JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

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

Jiacong Xu, Zixiang Xiong, and Shankar P. Bhattacharyya

Email: jxu155@jhu.edu, zx@ece.tamu.edu, spb@tamu.edu

Code: https://github.com/XuJiacong/PIDNet

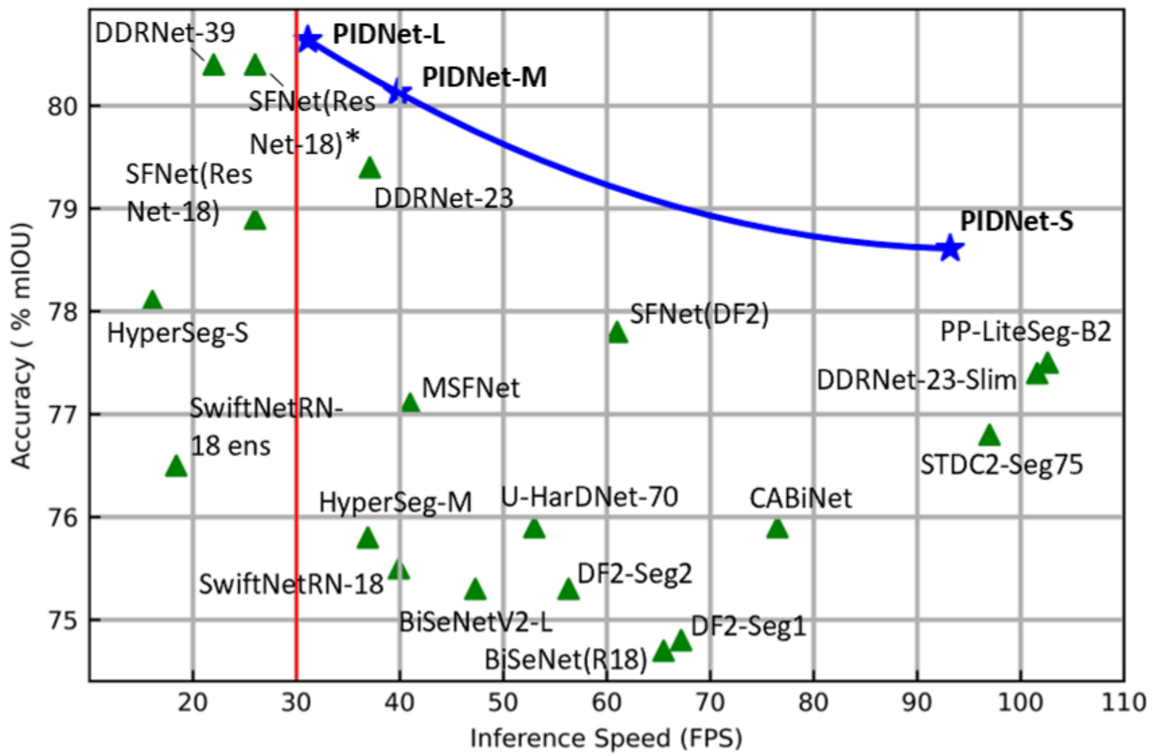
Paper: https://arxiv.org/abs/2206.02066

Poster: THU-AM-290

TEXAS A&M UNIVERSITY

Overview

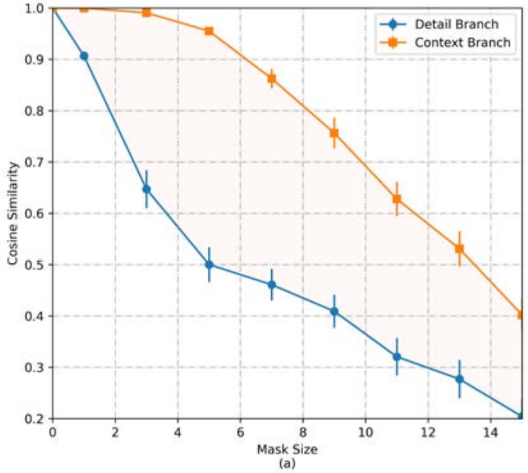
- We make a connection between deep CNN and PID controller and propose a family of three-branch networks based on the PID controller architecture.
- Efficient modules, such as Bag fusion module designed to balance detailed and context features, are proposed to boost the performance of PIDNets.
- PIDNet achieves the best trade-off between inference speed and accuracy among all the existing models.



