Synergy between Semantic Segmentation and Image Denoising via Alternate Boosting

SHUNXIN XU, KE SUN, DONG LIU, ZHIWEI XIONG, and ZHENG-JUN ZHA,
University of Science and Technology of China, China

The capability of image semantic segmentation may be deteriorated due to the noisy input image, where image denoising prior to segmentation may help. Both image denoising and semantic segmentation have been developed significantly with the advance of deep learning. In this work, we are interested in the synergy between these two tasks by using a holistic deep model. We observe that not only denoising helps combat the drop of segmentation accuracy due to the noisy input, but also pixel-wise semantic information boosts the capability of denoising. We then propose a boosting network to perform denoising and segmentation alternately. The proposed network is composed of multiple segmentation and denoising blocks (SDBs), each of which estimates a semantic map and then uses the map to regularize denoising. Experimental results show that the denoised image quality is improved substantially and the segmentation accuracy is improved to close to that on clean images, and segmentation and denoising are both boosted as the number of SDBs increases. On the Cityscapes dataset, using three SDBs improves the denoising quality to 34.42 dB in PSNR, and the segmentation accuracy to 66.5 in mIoU, when the additive white Gaussian noise level is 50.

CCS Concepts: • Computing methodologies → Image segmentation; Image processing;

Additional Key Words and Phrases: Alternate boosting, deep learning, image denoising, semantic segmentation

ACM Reference format:
Shunxin Xu, Ke Sun, Dong Liu, Zhiwei Xiong, and Zheng-Jun Zha. 2023. Synergy between Semantic Segmentation and Image Denoising via Alternate Boosting. ACM Trans. Multimedia Comput. Commun. Appl. 19, 2, Article 69 (February 2023), 23 pages.
https://doi.org/10.1145/3548459

1 INTRODUCTION

Image semantic segmentation is a fundamental task in image understanding. The goal of semantic segmentation is to assign a category label for each pixel in an input image, which can be seen as pixel-level classification. Semantic segmentation has been actively studied in recent years because of its broad applications, such as augmented reality, autonomous driving, and satellite image

This work was supported by the Natural Science Foundation of China under Grants 62022075 and 62021001, and by the Fundamental Research Funds for the Central Universities under Grant WK349000006.

Authors’ address: S. Xu, K. Sun, D. Liu (corresponding author), Z. Xiong, and Z.-J. Zha, University of Science and Technology of China, No. 443 Huangshan Road, Hefei, Anhui, China, 230027; emails: {sxu, sunk}@mail.ustc.edu.cn, {dongeliu, zwxiong, zhazj}@ustc.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.
1551-6857/2023/02-ART69 $15.00
https://doi.org/10.1145/3548459
analysis. Recent methods [12, 32, 46, 64] were inspired by the groundbreaking work named **Fully Convolutional Network (FCN)** [38]. FCN discarded fully connected layers, which were adopted in previous approaches, and was able to deal with arbitrary resolution.

These methods studied semantic segmentation on clean images, but the capability of segmentation may be deteriorated due to the noisy input image. As we have observed in our experiments on the Cityscapes [13] dataset, the segmentation accuracy obtained on noisy images (training and testing are both on noisy images) is much lower than that on clean images (training and testing are both on clean images) by as high as 10% in mIoU, when the noise level (standard deviation of the additive Gaussian noise) is 50. Performing denoising prior to segmentation is a straightforward idea and is verified effective in our experiments. Note that image denoising, as one of the most classic problems in image processing, has been enhanced greatly by the powerful **convolutional neural network (CNN)** [40, 70, 72] in recent years. However, almost all of the existing denoising methods pursue visual quality of denoised image, with little concern of the utility of denoising for downstream tasks like segmentation. As the (seemingly) only exception, Liu et al. [35, 36] take into account the segmentation accuracy of denoised results when training their denoising network. However, it is observed in References [35, 36] that the denoising performance, especially in terms of PSNR, may worsen because of the additional consideration of segmentation.

Is it possible that the denoising performance can be improved with the help of semantic segmentation results? Let us consider a classic denoising method, BM3D [14], where non-local correlation is used to enhance denoising ability via collaborative processing of similar image content. Given a semantic segmentation map, the image content similarity may be better identified, and the non-local correlation may be better exploited.

With the aforementioned twofold motivations, we are interested in the synergy between image denoising and semantic segmentation by using a holistic deep model. Since the denoised image can be better segmented and segmentation result can assist in denoising, we propose a boosting method to perform the two tasks alternately. Specifically, we propose a **segmentation and denoising alternate boosting network (SDABN)**. SDABN consists of multiple **segmentation and denoising blocks (SDBs)**, each of which estimates a segmentation probability map from its input image and then uses the map to help image denoising. The output of one SDB is taken as the input to the next SDB, making a cascade that resembles boosting. Note that boosting has been proposed to improve image denoising [8, 9] and classification [19, 41]. But our network is different as we perform two tasks—image denoising and semantic segmentation—alternately. We verify that the alternate boosting idea improves the performance of both denoising and segmentation.

