PIDNet: A Real-time Semantic Segmentation Network Inspired by PID Controllers

Jiacong Xu Zixiang Xiong Shankar P. Bhattacharyya
Dept of ECE, Texas A&M University, College Station, TX 77843
jxu155@jhu.edu, zx@ece.tamu.edu, spb@tamu.edu

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.01871

Abstract

Two-branch network architecture has shown its efficiency and effectiveness in real-time semantic segmentation tasks. However, direct fusion of high-resolution details and low-frequency context has the drawback of detailed features being easily overwhelmed by surrounding contextual information. This overshoot phenomenon limits the improvement of the segmentation accuracy of existing two-branch models. In this paper, we make a connection between Convolutional Neural Networks (CNN) and Proportional-Integral-Derivative (PID) controllers and reveal that a two-branch network is equivalent to a Proportional-Integral (PI) controller, which inherently suffers from similar overshoot issues. To alleviate this problem, we propose a novel three-branch network architecture: PIDNet, which contains three branches to parse detailed, context and boundary information, respectively, and employs boundary attention to guide the fusion of detailed and context branches. Our family of PIDNets achieve the best trade-off between inference speed and accuracy and their accuracy surpasses all the existing models with similar inference speed on the Cityscapes and CamVid datasets. Specifically, PIDNet-S achieves 78.6% mIOU with inference speed of 93.2 FPS on Cityscapes and 80.1% mIOU with speed of 153.7 FPS on CamVid.

1. Introduction

Proportional-Integral-Derivative (PID) Controller is a classic concept that has been widely applied in modern dynamic systems and processes such as robotic manipulation [3], chemical processes [24], and power systems [25]. Even though many advanced control strategies with better control performance have been developed in recent years, PID controller is still the go-to choice for most industry applications due to its simplicity and robustness. Furthermore, the idea of PID controller has been extended to many other areas. For example, researchers introduced the PID concept to image denoising [32], stochastic gradient decent [1] and numerical optimization [50] for better algorithm performance. In this paper, we devise a novel architecture for real-time semantic segmentation tasks by employing the basic concept of PID controller and demonstrate that the performance of our model surpasses all the previous works and achieves the best trade-off between inference speed and accuracy, as illustrated in Figure 1, by extensive experiments.

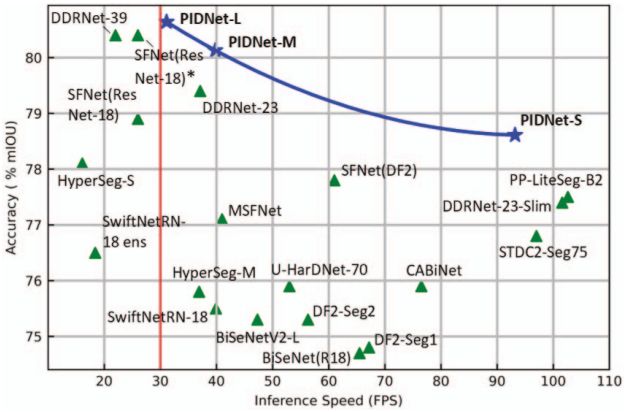

Figure 1. The trade-off between inference speed and accuracy (reported) for real-time models on the Cityscapes [12] test set. Blue stars refer to our models while green triangles represent others.

Semantic segmentation is a fundamental task for visual scene parsing with the objective of assigning each pixel in the input image to a specific class label. With the increasing demand of intelligence, semantic segmentation has become the basic perception component for applications such as autonomous driving [16], medical imaging diagnosis [2] and remote sensing imagery [54]. Starting from FCN [31], which achieved great improvement over traditional methods, deep convnets gradually dominated the semantic segmentation field and many representative models have been proposed [4, 6, 40, 48, 59, 60]. For better performance, various strategies were introduced to equip these models with the capability of learning contextual dependencies among

[1]Work supported in part by NSF grants ECCS-1923803 and CCF-2007527.

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.01871

pixels in large scale without missing important details. Even though these models achieve encouraging segmentation accuracy, too much computational cost are required, which significantly hinder their application in real-time scenarios, such as autonomous vehicle [16] and robot surgery [44].

To meet real-time or mobile requirements, researchers have come up with many efficient and effective models in the past for semantic segmentation. Specifically, ENet [36] adopted lightweight decoder and downsampled the feature maps in early stages. ICNet [58] encoded small-size inputs in complex and deep path to parse the high-level semantics. MobileNets [21, 42] replaced traditional convolutions with depth-wise separable convolutions. These early works reduced the latency and memory usage of segmentation models, but low accuracy significantly limits their real-world application. Recently, many novel and promising models based on Two-Branch Network (TBN) architecture have been proposed in the literature and achieve SOTA trade-off between speed and accuracy [15, 20, 38, 39, 52].

In this paper, we view the architecture of TBNs from the prospective of PID controller and point out that a TBN is equivalent to a PI controller, which suffers from the overshoot issue as illustrated in Figure 2. To alleviate this problem, we devise a novel three-branch network architecture, namely PIDNet, and demonstrate its superiority on Cityscapes [12], CamVid [5] and PASCAL Context [33] datasets. We also provide ablation study and feature visualization for better understanding of the functionality of each module in PIDNet. The source code can be accessed via: https://github.com/XuJiacong/PIDNet

The main contributions of this paper are three-fold:

- We make a connection between deep CNN and PID controller and propose a family of three-branch networks based on the PID controller architecture.
- Efficient modules, such as Bag fusion module designed to balance detailed and context features, are proposed to boost the performance of PIDNets.
- PIDNet achieves the best trade-off between inference speed and accuracy among all the existing models. In particular, PIDNet-S achieves 78.6% mIOU with speed of 93.2 FPS and PIDNet-L presents the highest accuracy (80.6% mIOU) in real-time doman on Cityscapes test set without acceleration tools.

