49	83.41	63.85	95.16	63.11	69.83	63.91	62.40	72.80	75.07
TriangleNet[1]	98.37	84.87	92.68	53.22	58.56	65.01	72.79	76.63	93.22	72.77	95.59	84.55	66.26	95.37	64.63	76.59	80.91	64.16	73.63	77.36
	0.32	2.17	0.69	1.41	**3.23**	**4.04**	2.66	**2.99**	0.29	0.48	0.22	1.46	2.07	0.54	**3.01**	**6.15**	**17.94**	**3.35**	1.60	2.87

Bold values denote categories that have gained significant improvements.

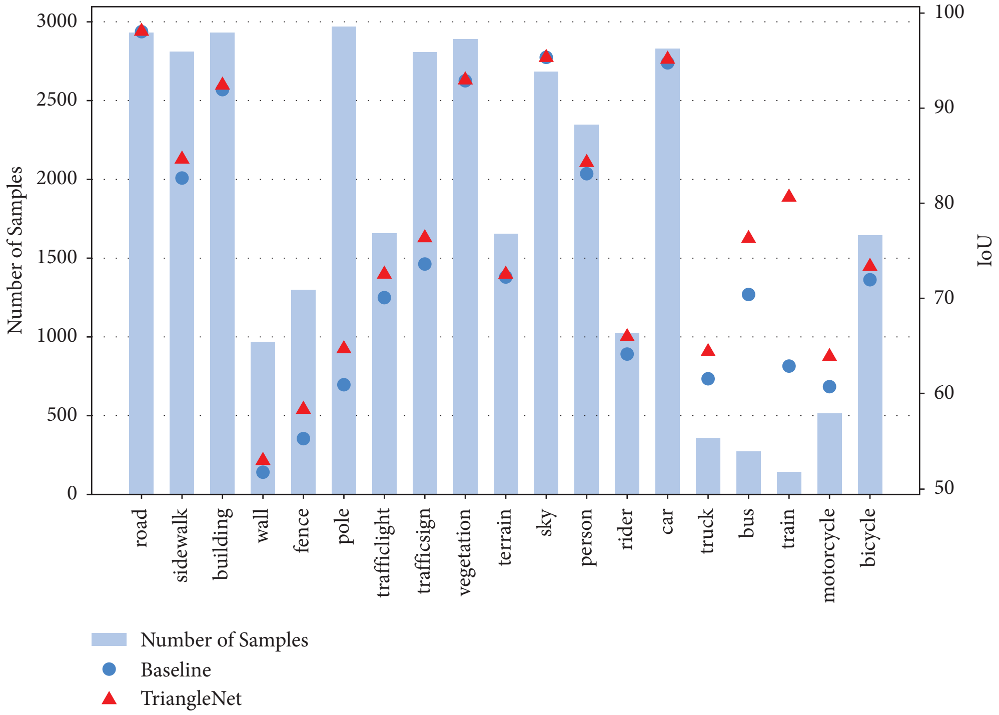


Figure 5: Category-level analysis of samples and Intersection over Union (IoU) on Cityscapes test set.

However, despite the benefits of multitask learning, the implicit learning of cross-task consistency in multitask networks is limited. To address this limitation, we introduced an additional restriction called decoupled cross-task consistency loss, denoted as L_c^d, to explicitly enhance cross-task consistency between semantic edge detection and semantic segmentation.

The explicit enhancement of consistency through this restriction resulted in a further significant improvement of 1.83% in mIoU on the Cityscapes test set. These findings underscore the importance of incorporating explicit consistency constraints during the learning process to achieve enhanced performance in multitask computer vision systems.

Upon analysing the IoU of each category, as depicted in Table 7, we observe significant improvements for several categories, with some experiencing increases of more than 3%. Notably, the largest improvement amounts to nearly 18% of the IoU score. This further validates the significance of incorporating edge information through our edge prior augmentation approach, particularly for categories such as "truck," "bus," and "train," where distinct edges and boundaries are prevalent.