In summary, this work has made the following contributions:

- We propose a network known as segmentation and denoising alternate boosting network (SDABN), which performs segmentation and denoising alternately to enhance the performance of the two tasks.
- We propose the segmentation and denoising block (SDB), which is the first work that reports signal fidelity improvement (measured by PSNR and SSIM) of denoising via semantic segmentation, to the best of our knowledge.
- We conduct experiments on the Cityscapes [13] and OutdoorSeg [60] datasets with simulated noise and investigate the generalization ability to process real-world noisy images. Experimental results demonstrate the efficiency of our method. We also have interesting observations about the boosting effect.

Our code and models are publicly available at: https://github.com/powder21/SDABN.
Synergy between Semantic Segmentation and Image Denoising via Alternate Boosting

Fig. 1. (a) Denoising followed by segmentation: denoised results from $D_{\text{base}}$ are input to the segmentation module $S$. (b) Segmentation followed by denoising: $S$ segments the noisy images and provides the segmentation results as additional inputs to the denoising module $D$ (see Section 3.2 for more details). (c) The proposed segmentation and denoising alternate boosting network (SDABN), which is concatenating a series of segmentation and denoising blocks (SDBs). $y$, $s$, $\hat{x}$ stand for noisy image, segmentation (probability) map, and denoised image, respectively.

2 RELATED WORK

2.1 Semantic Segmentation

Recently, researchers have great interest in semantic segmentation. **Fully Convolutional Network (FCN)** [38] first discarded the fully connected layers and adopted the convolution layers throughout. Because of this, FCN can deal with arbitrary resolution images and becomes the most popular network architecture in semantic segmentation. To increase the receptive field, some methods [12, 65] adopted the dilated convolution layers in FCN. Later, most network structure for semantic segmentation can be regarded as encoder-decoder [6, 32, 46, 56, 74]. Encoder reduces the resolution of the feature maps, which can enlarge the receptive field, while decoder upsamples the feature maps to restore the resolution. The dilemma of encoder-decoder-based networks is that downsampling in encoder loses the details of the high-resolution feature maps, which influences the accuracy of segmentation. To address this problem, U-net [46] introduced skip-connections to provide the decoder with feature maps of the encoder with the same resolution. SegNet [6] recorded the location of the maximum response in max pooling, so it can restore the position of the maximum response in unpooling. RefineNet [32] restored the high-resolution feature maps by fusing the feature maps with different resolutions generated in the process of encoding. HRNet [56] maintained the high-resolution feature maps with a plain convolutional network and fused different resolution feature maps with concatenation. Yuan et al. proposed to characterize each pixel according to the corresponding object class [66]. Some more recent studies [37, 51, 59, 75] exploited Transformers [54] for image semantic segmentation. All these methods consider semantic segmentation on clean images and may have a problem in dealing with noisy input.

2.2 Image Denoising

Image denoising is one of the most fundamental problems in image processing and has been studied for several decades. Dictionary-based methods [15, 16, 39] solved the denoising problem by learning the sparse dictionary from clean images and coding the noisy images with the dictionary. BM3D [14], grouping similar patches for each location and filtering the groups collaboratively, is one of the milestones in denoising algorithm. Later, due to the development of deep learning, image denoising has made great progress in recent years. Zhang et al. [70] proposed DnCNN, which
adopted residual learning [21] rather than learning the pair relation between noisy and cleaning images directly. Batch normalization [23] also played a key role in their network. After that, some methods [3, 9, 10, 68] proposed different networks for denoising, which concentrated on the structure of networks. More recently, some studies [31, 61, 67] adopted Transformers [54] for image denoising. Besides, Lehtinen et al. proposed Noise2Noise [30], which can train a denoising network with paired noisy images whose identity is the same. In Noise2Void [27], the authors designed the blind-spot network to train the denoiser by self-supervised learning. Reference [53] trained their denoising network by mapping a random vector to the target noisy image, and the network will output the denoised image before convergence. Boosting, an algorithm for improving the performance of various tasks by cascading the same models, has been adopted in image denoising not only in the traditional way [8, 52] but also with CNN-based approach [9]. All the aforementioned methods concern the signal-to-noise ratio of denoised images, but high signal-to-noise ratio does not necessarily lead to high performance in the downstreaming tasks, like segmentation. In addition, it was never reported that signal quality of denoised images can be improved by using segmentation results, to our best knowledge.

2.3 Collaboration between Segmentation and Denoising

Liu et al. have proposed a framework, named Denoising Meeting Segmentation (DMS) [35, 36], taking into account denoising and segmentation at the same time. Since Reference [35] is the improved version of Reference [36], DMS mentioned in the following refers to Reference [35].