2. Related Work

Representative methods towards high-accuracy and real-time requirements are discussed separately in this section.

2.1. High-accuracy Semantic Segmentation

Early approaches for semantic segmentation were based on an encoder-decoder architecture [4, 31, 40], where the encoder gradually enlarges its receptive field by strided convolutions or pooling operations and the decoder recovers detailed information from high-level semantics using deconvolutions or upsampling. However, spatial details could be easily ignored in the process of downsampling for encoder-decoder network. To alleviate this problem, dilated convolution [53] was proposed to enlarge the field of view without reducing the spatial resolution. Based on this, DeepLab series [7–9] achieved great improvement over previous works by employing dilated convolution with different dilation rates in the network. Note that dilated convolution is not suitable for hardware implementation due to its non-contiguous memory accesses. PSPNet [59] introduced a pyramid pooling module (PPM) to parse multi-scale context information and HRNet [48] utilized multiple paths and bilateral connections to learn and fuse the representations in different scales. Inspired from the long-range dependency parsing ability of attention mechanism [47] for language machine, non-local operation [49] was introduced into computer vision and led to many accurate models [17, 23, 55].

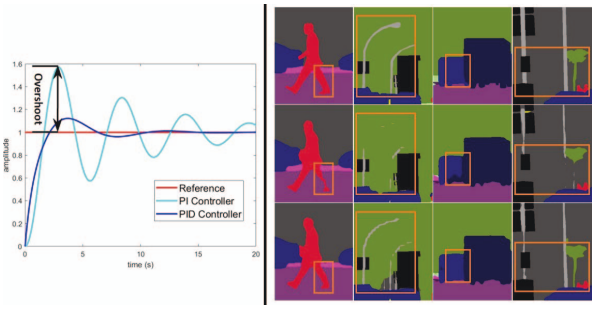

Figure 2. Overshoot issue for dynamic system (left |) and image segmentation (| right). Left |: Step responses of PI and PID controllers for a second-order system; | Right: From the first row to the last row, the images are cropped from ground truth, outputs of DDRNet-23 [20] and ADB-Bag-DDRNet-23 (ours), respectively.

2.2. Real-time Semantic Segmentation

Many network architectures have been proposed to achieve the best trade-off between inference speed and accuracy, which could be roughly summarized as below.

Light-weight encoder and decoder SwiftNet [35] employed one low-resolution input to obtain high-level semantics and another high-resolution input to provide sufficient details for its lightweight decoder. DFANet [27] introduced a light-weight backbone by modifying the architecture of Xception [11], which was based on depth-wise separable convolution, and reduced the input size for faster inference speed. ShuffleSeg [18] adopted ShuffleNet [57], which combined channel shuffling and group convolution, as its backbone to reduce the computational cost. However, most of these networks are still in the form of encoder-decoder architecture and they require the information flow go through

the deep encoder and then reverse back to pass the decoder, which introduces too much latency. Besides, since the optimization for depth-wise separable convolution on GPU is not mature, traditional convolution presents faster speed while having more FLOPs and parameters [35]. Thus, we seek for more efficient model that avoids convolution factorization and encoder-decoder architecture.

Two-branch network architecture Contextual dependency can be extracted by large receptive field, and spatial details are vital for boundary delineation and small-scale object recognition. To take both sides into account, authors of BiSeNet [52] proposed a two-branch network (TBN) architecture, which contains two branches with different depths for context embedding and detail parsing along with a feature fusion module (FFM) to fuse the context and detailed information. Several follow-up works based on this architecture have been proposed to boost its representation ability or reduce its model complexity [38, 39, 51]. Specifically, DDRNet [20] introduced bilateral connections to enhance information exchange between context and detailed branches, achieving state-of-the-art results in real-time semantic segmentation. Nevertheless, direct fusion of original detailed semantics and low-frequency context information has the risk of that object boundaries being overly corroded by surrounding pixels and small objects being overwhelmed by adjacent large ones (as shown in Figure 2 and 3).

3. Method

A PID controller contains three components: a proportional (P) controller, an integral (I) controller and a derivative (D) controller, as illustrated in Figure 3-Upper. The implementation of PI controller could be written as:

$$c_{out}[n] = k_p e[n] + k_i \sum_{i=0}^{n} e[i] \tag{1}$$

P controller focuses on current signal, while I controller accumulates all the past signals. Due to the inertia effect of accumulation, overshoot will happen to the output of simple PI controller when the signal changes oppositely. Then, D controller was introduced and if the signal become smaller, the D component will become negative and serves as a damper to reduce the overshoot. Similarly, TBNs parse the context and detailed information by multiple convolutional layers with and without strides, respectively. Consider a simple 1D example, where both detailed and context branches consist of 3 layers without BNs and ReLUs. Then, the output maps can be calculated as:

$$O_D[i] = K^D_{i-3} I[i-3] + ... + K^D_i I[i] + ... + K^D_{i+3} I[i+3] \tag{2}$$

$$O_C[i] = K^C_{i-7} I[i-7] + ... + K^C_i I[i] + ... + K^C_{i+7} I[i+7] \tag{3}$$

where, $K^D_i = k_{31}k_{22}k_{13} + k_{31}k_{23}k_{12} + k_{32}k_{21}k_{13} + k_{32}k_{22}k_{12} + k_{32}k_{23}k_{13} + k_{33}k_{21}k_{12} + k_{33}k_{22}k_{11}$ and $K^C_i = k_{32}k_{22}k_{12}$. Here, k_{mn} refers to the n-th value of the kernel in layer m. Since $|k_{mn}|$ are mostly distributed in $(0, 0.01)$ (92% for DDRNet-23) and are bounded by 1, the coefficient for each item will decrease exponentially with more layers. Thus, for each input vector, a larger number of items means a higher possibility to contribute to the final output. For detail branch, $I[i-1]$, $I[i]$, and $I[i+1]$ occupy over 70% of the total items, which means that **the detail branch focuses more on the local information**. On the contrary, $I[i-1]$, $I[i]$, and $I[i+1]$ only occupies less than 26% of the total items in context branch, so **the context branch emphasizes the surrounding information**. Figure 3-Bottom shows that the context branch is less sensitive to the change of local information than the detail branch. The behavior of detail and context branches in the spatial domain is similar to the P (current) and I (all previous) controllers in time domain.