Furthermore, we investigate the relationship between the number of sample images per category and their respective IoU scores, as visualized in Figure 5. The analysis reveals that categories with higher IoU scores tend to have more samples, while those with lower IoU scores have fewer samples, aligning with our expectations. Notably, categories such as "truck," "bus," and "train," despite having smaller sample sizes, exhibit remarkable improvements in IoU. This suggests that TriangleNet can effectively generalize from limited samples, resulting in enhanced performance in challenging categories.

However, certain categories, such as "wall," "fence," "pole," and "trafficsign," possess abundant samples but fail to achieve the expected higher IoU values. The underperformance of these categories may be attributed to factors such as complex semantic patterns or limitations in the model architecture to accurately capture their unique features. Further investigation is warranted to identify the specific reasons behind these discrepancies and to devise strategies to enhance the segmentation performance for these categories.

To summarize, TriangleNet demonstrates a remarkable 2.88% improvement in mean Intersection over Union (mIoU) on the Cityscapes test set compared to the Baseline. Our model outperforms the Baseline in all categories, particularly in scenarios with distinct edges and boundaries, substantiating the efficacy of multitask learning and explicit cross-task consistency enhancement.

4.1.6. Visualization. We performed a qualitative comparison by visualizing segmentation maps of various categories. In Figure 6, we present the segmentation maps generated by different models for this purpose. Notably, in the second row, TriangleNet demonstrates its ability to accurately locate pixels along two objects by effectively utilizing edge priors or shapes of objects, which the Baseline fails to achieve. In addition, in the first, third, and fourth rows, TriangleNet leverages the priors obtained from semantic edge detection to perceive trains, buses, and walls as coherent entities, while the Baseline tends to split

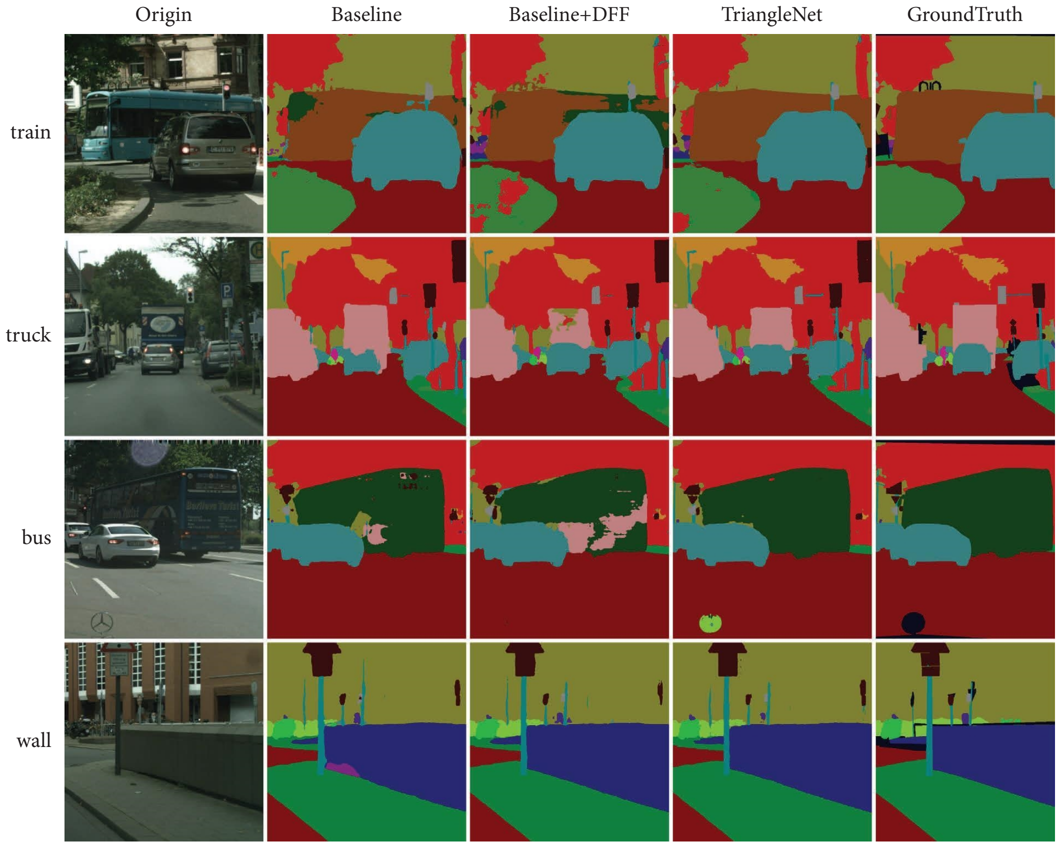


Figure 6: Visualization on Cityscapes.