The pipeline of DMS is a cascade of denoising and segmentation, where the segmentation module is fixed and pretrained on clean images. Only the denoising module is updated according to a joint loss, which is composed of three terms: a signal reconstruction loss, a perceptual loss, and a segmentation loss. The signal reconstruction loss is Mean Squared Error (MSE), which is a common term used in supervised denoising method. The perceptual loss measures the feature distance, which was proposed by Johnson et al. [24] for style transfer and super-resolution, and the feature maps are from the VGG [49] backbone in the segmentation module. The segmentation loss is the cross-entropy between segmentation results of denoised images and segmentation labels.

Our method differs from DMS in the following aspects: First, we optimize the segmentation module to adapt to the noisy images, but in DMS the segmentation module is fixed and pretrained on clean images. Second, we utilize segmentation by conditional normalization to help denoising, but DMS uses the losses related to segmentation to update denoising module via back-propagation. Third, we optimize denoising module with only MSE, but DMS uses a joint loss, and the perceptual and segmentation losses are the regularization of MSE. So, our method still targets the signal reconstruction of clean images, but DMS not. The quantitative denoising performance of DMS drops because of the additional losses. Fourth, we use an alternate boosting framework, whereas DMS uses a multi-task one.

Moreover, Wang et al. proposed an unsupervised multi-task framework dealing with image denoising without knowing semantic segmentation labels [58]. It is different from our work, since we use ground-truth semantic segmentation labels for supervised learning. A number of studies have been conducted for joint denoising and segmentation on microscopy images [2, 7] or medical images [28, 29, 55, 62, 63]. In this work, we focus on natural images. There had been some non-learning-based methods for image denoising using image segmentation [22, 50], but that image segmentation had no semantic relation.

2.4 Collaboration between Other High-level and Low-level Vision Tasks

High-level tasks, such as classification and semantic segmentation, are closer to image understanding, aiming to explore the semantic information of images. Low-level tasks, such as image
denoising and super-resolution, are closer to image processing, targeting the restoration of degraded images. Besides DMS [35], there were also some studies on the collaboration between other high-level and low-level tasks. References [4, 44] brought the class prior to image denoising and needed to train a specific denoiser for each image class. Reference [34] adopted denoising to address the task of noisy image classification. In video deblurring, Reference [45] estimated the segmentation probability map to help the extraction of optical flow first and improved deblurring via the segmentation prior and the better optical flow. To improve classification, Reference [48] enhanced the target image and Reference [69] increased the resolution of the input image, and Reference [56] restored the degraded images to clean images in the feature domain of a classifier. The work above focused on the performance of only one task. In References [57, 73], the authors adopted multi-task frameworks to combine semantic segmentation and other tasks (e.g., depth estimation and super-resolution). Different from these studies, our work adopts an alternate boosting framework. The framework is studied for denoising and segmentation and may be extended to other high-level and low-level tasks.

3 PROPOSED METHOD

In Section 3.1, we first detail the exploration studies about segmentation and denoising. Second, we explain the detail of the basic unit named SDB in Section 3.2. Then, we introduce the proposed segmentation and denoising alternate boosting network (SDABN) in Section 3.3. Finally, we present the details of our training process in Section 3.4.

3.1 Exploration Studies

In this section, the observations are all on Cityscapes [13] and noisy data is with additive white Gaussian noise (AWGN).

It is intuitive that the performance of a segmentation network \( S \), trained on clean images, will drop when tested on noisy images. We verify this intuition. The results are illustrated in Figure 2, and we can observe that the drop becomes larger as the noise level increases. One may easily recall that the domain gap between training data and testing data may cause the drop. To exclude the domain gap, in the following, when we train a model with clean (or noisy, or denoised) images, we test it on clean (or noisy, or denoised) images, too. We conduct the following experiments when the noise level is 50, and the related results are in Table 1.
Table 1. Results of the Explorative Studies on Cityscapes (cf. Section 3.1)

| Denoising for segmentation | mIoU/Pix. Acc./Mean Acc. |
|----------------------------|-------------------------|
| S on clean images          | 76.2/96.0/84.5          |
| S on noisy images          | 63.9/93.4/74.0          |
| S on denoised images       | 64.7/93.8/74.7          |

| Segmentation for denoising | PSNR/SSIM              |
|----------------------------|------------------------|
| $D_{\text{base}}$ (without condition) | 34.16/0.9082          |
| $D$ with segmentation results ($SD$) | **34.23/0.9100**     |
| $D$ with noisy images ($ImgD$)       | 34.09/0.9072          |
| $D$ with constant maps ($CstD$)       | 34.09/0.9074          |

$D_{\text{base}}$ is direct denoising. $D$ accepts an extra condition as input to help denoising (Figure 1(a)).