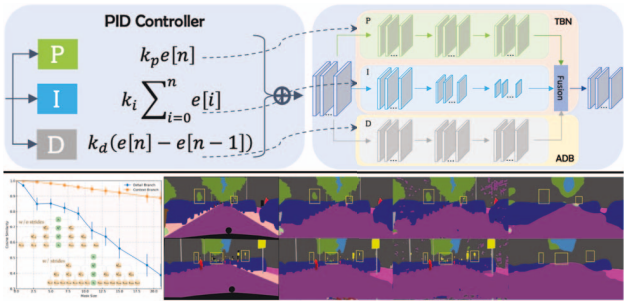

Figure 3. Upper |: The analogy between PID controller and proposed network; | Bottom: Left: Zero out surrounding mask area and calculate the similarity between current and original features for each pixel; Right: From the first to the last column, the images refer to ground truth, predictions of all branches, the detailed branch only, and the context branch only of DDRNet-23.

Replace z^{-1} by $e^{-j\omega}$ in the z-transform of a PID controller, which could be represented as:

$$C(z) = k_p + k_i(1 - e^{-j\omega})^{-1} + k_d(1 - e^{-j\omega}) \tag{4}$$

when the input frequency ω increases, the gain of I and D controllers will becomes smaller and larger, respectively, so the P, I, and D controllers work as allpass, lowpass filter, and highpass filter. Since PI controller focuses more on the low-frequency part of the input signal and cannot react immediately to the rapid change of the signal, it inherently suffers from the overshoot problem. The D controller reduces the overshoot by enabling the control output sensitive to the change of input signal. Figure 3-Bottom shows that the detail branch parses all kinds of semantic information even though not accurate, whereas **the context branch aggregates the low-frequency context information** and works similarly with a large averaging filter on semantics. Direct fusion of detailed and context information leads to missing of some detailed features. Thus, we conclude that TBN is equivalent to a PI controller in Fourier domain.

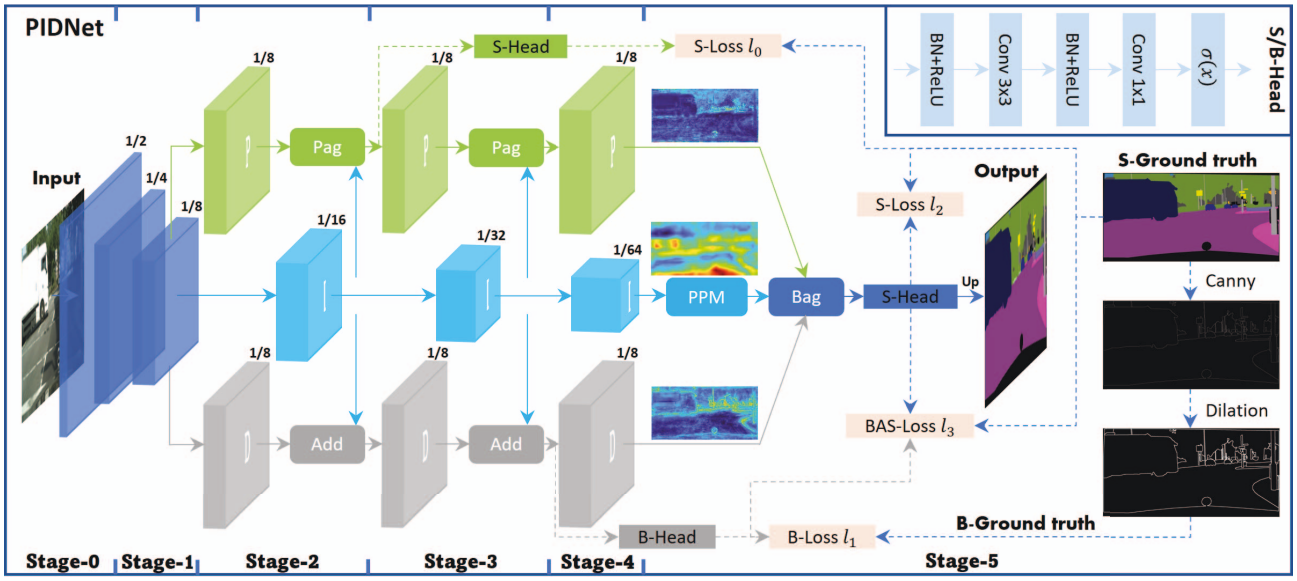

Figure 4. **An overview of the basic architecture of our proposed Proportional-Integral-Derivative Network (PIDNet).** S and B denote semantic and boundary, and Add and Up refer to element-wise summation and bilinear Upsampling operation, respectively; BAS-Loss represents the boundary-awareness CE loss [46]. Dashed lines and associate blocks will be ignored in the inference stage.