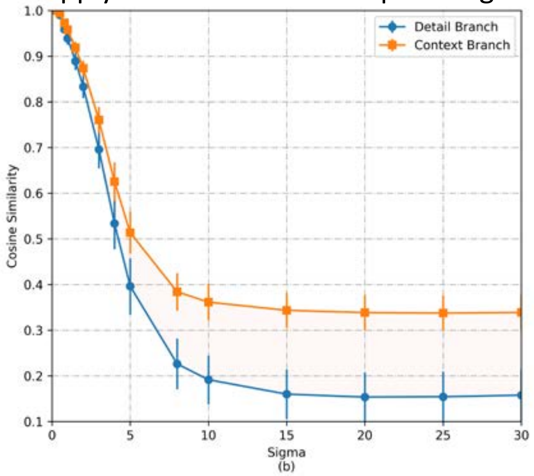
Motivation

- P controller focuses on current signal and works as all-pass filter, while I controller accumulates all the past signals and shows low-pass characteristics.
- Two-Branch Network (TBN) possesses similar properties with PI controller in both Time and Fourier domains.
- The detail branch focuses more on the local information, and the context branch emphasizes the surrounding information.
- The context branch contains more low-frequency information than the detail branch and is less sensitive to the loss of high-frequency signals.
- Some detailed predictions in detailed branch are overwhelmed in the final output of TBN – **overshoot**.

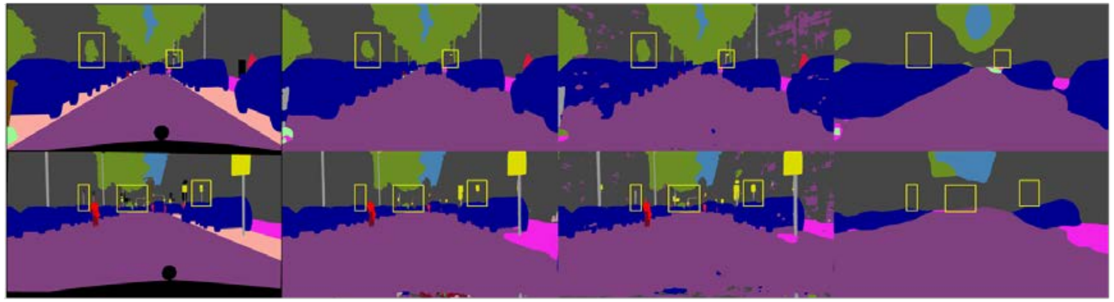
Zero out the mask area



Apply Gaussian filter on input image

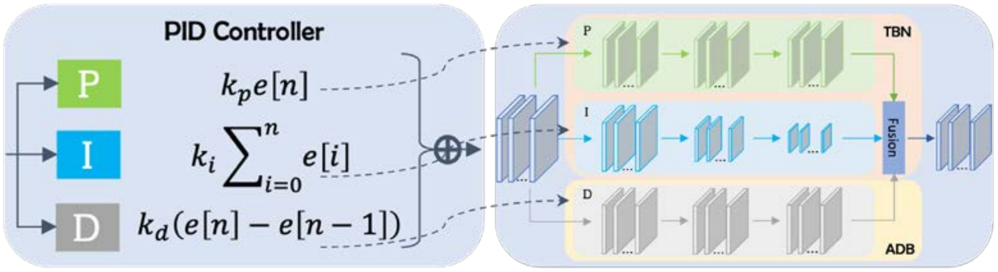


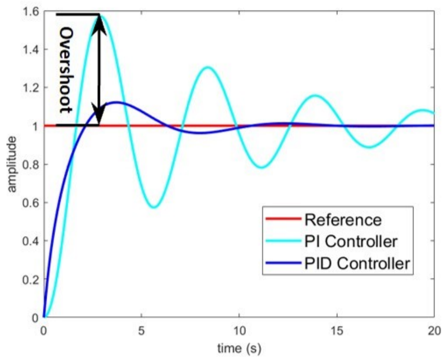
GT TBN Output Detailed Output Context Output