**Observation 1 (Denoising Helps Segmentation).** We train a segmentation network $S$ on clean images, and the testing mIoU/pixel accuracy (Pix. Acc.)/mean accuracy (Mean Acc.) on clean images is 76.2/96.0/84.5. However, if we replace the training data with the corresponding noisy images and train a new segmentation module, then its testing result on noisy images drops to 63.9/93.4/74.0. Because the denoising method can reduce the noise, it is intuitive to adopt a denoiser $D_{\text{base}}$ before semantic segmentation to make up the performance drop from clean images to noisy images. As shown in Figure 1(a), we train one denoising network $D_{\text{base}}$ on Cityscapes, too, and train another segmentation network $S$ on denoised images. In this situation, the testing result on denoised images increases to 64.7/93.8/74.7. The comparison presents that noise has negative effects on semantic segmentation and denoising can reduce such effects. These results are reasonable, because noise disturbs the intensity of each pixel, which also damages the semantic information of the image and worsens the performance of segmentation methods. Therefore, denoising, as a technique for noise reduction, can help the segmentation module on noisy images.

**Observation 2 (Segmentation Helps Denoising).** Testing PSNR/SSIM of the denoiser $D_{\text{base}}$ mentioned just now is 34.16/0.9082. Then, we try to utilize semantic segmentation results to help denoising. We plug a new segmentation module $S$ for one denoising network, and this combination is as our basic unit SDB, denoted as $SD$. The denoiser accepts a segmentation condition estimated by $S$ and is denoted as $D$, which is shown in Figure 1(b). We built the segmentation-aware denoiser $D$ based on $D_{\text{base}}$, and the denoising result of $D$ increases to 34.23/0.9100. (Layers of the denoising network are normalized according to the segmentation condition, and more details will be introduced in Section 3.2.) However, more learnable parameters are brought to $D$ to utilize the segmentation condition, compared with $D_{\text{base}}$. To validate where the performance gain originates from, we replace the segmentation condition with the noisy images to train (and test) $D$, denoted as $ImgD$, which means the noisy images are inputted as not only denoising target but also condition. Besides, we replace the segmentation condition with a constant map, which is filled with constant 1, and denote it as $CstD$. $ImgD$ and $CstD$ are with the same parameter number as $SD$ in its denoising module but get worse denoising results (see Section 4.8 for further discussion). The results of these two indicate that the gain of $SD$ comes from the segmentation prior rather than the additional learnable parameters. Consequently, the experiments verify that the estimated segmentation probability map can indeed improve the performance of the denoiser.

### 3.2 Segmentation and Denoising Block (SDB)

For the denoising task, we define the clean image as $x \in \mathbb{R}^{H \times W \times C}$, where $C$ stands for the channel number of the image. Unless noted otherwise, we adopt RGB images in this article, so $C$ is equal to
3. Then the noisy image \( y \) can be formed as \( y = V(x) \), where \( V \) stands for the degradation from the clean image to the noisy image. Finally, with a denoising module \( D_{base} \), the denoising result \( \hat{x} \) can be formulated as \( \hat{x} = D_{base}(y) \).

In SDB, the noisy image \( y \) first goes through a semantic segmentation module \( S \), and the estimated segmentation probability maps can be formed by \( s = S(y) \), where \( s \in \mathbb{R}^{H \times W \times N} \) and \( N \) is the number of the categories in the dataset. We denote the segmentation-aware denoiser as \( D \), whose backbone is the common denoiser \( D_{base} \). SDB can be formulated as:

\[
\begin{align*}
\{ & s = S(y), \\
& \hat{x} = D(y, s) \}.
\end{align*}
\]

To utilize the estimated segmentation probability maps \( s \) and build our segmentation-aware denoiser \( D \), we insert Spatial Feature Transform (SFT) layers \([60]\) to the denoising backbone \( D_{base} \) (see Figure 3(a)). SFT is a kind of conditional normalization, which transforms some feature maps according to the given condition \( s \). We illustrate SFT in Figure 3(b). Specifically, there are two inputs: One is a feature maps \( F \), and the other is the segmentation probability maps \( s \) as the condition. The outputs of SFT are the normalized feature maps \( F' = (\gamma \otimes 1) \odot F \oplus \beta \), where \( \otimes \) and \( \oplus \) denote the element-wise multiplication and addition, respectively, and \( (\gamma, \beta) \) are normalization coefficients mapped from \( s \) through convolutional layers. We insert SFT after each layer of \( D_{base} \) except the last one.

To our best knowledge, we are the first to apply conditional normalization to denoising. Our idea of adopting this approach to utilize the segmentation maps is inspired by the denoising method BM3D \([14]\), where non-local correlation is used to enhance denoising ability via collaborative processing of similar image content. In SDB, feature maps of the denoising network are normalized based on the segmentation maps, which provide non-local consistency of the category information for the area belonging to the same category.