3.1. PIDNet: A Novel Three-branch Network

To mitigate the overshoot problem, we attach an auxiliary derivative branch (ADB) to the TBN to mimic the PID controller spatially and highlight the high-frequency semantic information. The semantics for pixels inside each object are consistent and only become inconsistent along the boundary of adjacent objects, so the difference of semantics is nonzero only at the object boundary and the objective of ADB is boundary detection. Accordingly, we establish a new three-branch real-time semantic segmentation architecture, namely Proportional-Integral-Derivative Network (PIDNet), which is shown in Figure 4.

PIDNet possesses three branches with complementary responsibilities: the proportional (P) branch parses and preserves detailed information in high-resolution feature maps; the integral (I) branch aggregates context information both locally and globally to parse long-range dependencies; and the derivative (D) branch extracts high-frequency features to predict boundary regions. As [20], we also adopt cascaded residual blocks [19] as the backbone for hardware friendliness. Besides, the depths for the P, I and D branches are set to be moderate, deep and shallow for efficient implementation. Consequently, a family of PIDNets (PIDNet-S, M and L) are generated by deepening and widening the model.

Following [20, 28, 51], we place a semantic head at the output of the first Pag module to generate the extra semantic loss l_0 for better optimization of entire network. Instead of dice loss [13], weighted binary cross entropy loss l_1 is adopted to deal with the imbalanced problem of boundary detection since coarse boundary is preferred to highlight the boundary region and enhance the features for small objects. l_2 and l_3 represents the CE loss, while we utilize the boundary-awareness CE loss [46] for l_3 using the output of boundary head to coordinate semantic segmentation and boundary detection tasks and enhance the function of Bag module. The calculation of BAS-Loss can be written as:

$$l_3 = -\sum_{i,c}\{1 : b_i > t\}(s_{i,c} log \hat{s_{i,c}}) \tag{5}$$

where t refers to predefined threshold and b_i, $s_{i,c}$ and $\hat{s_{i,c}}$ are the output of boundary head, segmentation ground-truth and prediction result of the i-th pixel for class c, respectively. Therefore, the final loss for PIDNet is:

$$Loss = \lambda_0 l_0 + \lambda_1 l_1 + \lambda_2 l_2 + \lambda_3 l_3 \tag{6}$$

Empirically, we set the parameters for the training loss of PIDNet as $\lambda_0 = 0.4$, $\lambda_1 = 20$, $\lambda_2 = 1$, $\lambda_3 = 1$ and $t = 0.8$.

3.2. Pag: Learning High-level Semantics Selectively

The lateral connection utilized in [20, 35, 48] enhances the information transmission between feature maps in different scales and improves the representation ability of their models. In PIDNet, the rich and accurate semantic information provided by I branch is crucial for detail parsing and boundary detection of the P and D branches, both of which contain relatively less layers and channels. Thus, we treat the I branch as the backup for other two branches and enable it to provide required information to them. Different from

the D branch that directly adds the provided feature maps, we introduce a **P**ixel-**a**ttention-**g**uided fusion module (Pag), which is shown in Figure 5, for the P branch to selectively

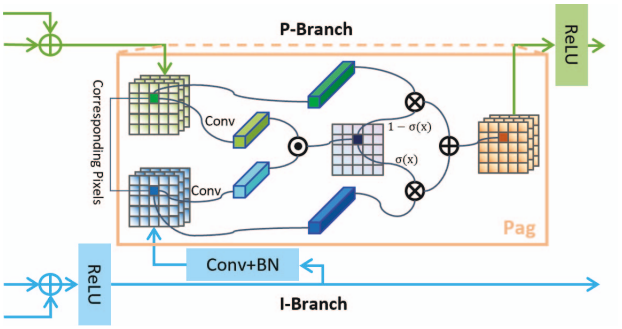

Figure 5. Illustration of Pag module. $\sigma(x)$ denotes the Sigmoid function; The kernel sizes of all the convolutions here are 1×1.

learn the useful semantic features from I branch without being overwhelmed. The underlying concept for Pag is borrowed from attention mechanisms [47]. Define the vectors for the corresponding pixels in feature maps from the P and I branch as $\vec{v_p}$ and $\vec{v_i}$, respectively, then the output of the Sigmoid function could be represented as:

$$\sigma = Sigmoid(f_p(\vec{v_p}) \cdot f_i(\vec{v_i})) \tag{7}$$

where σ indicates the possibility of these two pixels belonging to the same object. If σ is high, we trust $\vec{v_i}$ more since the I branch is semantically rich and accurate, and vise versa. Thus, the output of the Pag can be written as:

$$Out_{Pag} = \sigma\vec{v_i} + (1-\sigma)\vec{v_p} \tag{8}$$

3.3. PAPPM: Fast Aggregation of Contexts

For better global scene prior construction, PSPNet [59] introduced a pyramid pooling module (PPM), which concatenates multi-scale pooling maps before convolution layer to form local and global context representations. Deep Aggregation PPM (DAPPM) proposed by [20] further improved the context embedding ability of PPM and showed superior performance. Nevertheless, the computation process of DAPPM cannot be parallelized regarding its depth, which is time-consuming and DAPPM contains too many channels for each scale, which may surpasses the representation ability of lightweight models. Thus, we modify the connections in DAPPM to make it parallelizable, which is shown in Figure 6, and reduce the number of channels for each scale from 128 to 96. This new context harvesting module is called Parallel Aggregation PPM (PAPPM) and is applied in PIDNet-M and PIDNet-S to guarantee their speeds. For our deep model: PIDNet-L, we still choose the DAPPM considering its depth but reduce its number of channels for less computation and faster speed.

