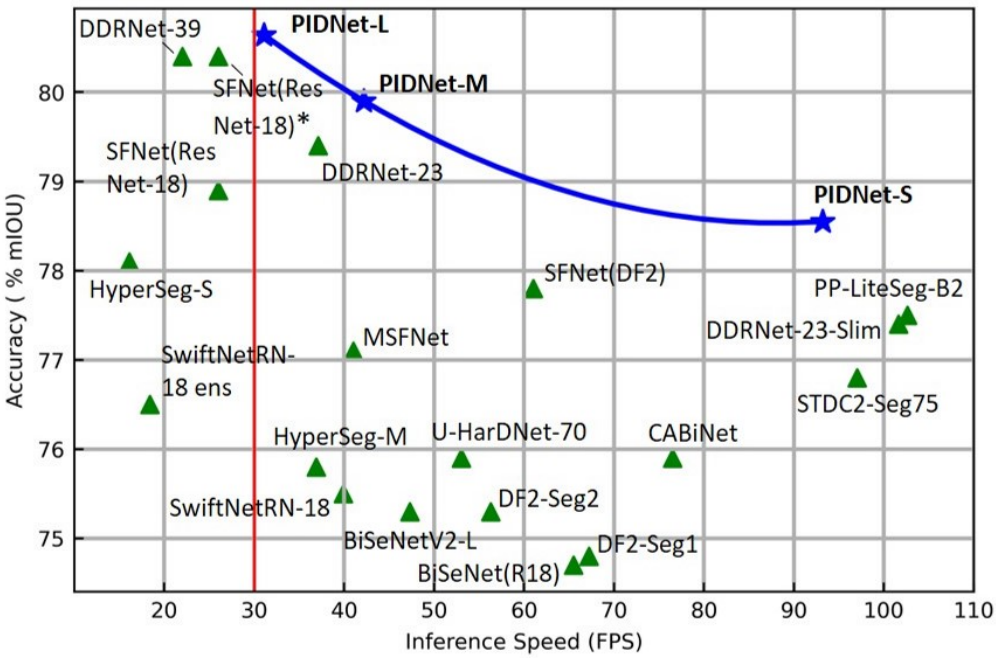
# PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller



This is the official repository for our recent work: PIDNet (PDF)

## Highlights




Comparison of inference speed and accuracy for real-time models on test set of Cityscapes.

- **Towards Real-time Applications**: PIDNet could be directly used for the real-time applications, such as autonomous vehicle and medical imaging.
- **A Novel Three-branch Network**: Addtional boundary branch is introduced to two-branch network to mimic the PID controller architecture and remedy the overshoot issue of previous models.
- **More Accurate and Faster**: PIDNet-S presents 78.6% mIOU with speed of 93.2 FPS on Cityscapes test set and 80.1% mIOU with speed of 153.7 FPS on CamVid test set. Also, PIDNet-L becomes the most accurate one (80.6% mIOU) among all the real-time networks for Cityscapes.

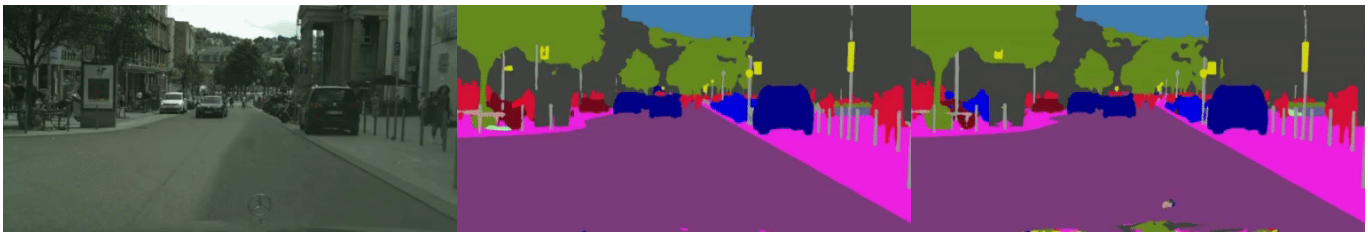
## Updates

- Fixed the data bug for Camvid and the new version of arXiv preprint will be available on Jun 13th. (Jun/09/2022)
- Our paper was marked as state of the art in Papers with Code. (Jun/06/2022)
- Our paper was submitted to arXiv for public access. (Jun/04/2022)
- The training and testing codes and trained models for PIDNet are available here. (Jun/03/2022)