### 3.3 Segmentation and Denoising Alternate Boosting Network (SDABN)

From the exploration studies in Section 3.1, the denoised image can be better segmented and the segmentation result can assist in denoising, so we propose to perform the two tasks alternately. We construct our segmentation and denoising alternate boosting network (SDABN) and show it in Figure 1(c), which is a cascade of several SDBs. Based on Equation (1), the formula of SDABN
can be formulated iteratively:
\[
\begin{align*}
    s_1 &= S_1(y), \\
    \hat{x}_1 &= D_1(y, s_1), \\
    s_i &= S_i(\hat{x}_{i-1}), \\
    \hat{x}_i &= D_i(\hat{x}_{i-1}, s_i, y),
\end{align*}
\]
where \( i \) indicates the different SDB and \( n \) stands for the number of SDBs in total.

In SDABN, each \( D_i \) will be helped by the paired \( S_i \) and also provides cleaner images to assist \( S_{i+1} \) in the next segmentation-aware denoiser \( S D_{i+1} \), where \( S_{i+1} \) estimates segmentation maps to improve \( D_{i+1} \) further, here \( i \geq 1 \). Ignoring the segmentation part of each SDB, our SDABN can be regarded as a cascaded denoiser, whose performance is boosted as the number of denoising modules increases [9]. In Figure 1(c), there are skip connections between the noisy image and inputs of each SDB. Chen et al. [9] claimed that these connections are helpful for the boosting of image denoising.

As a result, the denoising capacity in SDABN is improved by not only segmentation prior but also the former denoising results. As for segmentation, its performance is improved by denoising, which has been shown in Section 3.1. Besides, we will verify that segmentation also helps the later segmentation modules in Section 4.5, which is bridged by the middle denoiser.

### 3.4 Training Strategy

In this article, we require noisy images, clean images, and segmentation labels for training. Note that only noisy images are required in the inference stage. To generate the training data, we follow the common settings of the supervised learning-based denoising research, i.e., collecting some clean images and producing their noisy version by introducing simulated noise, such as AWGN. AWGN had been used in most of the previous work, such as References [9, 70, 72], especially in DMS [35], which is the most related work to ours. Some recent studies have shown that the denoising models trained with AWGN may not generalize well on real-world noisy images [43]. One possible solution is using the heteroscedastic Gaussian noise instead of AWGN [18, 25]. No matter using which kind of simulated noise, as long as we have some well-labeled data including clean images and their segmentation maps, such as Cityscapes [13] and OutdoorSeg [60], we can obtain noisy images by introducing AWGN or the heteroscedastic Gaussian noise.

We train SDABN progressively in the order of \( S_1 \rightarrow D_1 \rightarrow S_2 \rightarrow D_2 \rightarrow \ldots \) All the trained modules are fixed when we train the next one. For training the denoising module \( D_i \), the loss function \( L_{D_i} \) is the **mean square error (MSE)** between the output of the current denoiser \( \hat{x}_i \) and the clean image \( x \). The loss function for semantic segmentation module \( S_i \) is the cross-entropy loss \( L_{S_i} \) between the estimated probability maps \( s_i \) and the segmentation label \( l \). Each module is trained with only one loss, so there is no need to consider weighing different losses, like in Reference [35], in our SDABN. The training strategy is shown in Algorithm 1.

### 4 EXPERIMENTS

In this section, \( S_1 D_1 S_2 D_2 \ldots S_n D_n \) denotes one SDABN composed of \( n \) SDBs, and \( S_1 D_1 S_2 D_2 \ldots S_n D_n \) stands for \( S_1 D_1 S_2 D_2 \ldots S_n D_n \) discarding the last denoising module. Considering the trade-off between the computational cost and the boosted performance, \( n \) is at the most 3 in this article.

#### 4.1 Experimental Settings

Unless otherwise noted, all the models in this section are trained and tested on noisy images. In most of this section, we generate the noisy images by adding AWGN to clean images. But in Sections 4.9 and 4.10, we use Poisson noise and heteroscedastic Gaussian noise, respectively, to investigate the generalization ability.
**ALGORITHM 1:** Training SDABN

**Input:** noisy images $y$, clean images $x$ and segmentation labels $l$.

1. **while** not converged **do**
2. Sample minibatch of paired clean images $x$ and segmentation labels $l$ from dataset;
3. Optimize the parameters of $S_{\text{clean}}$ (segmentation network trained on clean images) with $L_{S_{\text{clean}}}$;
4. **end while**
5. **for** $i = 1, 2, 3, \ldots, n$ **do**
6. **if** $i = 1$ **then**
7. Initialize the parameters of $S_1$ by those of $S_{\text{clean}}$;
8. **else**
9. Initialize the parameters of $S_i$ by those of $S_{i-1}$;
10. **end if**
11. **while** not converged **do**
12. Sample minibatch of paired noisy images $y$ and segmentation labels $l$ from dataset;
13. Optimize the parameters of $S_i$ with $L_{S_i}$;
14. **end while**
15. **while** not converged **do**
16. Sample minibatch of paired noisy images $y$ and clean images $x$ from dataset;
17. Optimize the parameters of $D_i$ with $L_{D_i}$;
18. **end while**
19. **end for**

Note that we propose performing segmentation and denoising alternately to enhance the performance of the two tasks. We do not intend to design an advanced segmentation module or an advanced denoising module. Instead, our method is generic to encompass any segmentation or denoising networks. Thus, the basic modules of segmentation and denoising are borrowed from the previous work and can be replaced by any two networks for segmentation and denoising, respectively. In this article, we adopt HRNetV2-W18-Small-v2 [56] and DDFN×1 [9] as the segmentation module ($S_i$) and denoising backbone ($D_{\text{base}}$), respectively, which both achieve the state-of-the-art performance in their tasks. DDFN×$n$ represents a cascade of $n$ DDFNs, which is the boosted version in Reference [9].