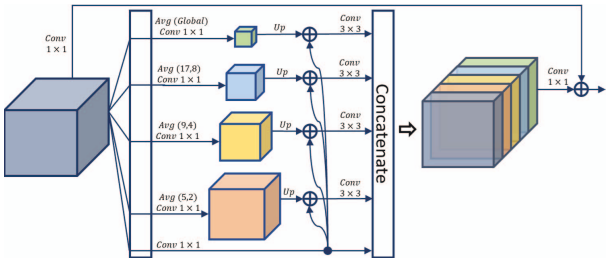

Figure 6. The parallel structure of PAPPM. Avg (5,2) means average pooling with kernel size of 5×5 and strides of 2.

3.4. Bag: Balancing the Details and Contexts

Given the boundary features extracted by ADB, we employ boundary attention to guide the fusion of detailed (P) and context (I) representations. Specifically, we design a **B**oundary-**a**ttention-**g**uided fusion module (Bag), shown in Figure 7, to fill the high-frequency and low-frequency areas with detailed and context features, respectively. Note that the context branch is semantically accurate but it loses too much spatial and geometric details especially for the boundary region and small object. Thanks to the detailed branch, which preserves spatial details better, we force the model to trust the detailed branch more along the boundary region and utilize the context features to fill other areas. Define

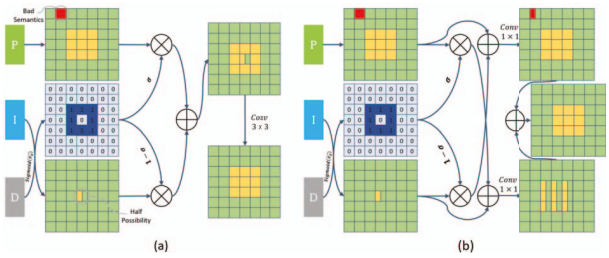

Figure 7. Single channel implementations of (a) Bag and (b) Light-Bag modules in extreme case. P, I and D refer to the outputs of detailed, context and boundary branches, respectively. σ denotes the output of Sigmoid function.

the vectors for the corresponding pixels of P, I and D feature maps as $\vec{v_p}$, $\vec{v_i}$ and $\vec{v_d}$, respectively, then the outputs of Sigmoid , Bag and Light-Bag could be represented as:

$$\boldsymbol{\sigma} = Sigmoid(\vec{v_d}) \tag{9}$$

$$Out_{bag} = f_{out}((1-\boldsymbol{\sigma}) \otimes \vec{v_i} + \boldsymbol{\sigma} \otimes \vec{v_p}) \tag{10}$$

$$Out_{light} = f_p((1-\boldsymbol{\sigma}) \otimes \vec{v_i} + \vec{v_p}) + f_i(\boldsymbol{\sigma} \otimes \vec{v_p} + \vec{v_i}) \tag{11}$$

where f refers to the composition of convolutions, batch normalizations and ReLUs. Even though we replace 3×3 convolutions in Bag by two 1×1 convolutions in Light-Bag, the functionalities of Bag and Light-Bag are similar, that is when $\sigma > 0.5$ the model trusts more on detailed features, otherwise context information is preferred.

4. Experiment

In this section, our models will be trained and tested on Cityscapes, CamVid and PASCAL Context benchmarks.

4.1. Datasets

Cityscapes. Cityscapes [12] is one of the most well-known urban scene parsing datasets, which contains 5000 images collected from the car perspective in different cities. These images are divided into sets with numbers of 2975, 500, and 1525 for training, validation and testing. The image resolution is 2048×1024, which is challenging for real-time models. Only the fine annotated dataset is used here.

CamVid. CamVid [5] provides 701 images of driving scenes, which is partitioned into 367, 101 and 233 for training, validation and test. The image resolution is of 960×720 and the number of annotated categories is 32, of which 11 classes are used for fair comparison with previous works.

PASCAL Context. Semantic labeling for whole scene is provided in PASCAL Context [33], which contains 4998 images for training and 5105 images for validation. While this dataset is mainly used for benchmarking high-accuracy models, we utilized it here to show the generalization ability of PIDNets. Both 59 and 60-class scenarios are evaluated.

4.2. Implementation Details

Pretraining. Before fine-tuning our models, we pre-train them by ImageNet [41] as most of previous works doing [20, 34, 35]. We remove the D branch and directly merge the features in final stage to construct the classification models. The total number of training epochs is 90 and the learning rate is scheduled to be 0.1 initially and multiplied by 0.1 at epoch 30 and 60. The images are randomly cropped into 224×224 and flipped horizontally for data augmentation.

Training. Our training protocols are almost the same as previous works [15, 20, 52]. Specifically, we adopt the poly strategy to update the learning rate and random cropping, random horizontal flipping, and random scaling in the range of $[0.5, 2.0]$ for data augmentation. The number of training epochs, the initial learning rate, weight decay, cropped size and batch size for Cityscapes, CamVid and PASCAL Context could be summarized as [484, $1e^{-2}$, $5e^{-4}$, 1024×1024, 12], [200, $1e^{-3}$, $5e^{-4}$, 960×720, 12] and [200, $1e^{-3}$, $1e^{-4}$, 520×520, 16], respectively. Following [20, 51], we fine-tune the Cityscapes pretrained models for CamVid and stop the training process when $lr < 5e^{-4}$ to avoid overfitting.

Inference. Before testing, our models are trained by both train and val set for Cityscapes and CamVid. We measure the inference speed on the platform consists of single RTX 3090, PyTorch 1.8, CUDA 11.2, cuDNN 8.0 and Windows-Conda environment. Using the measurement protocol proposed by [10] and following [20, 35, 45], we integrate the batch normalization into the convolutional layers and set the batch size to be 1 for measurement of inference speed.