## Demos

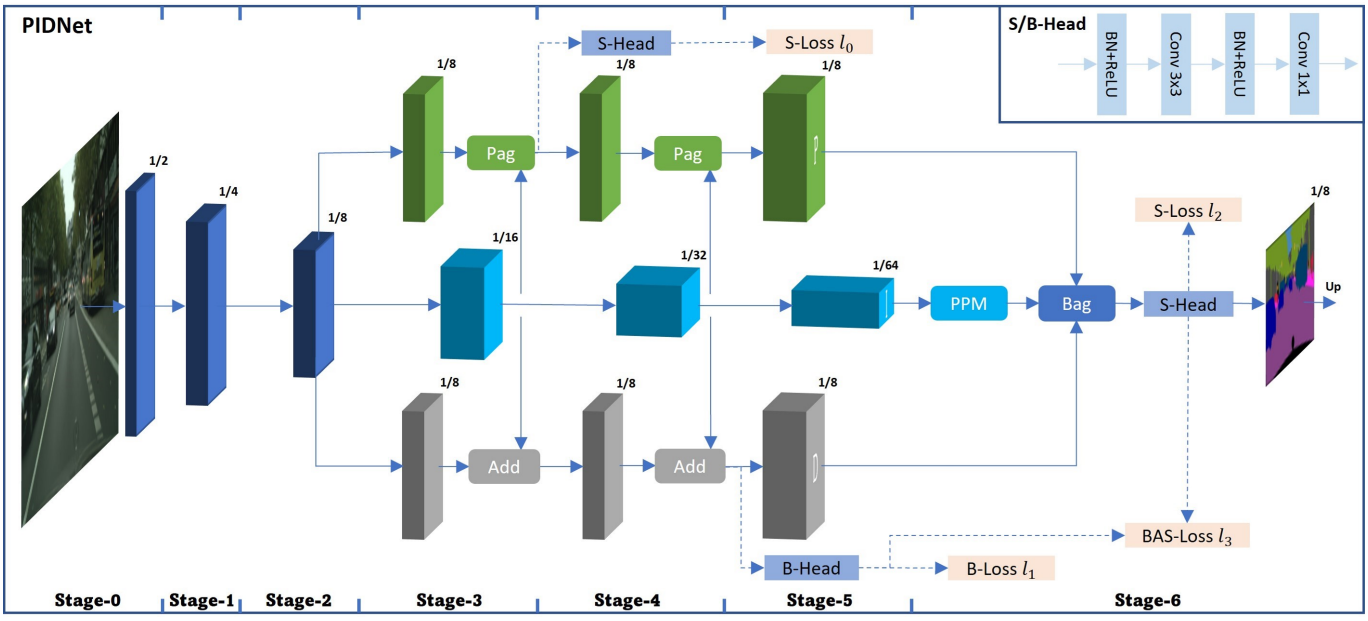
A demo of the segmentation performance of our proposed PIDNets: Original video (left) and predictions of PIDNet-S (middle) and PIDNet-L (right)

Cityscapes Stuttgart demo video #1



Cityscapes Stuttgart demo video #2

## Overview




An overview of the basic architecture of our proposed Proportional-Integral-Derivative Network (PIDNet).

P, I and D branches are responsiable for detail preservation, context embedding and boundary detection, respectively.

### Detailed Implementation

| Stage | Operation | | | Output |
|---|---|---|---|---|
| 0 | Conv 3×3, 2, C<br>Conv 3×3, 2, C | | | 256 |
| 1 | m×RB, 1, C | | | 256 |
| 2 | m×RB, 2, 2C | | | 128 |
| 3 | m×RB, 1, 2C<br>Pag | n×RB, 2, 4C<br>← → | RB, 1, C\2C<br>Add | 64 |
| 4 | m×RB, 1, 2C<br>Pag | n×RB, 2, 8C<br>← → | RBB\RB, 1, 2C<br>Add | 32 |
| 5 | RBB, 1, 4C | 2×RBB, 2, 16C | RBB, 1, 4C | 16 |
| 6 | ↓<br>→ | PPM<br>Bag | ↓<br>← | 128 |
| | Conv 3 × 3, 1, 256\128 | | | |
| | Conv 1 × 1, 1, No. Classes | | | |

Instantiation of the PIDNet for semantic segmentation.

For operation, "OP, N, C" means operation OP with stride of N and the No. output channel is C; Output: output size given input size of 1024; mxRB: m residual basic blocks; 2xRBB: 2 residual bottleneck blocks; $OP_1$\$OP_2$: $OP_1$ is used for PIDNet-L while $OP_1$ is applied in PIDNet-M and PIDNet-S. (m,n,C) are scheduled to be (2,3,32), (2,3,64) and (3,4,64) for PIDNet-S, PIDNet-M and PIDNet-L, respectively.