Table 8: Experiments on the FloodNet test set.

Index	Class	Images
0	Background	—
1	Building-flooded	275
2	Building-non-flooded	1,272
3	Road-flooded	335
4	Road-non-flooded	1,725
5	Water	1,262
6	Tree	2,507
7	Vehicle	1,105
8	Pool	676
9	Grass	—

them into different categories. This highlights TriangleNet's capacity to benefit from edge information and enhance its understanding of complex object structures, resulting in improved semantic segmentation performance.

4.2. Experiments on FloodNet. We also performed experiments on FloodNet [61] which is an unmanned aerial vehicle (UAV) dataset to assess the damage from natural disasters to further prove the compatibility of our method. The dataset contains 2,343 images in total, which were divided into sets numbered 1,445, 450, and 448 for training, validation, and testing. This dataset contains 10 classes, and the index and specific meaning of each class are given by Table 8.

4.2.1. Implementation Details. We verified the performance of the Baseline and TriangleNet aforementioned on the FloodNet dataset. Different from experiments on Cityscapes, some hyperparameters need to be changed. Specifically, the batch size of SGD is set to 16, and all these ResNet-18 [62] variants are trained for 20K iterations with 4 V100 GPUs. In terms of loss weights, C_s, C_e, and C_c are set to 1, 2, and 4, respectively.

4.2.2. Analyses. We analysed prominently improved categories on FloodNet. As shown in Table 9, TriangleNet still achieves better performance on all categories against the Baseline. It is worth noting that although DeepLabV3+ [76]

Table 9: Experiments on the FloodNet test dataset. Per-class results on the FloodNet test set.

Method	Backbone	1	2	3	4	5	6	7	8	9	mIoU
DeepLabV3+ [76]	ResNet-101	32.7	72.8	52	70.2	**75.2**	77.0	42.5	47.1	84.3	61.53
Baseline	ResNet-18	72.58	73.86	53.68	80.05	69.36	79.06	57.8	57.06	87.36	65.64
TriangleNet[1]	ResNet-18	**78.26**	**75.38**	**56.44**	**82.45**	74.57	**82.86**	**61.33**	**61.73**	**89.57**	**70.97**

Note: The results of the last two rows are obtained by single-scale inference. Bold values indicate the highest value in each column. In this context, the bold values specifically indicate that the model in the third row demonstrates superior performance on most of the classes.