We train our network on two datasets: Cityscapes [13] and OutdoorSeg [60]. Cityscapes is a segmentation dataset of street scenes recorded in 50 different cities, where the number of categories is 19. We follow the original splits: The dataset is divided into a training set with 2,975 images and a validation set with 500 images, where the validation set is used for testing in our experiments. OutdoorSeg is a merged dataset for outdoor scene segmentation, which collects 9,900 outdoor images from ADE dataset [76], COCO dataset [33], and Flickr website. The number of categories is 8. In this article, we divide these 9,900 images into 8,800 and 1,100 as training and testing sets, respectively. We set the hyperparameters of our training process empirically and list them in Table 2.

We adopt PSNR and SSIM as the metrics for denoising and compute mIoU, **pixel accuracy** (Pix. Acc.) and **mean accuracy** (Mean Acc.) for segmentation. For these five metrics, larger is better.

### 4.2 Comparison with Other Methods

We compare the performance of image denoising and semantic segmentation of different methods on Cityscapes [13] and OutdoorSeg [60], and the results are shown in Table 3. As weak baselines, we adopt some existing denoisers and cascade each denoiser and a segmentation module pretrained...
Table 2. Hyperparameters of the Training on Cityscapes [13] and OutdoorSeg [60], Where Power Is Used in the Poly Learning Rate Policy

| Module | Dataset     | Batch Size | Learning Rate | Power | Epoch | Optimizer | Momentum | Weight Decay |
|--------|-------------|------------|---------------|-------|-------|-----------|----------|--------------|
| $S_i$  | Cityscapes  | 40         | 0.03          | 0.9   | 484   | SGD       | 0.9      | 0.0005       |
| $S_i$  | OutdoorSeg  | 48         | 0.004         | 1.5   | 2,000 | Adam [26] | 0.9      | 0.0005       |
| $D_i$  | Cityscapes  | 64         | 0.001         | 1.5   | 700   | Adam [26] | 0.9      | 0.0005       |
| $D_i$  | OutdoorSeg  | 32         | 0.001         | 1.5   | 700   | Adam [26] | 0.9      | 0.0005       |

Table 3. Quantitative Comparison Results of Denoising and Segmentation

| Dataset  | $\sigma$ | Metric | BM3D+Seg. | DnCNN+Seg. | FFDNet+Seg. | IRCNN+Seg. | DDFN×1+Seg. | DMS | DMS* | SDB |
|----------|----------|--------|-----------|------------|-------------|------------|-------------|-----|-----|-----|
| Cityscapes | 50 | PSNR | 33.68 | 33.47 | 33.97 | 33.44 | 34.16 | 34.03 | 33.74 | 34.23 |
|          |         | SSIM  | 0.8953 | 0.8916 | 0.9049 | 0.8913 | 0.9082 | 0.9032 | 0.9005 | 0.9100 |
|          |         | mIoU  | 39.2   | 34.3   | 31.7   | 33.7   | 40.3   | 44.6   | 51.9   | 63.9  |
|          |         | Pix. Acc. | 69.6 | 65.0 | 60.7 | 65.6 | 72.5 | 85.9 | 88.6 | 93.4 |
|          |         | Mean Acc. | 56.9 | 49.2 | 45.4 | 49.6 | 59.2 | 53.7 | 61.4 | 74.0 |
| OutdoorSeg | 50 | PSNR | 27.52 | 27.82 | 27.92 | 27.82 | 27.97 | 27.98 | 27.93 | 28.01 |
|          |         | SSIM  | 0.7875 | 0.8026 | 0.8054 | 0.8060 | 0.8106 | 0.8080 | 0.8063 | 0.8115 |
|          |         | mIoU  | 74.1   | 73.2   | 70.4   | 73.7   | 72.8   | 71.9   | 73.0   | 75.2  |
|          |         | Pix. Acc. | 87.4 | 86.8 | 85.3 | 87.1 | 86.3 | 86.4 | 87.9 | 88.3 |
|          |         | Mean Acc. | 81.4 | 80.3 | 80.2 | 80.6 | 80.6 | 72.0 | 77.9 | 82.3 |
| OutdoorSeg | 50 | PSNR | 31.02 | 31.14 | 31.20 | 31.15 | 31.15 | 31.35 | 31.22 | 31.39 |
|          |         | SSIM  | 0.8880 | 0.8927 | 0.8938 | 0.8934 | 0.8966 | 0.8956 | 0.8925 | 0.8973 |
|          |         | mIoU  | 76.9   | 76.9   | 76.3   | 77.0   | 76.4   | 75.8   | 77.5   | 77.4  |
|          |         | Pix. Acc. | 88.9 | 88.8 | 88.4 | 88.9 | 88.3 | 88.6 | 89.2 | 89.3 |
|          |         | Mean Acc. | 86.0 | 86.0 | 85.1 | 85.1 | 85.2 | 86.6 | 86.7 | 86.9 |