## Models

For simple reproduction, we provide the ImageNet pretrained models here.

| Model (ImageNet) | PIDNet-S | PIDNet-M | PIDNet-L |
|---|---|---|---|
| Link | download | download | download |

Also, the finetuned models on Cityscapes and Camvid are available for direct application in road scene parsing.

| Model (Cityscapes) | Val (% mIOU) | Test (% mIOU) | FPS |
|---|---|---|---|
| PIDNet-S | 78.8 | 78.6 | 93.2 |
| PIDNet-M | 79.9 | 79.8 | 42.2 |
| PIDNet-L | 80.9 | 80.6 | 31.1 |

| Model (CamVid) | Val (% mIOU) | Test (% mIOU) | FPS |
|---|---|---|---|
| PIDNet-S | - | 80.1 | 153.7 |
| PIDNet-M | - | 82.0 | 85.6 |

## Prerequisites

This implementation is based on HRNet-Semantic-Segmentation. Please refer to their repository for installation and dataset preparation. The inference speed is tested on single RTX 3090 using the method introduced by SwiftNet. No third-party acceleration lib is used, so you can try TensorRT or other approaches for faster speed.

## Usage

### 0. Prepare the dataset

- Download the Cityscapes and CamVid datasets and unzip them in `data/cityscapes` and `data/camvid` dirs.
- Check if the paths contained in lists of `data/list` are correct for dataset images.

**:smiley_cat: Instruction for preparation of CamVid data (remains discussion) :smiley_cat:**

- Download the images and annotations from Kaggle, where the resolution of images is 960x720 (original);
- Unzip the data and put all the images and all the colored labels into `data/camvid/images/` and `data/camvid/labels`, respectively;
- Following the split of train, val and test sets used in SegNet-Tutorial, we have generated the dataset lists in `data/list/camvid/`;
- Finished!!! (We have open an issue for everyone who's interested in CamVid to discuss where to download the data and if the split in SegNet-Tutorial is correct. BTW, do not directly use the split in Kaggle, which is wrong and will lead to unnormal high accuracy. We have revised the CamVid content in the paper and you will see the correct results after its announcement.)

### 1. Training

- Download the ImageNet pretrained models and put them into `pretrained_models/imagenet/` dir.
- For example, train the PIDNet-S on Cityscapes with batch size of 12 on 2 GPUs:

```
python tools/train.py --cfg
configs/cityscapes/pidnet_small_cityscapes.yaml GPUS (0,1)
TRAIN.BATCH_SIZE_PER_GPU 6
```

- Or train the PIDNet-L on Cityscapes using train and val sets simultaneously with batch size of 12 on 4 GPUs:

```
python tools/train.py --cfg
configs/cityscapes/pidnet_large_cityscapes_trainval.yaml GPUS (0,1,2,3)
TRAIN.BATCH_SIZE_PER_GPU 3
```

### 2. Evaluation

- Download the finetuned models for Cityscapes and CamVid and put them into `pretrained_models/cityscapes/` and `pretrained_models/camvid/` dirs, respectively.
- For example, evaluate the PIDNet-S on Cityscapes val set:

```
python tools/eval.py --cfg configs/cityscapes/pidnet_small_cityscapes.yaml
\
```

```
                            TEST.MODEL_FILE
pretrained_models/cityscapes/PIDNet_S_Cityscapes_val.pt
```

- Or, evaluate the PIDNet-M on CamVid test set:

```
python tools/eval.py --cfg configs/camvid/pidnet_medium_camvid.yaml \
                            TEST.MODEL_FILE
pretrained_models/camvid/PIDNet_M_Camvid_Test.pt \
                            DATASET.TEST_SET list/camvid/test.lst
```

- Generate the testing results of PIDNet-L on Cityscapes test set:

```
python tools/eval.py --cfg
configs/cityscapes/pidnet_large_cityscapes_trainval.yaml \
                            TEST.MODEL_FILE
pretrained_models/cityscapes/PIDNet_L_Cityscapes_test.pt \
                            DATASET.TEST_SET list/cityscapes/test.lst
```

### 3. Speed Measurement

- Measure the inference speed of PIDNet-S for Cityscapes:

```
python models/speed/pidnet_speed.py --a 'pidnet-s' --c 19 --r 1024 2048
```

- Measure the inference speed of PIDNet-M for CamVid:

```
python models/speed/pidnet_speed.py --a 'pidnet-m' --c 11 --r 720 960
```

### 4. Custom Inputs

- Put all your images in `samples/` and then run the command below using Cityscapes pretrained PIDNet-L for image format of .png:

```
python tools/custom.py --a 'pidnet-l' --p
'../pretrained_models/cityscapes/PIDNet_L_Cityscapes_test.pt' --t '.png'
```

## Citation

If you think this implementation is useful for your work, please cite our paper:

```
```

```
@misc{xu2022pidnet,
      title={PIDNet: A Real-time Semantic Segmentation Network Inspired
from PID Controller},
      author={Jiacong Xu and Zixiang Xiong and Shankar P. Bhattacharyya},
      year={2022},
      eprint={2206.02066},
      archivePrefix={arXiv},
      primaryClass={cs.CV}
}
```

## Acknowledgement


- Our implementation is modified based on HRNet-Semantic-Segmentation.
- Latency measurement code is borrowed from the DDRNet.
- Thanks for their nice contribution.