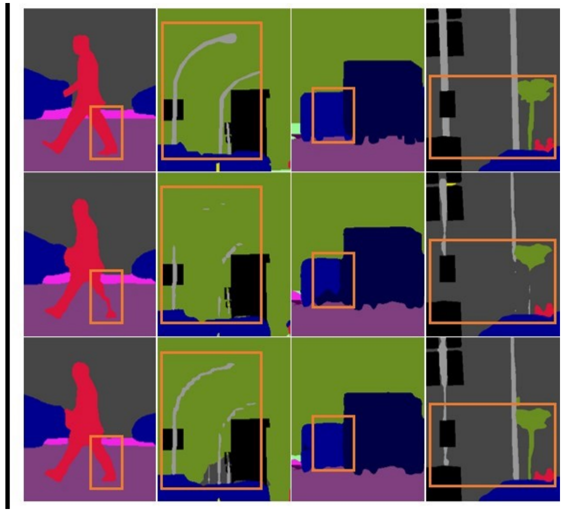


Motivation

- PI controllers suffers from the overshoot issue due to the inertia effect of accumulation in time domain and low-frequency property in Fourier domain.
- The D controller serves as a high-pass filter and reduces the overshoot by enabling the control output sensitive to the change of input signal.
- To mitigate the overshoot problem, we attach an auxiliary derivative branch (ADB) to the TBN to mimic the PID controller spatially. For simplicity, the objective of ADB is set to be boundary detection.