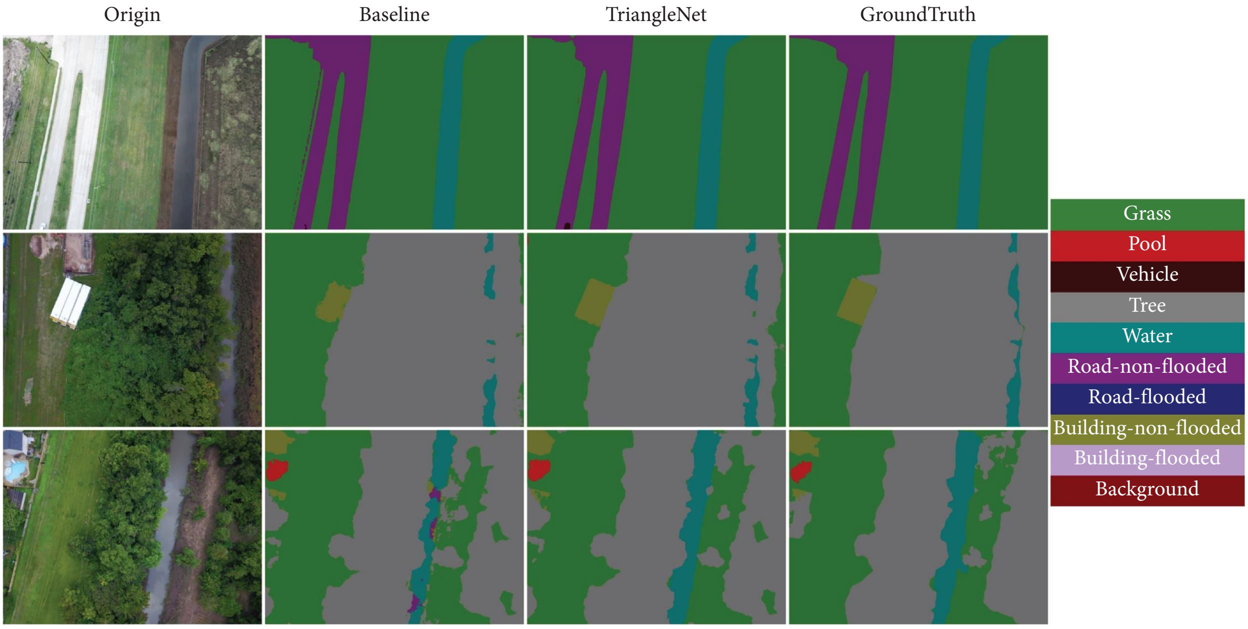


Figure 7: Visualization on FloodNet.

is based on ResNet-101 [62], its accuracy on FloodNet is not as good as our model based on ResNet-18 [62].

4.2.3. Visualization. The improvement can obviously be visualized in Figure 7. In the first row, TriangleNet is able to segment the sharp edge of the non-flooded road, while Baseline fails. In the second row, the non-flooded building is well segmented in TriangleNet. In the third row, the water is perceived as a whole, while Baseline divides it into parts.

5. Conclusion

In this paper, we presented TriangleNet, an innovative model that utilizes a decoupled architecture for joint training of multiple tasks without the need for fusion modules during inference. This design allows our model to reap the benefits of multitasking during training while avoiding any additional inference time, making TriangleNet a practical and efficient solution for real-time semantic segmentation. By employing multitask learning and explicit cross-task consistency enhancement, TriangleNet consistently achieves improvements on both the Cityscapes and FloodNet datasets, showcasing its robust generalization capabilities in various environmental conditions.

In summary, TriangleNet showcases its potential in semantic segmentation by effectively leveraging edge priors and incorporating explicit cross-task consistency. This unique combination not only enhances accuracy but also enables real-time inference, making it well-suited for various real-world applications. Further research exploring more detailed explicit constraints may lead to even greater performance improvements. The achievements of TriangleNet in the context of real-time semantic segmentation pave the way for future advancements in efficient and accurate computer vision systems.

Data Availability

The code and well-trained models used to support the findings of this study are available at https://github.com/nailperry-zd/PaddleSeg-TriangleNet.

Disclosure

A preprint has previously been published [77].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE at Minzu University of China.

References

[1] J. Long, E. Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, May 2015.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, Singapore, September, 2015.

[3] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "Fastfcn: rethinking dilated convolution in the backbone for semantic segmentation," 2019, https://arxiv.org/abs/1903.11816.

[4] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 5229–5238, Seoul, Korea (South), June 2019.

[5] L.-C. Chen, P. George, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," 2014, https://arxiv.org/abs/1606.00915.

[6] 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," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[7] L.-C. Chen, P. George, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, https://arxiv.org/abs/1706.05587.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 2961–2969, Venice, Italy, October, 2017.

[9] I. Ulku and E. Akagündüz, "A survey on deep learning-based architectures for semantic segmentation on 2d images," *Applied Artificial Intelligence*, vol. 36, pp. 1–45, 2022.

[10] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, https://arxiv.org/abs/1706.05098.

[11] M. Zhen, J. Wang, L. Zhou et al., "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13666–13675, Seattle, WA, USA, June, 2020.

[12] W. Liu, Z. Lu, and X. He, "Auxiliary edge detection for semantic image segmentation," in *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*, pp. 182–187, Tianjin China, April, 2020.

[13] A. R. Zamir, S. Alexander, N. Cheerla et al., "Robust learning through cross-task consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11197–11206, Seattle, WA, USA, June, 2020.

[14] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020.

[15] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, 2021.

[16] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2022.

[17] Y. Fisher, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 472–480, Honolulu, HI, USA, July, 2017.