4.3. Ablation Study

ADB for Two-branch Networks. To demonstrate the effectiveness of PID methodology, we combine ADB and Bag with existing models. Here, two representative two-branch networks: BiSeNet [52] and DDRNet [20] equipped with ADB and Bag are implemented and achieve much higher accuracy on Cityscapes val set compared with their original models, which is shown in Table 1. However, additional computation significantly slow down their inference speed, which then triggers us to establish PIDNet.

Model	ADB-Bag		mIOU	FPS
	w/o	w/		
BiSeNet(Res18)	✓		75.4	63.2
		✓	**76.7**	52.1
DDRNet-23	✓		79.5	51.4
		✓	**80.0**	39.2

Table 1. Ablation study of ADB-Bag for BiSeNet and DDRNet.

Collaboration of Pag and Bag. P branch utilizes Pag module to learn useful information from I branch without being overwhelmed before fusion stage and Bag module is introduced to guide the fusion of detailed and context features. As Table 2 shows, lateral connection could significantly improve the model accuracy and pretraining could further boost its performance. In our scenario, the combinations of Add lateral connection and Bag fusion module or Pag lateral connection and Add fusion module make little sense since preservation of details should be consistent in the entire network. Thus, we only need to compare the performance of Add + Add and Pag + Bag and the experimental results in Table 2 and 3 demonstrate the superiority of the collaboration of Pag and Bag (or Light-Bag). The visualization of feature maps in Figure 8 shows that the small objects become much darker compared with large objects in the Sigmoid map for second Pag, where I branch loses more detailed information. Also, the features in boundary

IM	Lateral			Fusion		mIOU
	None	Add	Pag	Add	Bag	
		✓		✓		79.3
			✓	✓		78.1
✓	✓			✓		80.0
✓		✓		✓		80.7
✓			✓	✓		80.5
✓		✓			✓	80.5
✓			✓		✓	**80.9**

Table 2. Ablation study of Pag and Bag on PIDNet-L. IM refers to ImageNet [41] pretraining, Add represents the element-wise summation operation and None means there is no lateral connection.

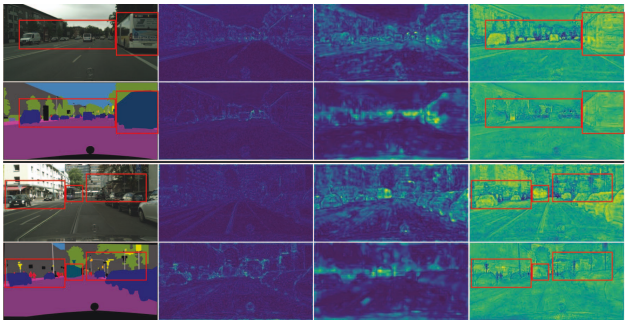

Figure 8. Feature visualization of Pag module. The maps in the first row from left to right are the original input image, P input, I input and output of Sigmoid function for the first Pag; The maps in the second row are groudtruth, P, I inputs and Sigmoid output for the second Pag; The third and fourth rows are for another image.

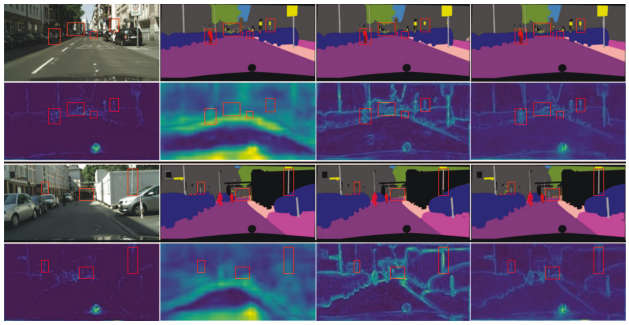

Figure 9. Feature visualization of Bag module. The maps in the first row from left to right are the original input image, ground truth, predictions of DDRNet-23 and PIDNet-M; The maps in the second row are P, I and D input and final output for Light-Pag in PIDNet-M; The third and fourth rows are for another image.

regions and small objects are greatly enhanced in the output of Bag module, which is illustrated in Figure 9 and explains the reason why we choose coarse boundary detection.

PPM		Fusion		mIOU	FPS
DAPPM	PAPPM	Add	Bag		
✓			✓	**78.8**	83.7
	✓	✓		78.4	**97.8**
	✓		✓	**78.8**	93.2

Table 3. Ablation study of PAPPM and Light-Bag on PIDNet-S.

Efficiency of PAPPM. For real-time models, a heavy context aggregation module could drastically slow down the inference speed and may surpass the representation ability of the network. Thus, we proposed the PAPPM, which is constituted by parallel structure and small number of parameters. The experimental results in Table 3 show that PAPPM achieves the same accuracy as DAPPM [20] but presents a speed-up of 9.5 FPS for our light-weight model.

Extra Loss			OHEM	mIOU
l_0	l_1	l_3		
				78.6
✓				78.8
✓	✓			79.9
✓	✓	✓		80.5
✓	✓	✓	✓	**80.9**

Table 4. Ablation study of extra losses and OHEM for PIDNet-L.

Effectiveness of Extra losses. Three extra losses were introduced to PIDNet to boost the optimization of entire network and emphasize the functionality for each components. According to Table 4, boundary loss l_1 and boundary-awareness loss l_3 are necessary for better performance, especially the boundary loss (+1.1% mIOU), which strongly proves the necessity of D branch, and Online Hard Example Mining (OHEM) [43] further improves the accuracy.