GT

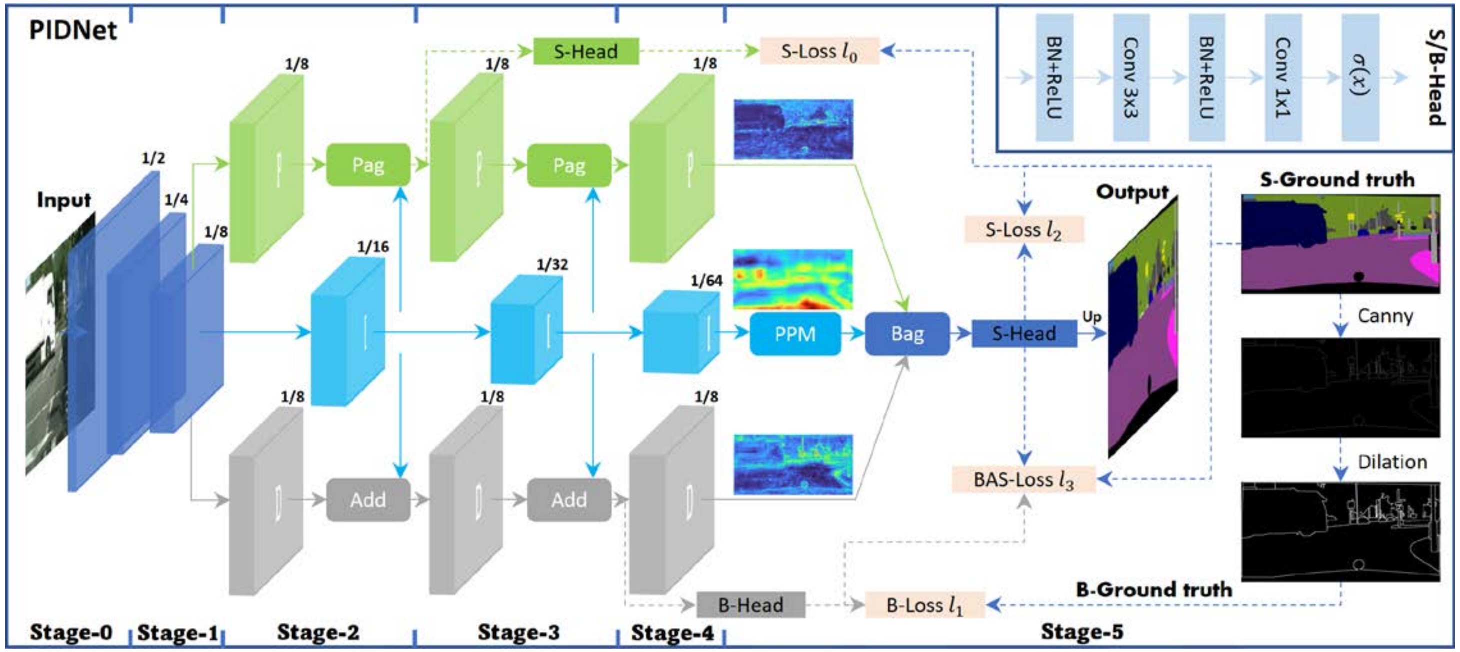
TBN

TBN+ADB

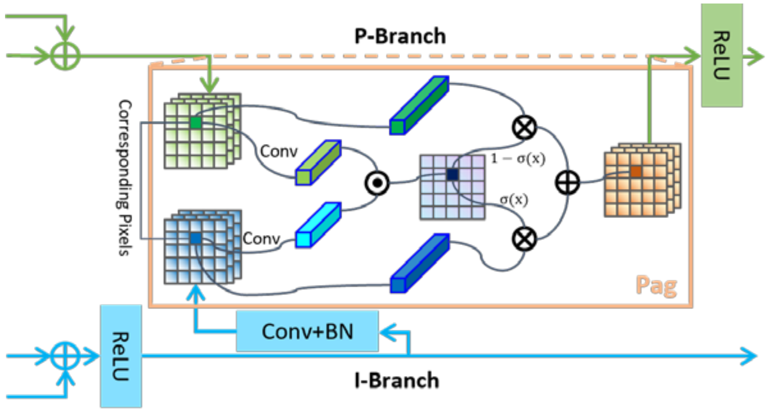
Method

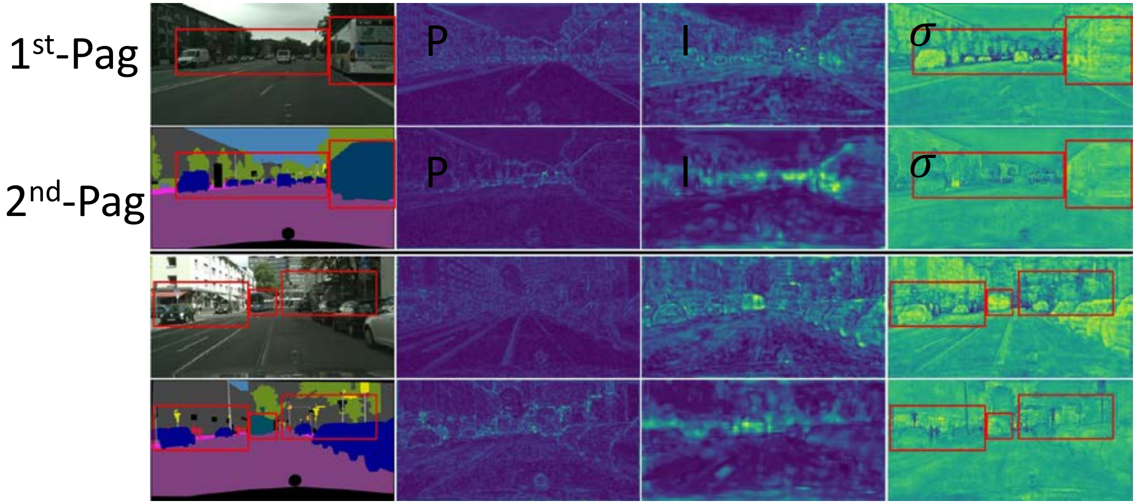
- PIDNet possesses three branches with complementary responsibilities:

1. the proportional (P) branch parses and preserves detailed information in high-resolution feature maps
2. the integral (I) branch aggregates context information both locally and globally to parse long-range dependencies
3. the derivative (D) branch extracts high-frequency features to predict boundary regions