[18] K. Sun, B. Xiao, L. Dong, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5693–5703, Long Beach, CA, USA, June, 2019.

[19] P. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proceedings of the International conference on machine learning*, pp. 82–90, PMLR, Lanzhou, China, July 2014.

[20] W. Byeon, T. M. Breuel, F. Raue, and L. Marcus, "Scene labeling with lstm recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3547–3555, Boston, MA, USA, June 2015.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890, Honolulu, HI, USA, July, 2017.

[22] L. Zhang, X. Li, A. Arnab, K. Yang, Y. Tong, and P. H. S. Torr, "Dual graph convolutional network for semantic segmentation," 2019, https://arxiv.org/abs/1909.06121.

[23] S. Zheng, S. Jayasumana, B. Romera-Paredes et al., "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 1529–1537, Santiago, Chile, June 2015.

[24] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, Salt Lake City, UT, USA, May 2018.

[25] J. Fu, J. Liu, H. Tian et al., "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3146–3154, Long Beach, CA, USA, June 2019.

[26] J. Cheng, X. Peng, X. Tang, W. Tu, and W. Xu, "Mifnet: a lightweight multiscale information fusion network," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5617–5642, 2022.

[27] Z. Zhang, Z. Cui, C. Xu, J. Zequn, L. Xiang, and J. Yang, "Joint task-recursive learning for semantic segmentation and depth estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 235–251, Munich, Germany, September 2018.

[28] A. Mousavian, H. Pirsiavash, and K. Jana, "Joint semantic segmentation and depth estimation with deep convolutional networks," in *Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV)*, pp. 611–619, IEEE, Stanford, CA, USA, October, 2016.

[29] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, "Real-time joint semantic segmentation and depth estimation using asymmetric annotations," in *Proceedings of the 2019 International Conference on Robotics and*

Automation (ICRA), pp. 7101–7107, IEEE, Stanford, CA, USA, October, 2019.

[30] L. He, J. Lu, G. Wang, S. Song, and J. Zhou, "Sosd-net: joint semantic object segmentation and depth estimation from monocular images," *Neurocomputing*, vol. 440, pp. 251–263, 2021.

[31] D. De Geus, P. Meletis, and G. Dubbelman, "Panoptic segmentation with a joint semantic and instance segmentation network," 2018, https://arxiv.org/abs/1809.02110.

[32] L. Zhao and W. Tao, "Jsnet: joint instance and semantic segmentation of 3d point clouds," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12951–12958, 2020.

[33] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 6819–6829, Seoul, Korea (South), June 2019.

[34] L.-C. Chen, J. T. Barron, P. George, K. Murphy, and A. L. Yuille, "Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4545–4554, Las Vegas, NV, USA, June 2016.

[35] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7482–7491, Salt Lake City, UT, USA, May 2018.

[36] I. Kokkinos, "Ubernet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6129–6138, Honolulu, HI, USA, July 2017.

[37] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "Pad-net: multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 675–684, San Juan, PR, USA, June 2018.

[38] Trevor Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *Proceedings of the International Conference on Machine Learning*, pp. 9120–9132, PMLR, Corvalis OR USA, June 2020.

[39] A. Nakano, S. Chen, and K. Demachi, "Cross-task consistency learning framework for multi-task learning," 2021, https://arxiv.org/abs/2111.14122.

[40] S. Zhu, G. Brazil, and X. Liu, "The edge of depth: explicit constraints between segmentation and depth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13116–13125, Seattle, WA, USA, June 2020.

[41] S. Papadopoulos, I. Mademlis, and I. Pitas, "Semantic image segmentation guided by scene geometry," in *Proceedings of the 2021 IEEE International Conference on Autonomous Systems (ICAS)*, pp. 1–5, IEEE, Québec, Canada, August, 2021.

[42] P. Adam, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: a deep neural network architecture for real-time semantic segmentation," 2016, https://arxiv.org/abs/1606.02147.

[43] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the european conference on computer vision (ECCV)*, pp. 552–568, Glasgow, UK, September, 2018.

[44] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 6848–6856, Salt Lake City, UT, USA, June 2018.

[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.

[46] G. Hinton, Oriol Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, https://arxiv.org/abs/1503.02531.