4.4. Comparison

CamVid. For CamVid [5] dataset, only the accuracy of DDRNet is comparable with our models, so we test its speed on our platform with the same setting for fair comparison considering our platform is more advanced than theirs. The experimental results in Table 5 show that the accuracy

Model	mIOU	#FPS	GPU
MSFNet [45]	75.4	91.0	GTX 2080Ti
PP-LiteSeg-T [37]	75.0	154.8	GTX 1080Ti
TD2-PSP50 [22]	76.0	11.0	TITAN X
BiSeNetV2† [51]	76.7	124.0	GTX 1080Ti
BiSeNetV2-L† [51]	78.5	33.0	GTX 1080Ti
HyperSeg-S [34]	78.4	38.0	GTX 1080Ti
HyperSeg-L [34]	79.1	16.6	GTX 1080Ti
DDRNet-23-S†* [20]	78.6	**182.4**	RTX 3090
DDRNet-23†* [20]	80.6	116.8	RTX 3090
PIDNet-S†	80.1	153.7	RTX 3090
PIDNet-S-Wider†	**82.0**	85.6	RTX 3090

Table 5. Comparison of speed and accuracy on CamVid. The models pretrained by Cityscapes [12] are marked with †; The inference speeds for models marked with * are tested on our platform.

of all our models exceeds 80% mIOU and PIDNet-S-Wider, which simply doubles the number of channels for PIDNet-S, achieves the highest accuracy with a big margin ahead of previous models. Besides, the accuracy of PIDNet-S surpasses previous state-of-art model: DDRNet-23-S by 1.5% mIOU with only around 1 ms latency increase.

Cityscapes. Previous real-time works treat Cityscapes [12] as the standard benchmark considering its high-quality annotation. As shown in Table 6, we test the inference speeds

Model	mIOU		#FPS	GPU	Resolution	#GFLOPs	#Params
	Val	Test					
MSFNet [45]	-	77.1	41	RTX 2080Ti	2048×1024	96.8	-
DF2-Seg1 [29]	75.9	74.8	67.2	GTX 1080Ti	1536×768	-	-
DF2-Seg2 [29]	76.9	75.3	56.3	GTX 1080Ti	1536×768	-	-
SwiftNetRN-18 [35]	75.5	75.4	39.9	GTX 1080Ti	2048×1024	104.0	11.8M
SwiftNetRN-18 ens [35]	-	76.5	18.4	GTX 1080Ti	2048×1024	218.0	24.7M
CABiNet [26]	76.6	75.9	76.5	RTX 2080Ti	2048×1024	12.0	2.64M
BiSeNet(Res18) [52]	74.8	74.7	65.5	GTX 1080Ti	1536×768	55.3	49M
BiSeNetV2-L [51]	75.8	75.3	47.3	GTX 1080Ti	1024×512	118.5	-
STDC1-Seg75* [15]	74.5	75.3	74.8	RTX 3090	1536×768	-	-
STDC2-Seg75* [15]	77.0	76.8	58.2	RTX 3090	1536×768	-	-
PP-LiteSeg-T2* [37]	76.0	74.9	96.0	RTX 3090	1536×768	-	-
PP-LiteSeg-B2* [37]	78.2	77.5	68.2	RTX 3090	1536×768	-	-
HyperSeg-M* [34]	76.2	75.8	59.1	RTX 3090	1024×512	7.5	10.1
HyperSeg-S* [34]	78.2	78.1	45.7	RTX 3090	1536×768	17.0	10.2
SFNet(DF2)* [28]	-	77.8	87.6	RTX 3090	2048×1024	-	10.53M
SFNet(ResNet-18)* [28]	-	78.9	30.4	RTX 3090	2048×1024	247.0	12.87M
SFNet(ResNet-18)†* [28]	-	80.4	30.4	RTX 3090	2048×1024	247.0	12.87M
DDRNet-23-S* [20]	77.8	77.4	**108.1**	RTX 3090	2048×1024	36.3	5.7M
DDRNet-23* [20]	79.5	79.4	51.4	RTX 3090	2048×1024	143.1	20.1M
DDRNet-39* [20]	-	80.4	30.8	RTX 3090	2048×1024	281.2	32.3M
PIDNet-S-Simple	78.8	78.2	100.8	RTX 3090	2048×1024	46.3	7.6M
PIDNet-S	78.8	78.6	93.2	RTX 3090	2048×1024	47.6	7.6M
PIDNet-M	**80.1**	80.1	39.8	RTX 3090	2048×1024	197.4	34.4M
PIDNet-L	**80.9**	**80.6**	31.1	RTX 3090	2048×1024	275.8	36.9M

Table 6. Comparison of speed and accuracy on Cityscapes. The models pretrained by other segmentation datasets are marked with †; The inference speeds for models marked with * are tested on our platform. The GFLOPs for PIDNet is derived based on full-resolution input.

of the models published in recent two years on the same platform without any acceleration tool as PIDNets for fair comparison. The experimental results show that PIDNets achieve the best trade-off between inference speed and accuracy. Specifically, PIDNet-L surpasses SFNet(ResNet-18)† and DDRNet-39 in terms of speed and accuracy and becomes the most accurate model in real-time domain by rising the test accuracy from 80.4% to 80.64% mIOU. PIDNet-M and PIDNet-S also provide much higher accuracy compared with other models with similar inference speeds. Removing Pag and Bag modules from PIDNet-S, we provide an even faster option: PIDNet-S-Simple, which has weaker generalization ability but still presents highest accuracy among models with latency less than 10 ms.

PASCAL Context. The $Avg(17, 8)$ path in PAPPM is removed since the image size is too small in PASCAL Context [33]. Different from other two datasets, multi-scale and flip inference are utilized here for fair comparison with previous models. Even though there are less detailed annotations in PASCAL Context compared with previous two datasets, our models still achieve competitive performance among existing heavy networks, as shown in Table 7.