Pag: Learning High-level Semantics Selectively by Pixel Attention

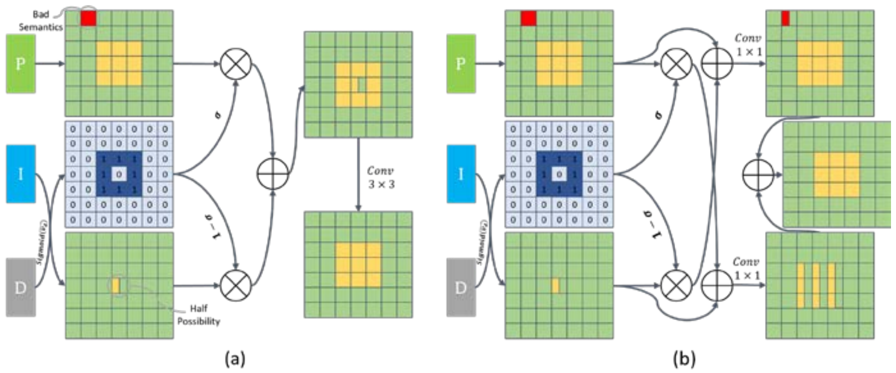


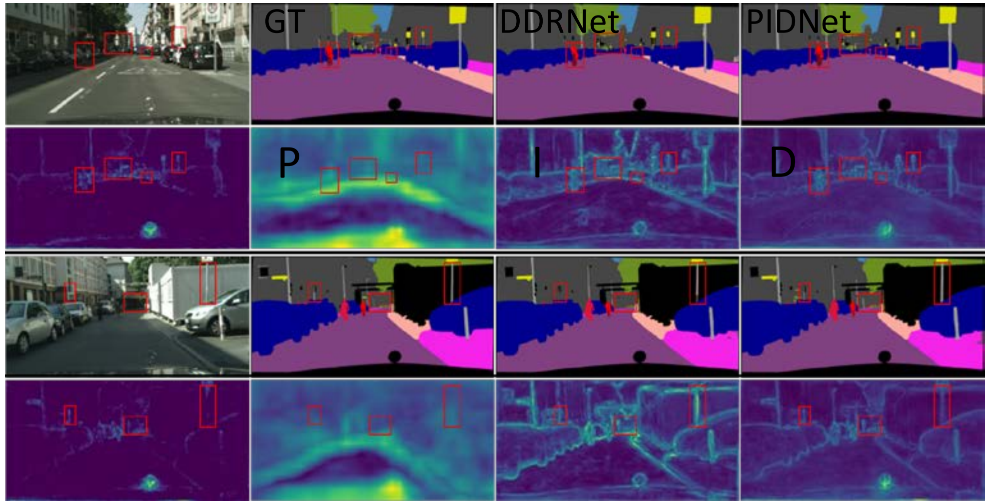


$$\sigma = Sigmoid(f_p(\vec{v_p}) \cdot f_i(\vec{v_i}))$$

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

Bag: Balancing the Details and Contexts by Boundary Attention





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

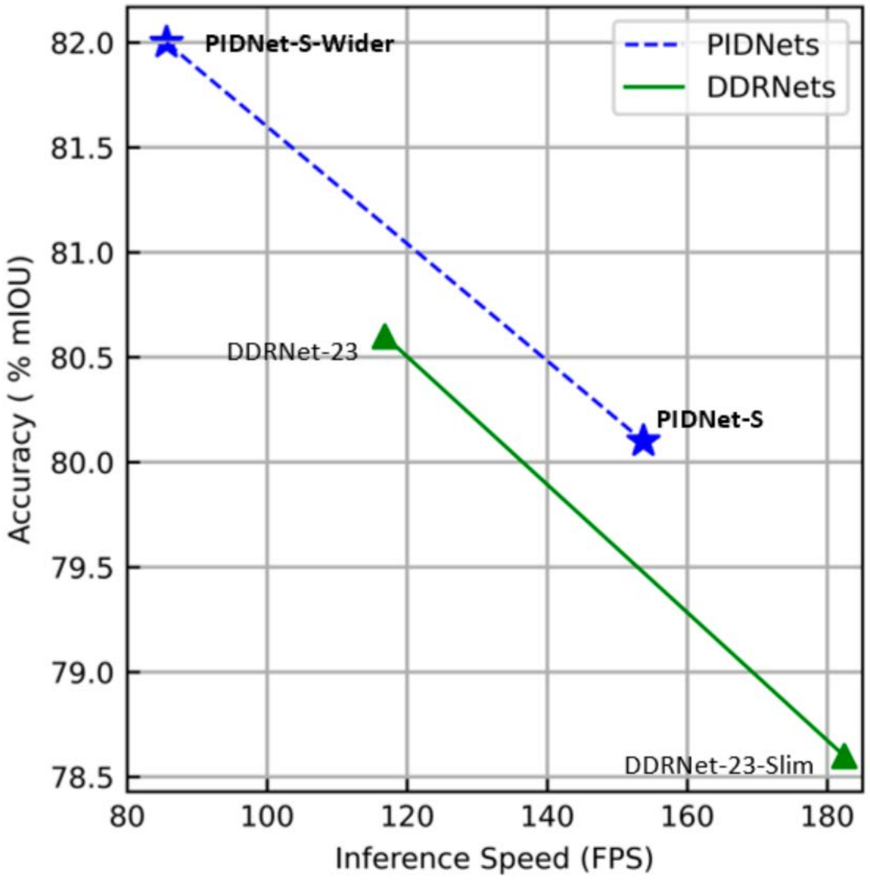
$$Out_{bag} = f_{out}((1-\boldsymbol{\sigma}) \otimes \vec{v_i} + \boldsymbol{\sigma} \otimes \vec{v_p})$$