[47] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2604–2613, Seattle, WA, USA, June 2019.

[48] H. Tong, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 578–587, Long Beach, CA, USA, August 2019.

[49] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 2736–2744, Venice, Italy, July 2017.

[50] B. Jacob, S. Kligys, B. Chen et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2704–2713, Salt Lake City, UT, USA, June 2018.

[51] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: training deep neural networks for wireless resource management," in *Proceedings of the 2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–6, IEEE, Sapporo, Japan, June 2017.

[52] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[53] P. Hu, F. Perazzi, F. C. Heilbron et al., "Real-time semantic segmentation with fast attention," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 263–270, 2021.

[54] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 1395–1403, Santiago, Chile, July 2015.

[55] Z. Yu, F. Chen, M.-Y. Liu, and S. Ramalingam, "Casenet: deep category-aware semantic edge detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5964–5973, Honolulu, HI, USA, June 2017.

[56] Y. Hu, Y. Chen, L. Xiang, and J. Feng, "Dynamic feature fusion for semantic edge detection," 2019, https://arxiv.org/abs/1902.09104.

[57] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, Honolulu, HI, USA, June 2017.

[58] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[59] Y. Liu, L. Chu, G. Chen et al., “Paddleseg: a high-efficient development toolkit for image segmentation,” 2021, https://arxiv.org/abs/2101.06175.

[60] M. Cordts, O. Mohamed, S. Ramos et al., “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, Las Vegas, NV, USA, June 2016.

[61] M. Rahnemoonfar, T. Chowdhury, A. Sarkar, D. Varshney, M. Yari, and R. R. Murphy, “Floodnet: a high resolution aerial imagery dataset for post flood scene understanding,” *IEEE Access*, vol. 9, pp. 89644–89654, 2021.

[62] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, July 2016.

[63] Y. Ma, D. Yu, T. Wu, and H. Wang, “Paddlepaddle: an open-source deep learning platform from industrial practice,” *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105–115, 2019.

[64] C. J. Holder and M. Shafique, “On Efficient Real-Time Semantic Segmentation: A Survey,” 2022, https://arxiv.org/abs/2206.08605.

[65] S. Kong and C. C. Fowlkes, “Recurrent scene parsing with perspective understanding in the loop,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 956–965, Salt Lake City, UT, USA, July 2018.

[66] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, “Icnet for real-time semantic segmentation on high-resolution images,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 405–420, Tel Aviv, Israel, August, 2018.

[67] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, “Espnetv2: a light-weight, power efficient, and general purpose convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9190–9200, Glasgow, UK, August, 2019.

[68] H. Wang, X. Jiang, H. Ren, Y. Hu, and B. Song, “Swiftnet: real-time video object segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1296–1305, Glasgow, UK, September, 2021.

[69] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Bisenet: bilateral segmentation network for real-time semantic segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341, Tel Aviv, Israel, August, 2018.

[70] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, “Bisenet v2: bilateral network with guided aggregation for real-time semantic segmentation,” *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3051–3068, 2021.

[71] W. Chen, X. Gong, X. Liu, Q. Zhang, L. Yuan, and Z. Wang, “Fasterseg: searching for faster real-time semantic segmentation,” 2019, https://arxiv.org/abs/1912.10917.

[72] M. Fan, S. Lai, J. Huang et al., “Rethinking bisenet for real-time semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9716–9725, Nashville, TN, USA, June, 2021.

[73] J. Peng, Y. Liu, S. Tang et al., “Pp-liteseg: a superior real-time semantic segmentation model,” 2022, https://arxiv.org/abs/2204.02681.

[74] C. Wang, Y. Zhang, M. Cui et al., “Active boundary loss for semantic segmentation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, pp. 2397–2405, 2022.

[75] I. Loshchilov and F. Hutter, “Sgdr: stochastic gradient descent with warm restarts,” 2016, https://arxiv.org/abs/1608.03983.

[76] L.-C. Chen, Y. Zhu, P. George, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, Glasgow, UK, August, 2018.

[77] D. Zhang and R. Zheng, “Trianglenet: edge prior augmented network for semantic segmentation through cross-task consistency,” 2022, https://arxiv.org/abs/2210.05152.