BM3D [14]+Seg., DnCNN [70]+Seg., FFDNet [72]+Seg., IRCNN [71]+Seg., and DDFN×1 [9]+Seg. stand for different denoisers followed by the segmentation module trained on clean images, respectively. DMS [35] is a joint denoising-segmentation method, and DMS* replaces its segmentation and denoising backbones with the corresponding ones used in our SDB to provide a more fair comparison.

Table 3. Quantitative Comparison Results of Denoising and Segmentation

on clean images directly. These baselines are denoted as DN+Seg., where DN is selected from BM3D [14], DnCNN [70], FFDNet [72], IRCNN [71], and DDFN×1 [9], and Seg. refers to HRNetV2-W18-Small-v2 [56]. BM3D is a traditional denoising method, and DnCNN, FFDNet, IRCNN, and DDFN×1 are four CNN-based denoising methods. For these five denoisers, we use the code released by the authors. Almost all of the existing segmentation methods were tested on clean images. Their performance on noisy images is severely degraded, as mentioned in Reference [35] and observed in Figure 2. Thus, it is unfair to compare with these segmentation methods. The only exception is DMS [35], which takes into account denoising and segmentation in one framework, so DMS is the most related work to ours. In Section 2.3, we have introduced DMS and compared it with our method in conceptual aspects. Here, we will compare them in performance. We reproduce DMS on our PyTorch [42] environment.

In DMS, the segmentation module is DeepLab-LargeFOV, proposed in Reference [11], and the denoiser is a multi-scale network designed by the authors. To present more comparable results, we revise DMS to DMS*, i.e., we replace the segmentation and denoising modules in DMS with...
Table 4. Denoising Results of One SDB with Different Backbones

| Backbone | DDFN×1 | DDFN×2 | DDFN×3 |
|----------|--------|--------|--------|
| PSNR     | 34.23  | 34.34  | 34.41  |
| SSIM     | 0.9100 | 0.9115 | 0.9121 |

DDFN×1 [9] is the denoising backbone used by default, but it can be boosted by cascading, e.g., DDFN×2 and DDFN×3.

HRNetV2-W18-Small-v2 and DDFN×1, respectively. Note that DMS* has exactly the same segmentation and denoising network structures as our SDB. We will discuss more about DMS and DMS* in Section 4.7.

In Table 3, each DN+Seg. inherits the denoising performance of the corresponding DN, since the denoising results of each DN+Seg. are exactly those of the corresponding DN. As for DMS [35], its perceptual and segmentation losses deviate the optimization from signal quality. But in our method, the denoising module is trained with only MSE and benefits from the SFT layers, so it is always better than the backbone (DDFN×1). Overall, the denoising performance of SDB is almost always better than these compared methods. Furthermore, SDB will be improved if we adopt stronger backbones, such as DDFN×2 and DDFN×3, and the results are shown in Table 4. The denoising performance of SDB depends entirely on the adopted backbone, so we can apply a strong enough backbone if SDB is compared with a strong baseline about denoising. We want to claim that our framework can improve the denoising results quantitatively with the help of segmentation, but that of DMS will drop because of considering segmentation. The contribution here is the approach of utilizing the segmentation to improve denoising, which overcomes the drawback of DMS.

As for segmentation, our method is also superior to the compared methods. The segmentation modules in the other methods are all fixed and pretrained on clean images. Even though DMS adopts the segmentation loss as guidance, our method still outperforms it, because we optimize the segmentation module for the noisy inputs. Since the network structures are the same, the comparison between DMS* and SDB demonstrates the gain is from our better framework rather than the different network structures.

In Table 3, we list the results of SDB to compare with other methods, which is just a single basic unit. Our method can be further improved by alternate boosting, and the results will be shown in Section 4.4.

Additionally, the visual results in Figures 4 and 5 show that the denoised images of our method are cleaner and with more details of texture than others, which are also boosted as the unit number increases. In Figures 6 and 7, the segmentation results of DN+Seg. are with much false estimation. When compared with DMS, our method has better segmentation performance especially on small objects. The results of segmentation are also refined as the unit number increases.