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

Experiments -- CamVid

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

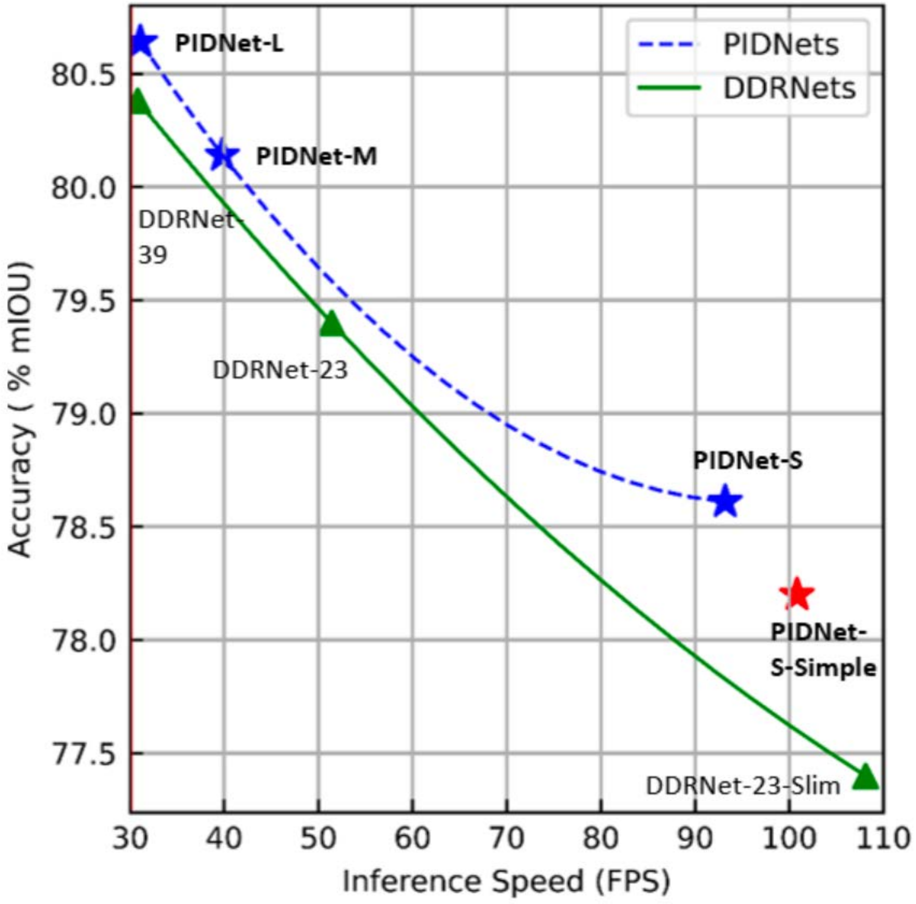
CamVid



Experiments -- Cityscapes

Model	mIOU Val	mIOU Test	#FPS	GPU	Resolution	#GFLOPs	#Params
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

Cityscapes



Experiment – Ablation

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

ADB can boost the performance of existing TBNs but introduces too much latency, so we redesign the entire network.

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

The collaboration of Pag and Bag improves the performance and generalization ability of PIDNets (the test accuracy of PIDNet-S is higher than PIDNet-Simple, where Pag and Bag are removed for faster speed.)

Conclusions

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