Model	BaseNet	mIOU-59	mIOU-60
DeepLab-v2 [7]	D-Res-101	-	45.7
RefineNet [30]	Res-152	-	47.3
PSPNet [59]	D-Res-101	47.8	-
Ding et al. [14]	D-Res-101	51.6	-
EncNet [56]	D-Res-101	52.6	-
HRNet [48]	V2-W48	54.0	48.3
PIDNet-M	-	51.0	46.0
PIDNet-L	-	51.9	46.6

Table 7. Comparison of accuracy on Pascal-Context (w/ and w/o background class). D-Res-101 refers to Dilated ResNet-101.

5. Conclusion

This paper presents a novel three-branch network architecture: PIDNet for real-time semantic segmentation. PIDNet achieves the best trade-off between inference time and accuracy. However, since PIDNet utilizes the boundary prediction to balance the detailed and context information, precise annotation around boundary, which usually requires a large amount of time, is preferred for better performance.

References

[1] Wangpeng An, Haoqian Wang, Qingyun Sun, Jun Xu, Qionghai Dai, and Lei Zhang. A pid controller approach for stochastic optimization of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8522–8531, 2018. 1

[2] Saeid Asgari Taghanaki, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. Deep semantic segmentation of natural and medical images: a review. *Artificial Intelligence Review*, 54(1):137–178, 2021. 1

[3] Helon Vicente Hultmann Ayala and Leandro dos Santos Coelho. Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Systems with Applications*, 39(10):8968–8974, 2012. 1

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1, 2

[5] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 2, 6, 7

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 1

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2, 8

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2

[10] Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang Wang. Fasterseg: Searching for faster real-time semantic segmentation. *arXiv preprint arXiv:1912.10917*, 2019. 6

[11] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 2

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1, 2, 6, 7

[13] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 562–578, 2018. 4

[14] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2393–2402, 2018. 8

[15] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9716–9725, 2021. 2, 6, 8

[16] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020. 1, 2

[17] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 2

[18] Mostafa Gamal, Mennatullah Siam, and Moemen Abdel-Razek. Shuffleseg: Real-time semantic segmentation network. *arXiv preprint arXiv:1803.03816*, 2018. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[20] Yuanduo Hong, Huihui Pan, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021. 2, 3, 4, 5, 6, 7, 8

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[22] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8827, 2020. 7

[23] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 2

[24] A Jayachitra and R Vinodha. Genetic algorithm based pid controller tuning approach for continuous stirred tank reactor. *Advances in Artificial Intelligence (16877470)*, 2014. 1

[25] A Khodabakhshian and R Hooshmand. A new pid controller design for automatic generation control of hydro power systems. *International Journal of Electrical Power & Energy Systems*, 32(5):375–382, 2010. 1

[26] Saumya Kumaar, Ye Lyu, Francesco Nex, and Michael Ying Yang. Cabinet: efficient context aggregation network for low-latency semantic segmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13517–13524. IEEE, 2021. 8

[27] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9522–9531, 2019. 2

[28] Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, Shaohua Tan, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision*, pages 775–793. Springer, 2020. 4, 8

[29] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9145–9153, 2019. 8

[30] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 8

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1, 2

[32] Ruijun Ma, Shuyi Li, Bob Zhang, and Zhengming Li. Towards fast and robust real image denoising with attentive neural network and pid controller. *IEEE Transactions on Multimedia*, 2021. 1

[33] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014. 2, 6, 8

[34] Yuval Nirkin, Lior Wolf, and Tal Hassner. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4061–4070, 2021. 6, 7, 8

[35] Marin Orsic, Ivan Kreso, Petra Bevandic, and Sinisa Segvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12607–12616, 2019. 2, 3, 4, 6, 8

[36] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 2

[37] Juncai Peng, Yi Liu, Shiyu Tang, Yuying Hao, Lutao Chu, Guowei Chen, Zewu Wu, Zeyu Chen, Zhiliang Yu, Yuning Du, et al. Pp-liteseg: A superior real-time semantic segmentation model. *arXiv preprint arXiv:2204.02681*, 2022. 7, 8

[38] Rudra PK Poudel, Ujwal Bonde, Stephan Liwicki, and Christopher Zach. Contextnet: Exploring context and detail for semantic segmentation in real-time. *arXiv preprint arXiv:1805.04554*, 2018. 2, 3

[39] Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019. 2, 3

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 2

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 6

[42] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2

[43] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 7

[44] Alexey A Shvets, Alexander Rakhlin, Alexandr A Kalinin, and Vladimir I Iglovikov. Automatic instrument segmentation in robot-assisted surgery using deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 624–628. IEEE, 2018. 2

[45] Haiyang Si, Zhiqiang Zhang, Feifan Lv, Gang Yu, and Feng Lu. Real-time semantic segmentation via multiply spatial fusion network. *arXiv preprint arXiv:1911.07217*, 2019. 6, 7, 8

[46] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5229–5238, 2019. 4

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 5

[48] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 1, 2, 4, 8

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[50] Jiacong Xu and Shankar P Bhattacharyya. A pid controller architecture inspired enhancement to the pso algorithm. In *Future of Information and Communication Conference*, pages 587–603. Springer, 2022. 1

[51] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021. 3, 4, 6, 7, 8

[52] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 2, 3, 6, 8

[53] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2

[54] Xiaohui Yuan, Jianfang Shi, and Lichuan Gu. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications*, 169:114417, 2021. 1

[55] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European conference on computer vision*, pages 173–190. Springer, 2020. 2

[56] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. 8

[57] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 2

[58] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018. 2

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1, 2, 5, 8

[60] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 1