Because of the requirement of segmentation labels for training, our method has some difficulty to be trained on the common denoising datasets like BSDS [5]. We design the following experiment to verify our method on BSDS. First, we train S1 on the OutdoorSeg training set (we choose OutdoorSeg instead of Cityscapes because the images in OutdoorSeg have more similar appearance to those in BSDS than the images in Cityscapes). Second, we divide the 500 color images in BSDS into the training set (432 images) and the test set (68 images), following the previous work [70]; we train D1 on the training set. Next, we perform test of SDB×1 on the test set. The test results are shown in Table 5. It can be observed that our SDB×1 outperforms its anchor DDFN×1 as well as DnCNN, which demonstrates the effectiveness of our method despite some domain mismatching in the conducted experiment.
Fig. 4. Denoising results of BM3D [14], DnCNN [70], DDFN×1 [9], DMS [35], and our method (SDB×1 and ×3) with noise level $\sigma = 50$. The original images come from Cityscapes [13].

Table 5. Average PSNR (dB) Results of BM3D [14], DnCNN [70], DDFN×1 [9], and our SDB×1 on the BSD68 Test Set

| Noise Level | BM3D | DnCNN | DDFN×1 | SDB×1 |
|-------------|------|-------|--------|-------|
| 50          | 27.38| 27.92 | 27.93  | 27.96 |
| 25          | 30.71| 31.23 | 31.22  | 31.26 |

4.3 Comparison of Different Methods to Utilize the Estimated Segmentation Results for Denoising

In Table 1, we have verified that segmentation can help denoising with SFT layers [60], and the gain of denoising is from segmentation prior rather than the additional trainable parameters of SFT.

In fact, we have tried different approaches to utilize the estimated segmentation map; the comparison is listed in Table 5. A naive approach is concatenating the noisy image $y \in \mathbb{R}^{H \times W \times 3}$ and segmentation map $s \in \mathbb{R}^{H \times W \times N}$ as the new input $y' \in \mathbb{R}^{H \times W \times (3+N)}$ of the denoising backbone $D_{\text{base}}$, e.g., DDFN×1, which is denoted as "baseline+concatenate seg. map." The performance is even worse than that of the baseline. Since the number of categories $N$ is much larger than 3, e.g., $N$ is 19 in Cityscapes dataset, the majority of the input $y'$ is not image, which may disturb the denoiser to catch the signal of image and worsen the performance of denoising. To overcome this influence, we replace the segmentation map $s$ with its estimated category layout $s' = \text{arg max}_i s(i)$, where $i$ is the index along the channel and $s' \in \mathbb{R}^{H \times W \times 1}$. Again, we concatenate the noisy image $y$ and the estimated category layout $s'$ to build the input $y'' \in \mathbb{R}^{H \times W \times (3+1)}$. We denote it as "baseline + concatenate category" in Table 5. We can observe that "baseline + concatenate category" is better than the common baseline slightly, but cannot reach the performance of "baseline + SFT."
Synergy between Semantic Segmentation and Image Denoising via Alternate Boosting

Fig. 5. Denoising results of BM3D [14], DnCNN [70], DDFN×1 [9], DMS [35], and our method (SDB×1 and ×3) with noise level $\sigma = 50$. The original images come from OutdoorSeg [60].

Table 6. Comparison of Different Ways Utilizing the Segmentation Information

| Method                                      | PSNR   | SSIM  |
|---------------------------------------------|--------|-------|
| Baseline (DDFN×1)                          | 34.16  | 0.9082|
| Baseline + concatenate seg. map             | 34.08  | 0.9084|
| Baseline + concatenate category             | 34.18  | 0.9090|
| Baseline + SFT                             | **34.23** | **0.9100** |

Baseline + concatenate seg. map and baseline + concatenate category refer to concatenating segmentation map and estimated category layout, respectively, to noisy images as input of the baseline. Baseline + SFT is the one adopted in our SDB.

The experiments in Table 6 verify the segmentation information can help denoising by two different methods, and the method based on SFT is superior to the other naive methods mentioned in Table 6, which means it utilizes the segmentation information more sufficiently in the denoising module.
Fig. 6. Segmentation results of BM3D [19] + Seg., DnCNN [70] + Seg., DDFN×1 [9] + Seg., DMS [35], and our method (SDB×1 and ×3) with noise level \( \sigma = 50 \). The original images come from Cityscapes [13].

### Table 7. Boosting Results of DDFNs [9] and Our SDBs for Denoising

| Dataset   | \( \sigma \) | #Units | DDFN          | SDB     |
|-----------|--------------|--------|---------------|---------|
|           |              |        | PSNR | SSIM | PSNR | SSIM |
| Cityscapes| 50           | ×1     | 34.16 | 0.9082 | 34.23 | 0.9100 |
|           |              | ×2     | 34.28 | 0.9099 | 34.34 | 0.9115 |
|           |              | ×3     | 34.34 | 0.9108 | 34.42 | 0.9122 |
|           | 30           | ×1     | 36.18 | 0.9315 | 36.27 | 0.9328 |
|           |              | ×2     | 36.28 | 0.9323 | 36.39 | 0.9340 |
|           |              | ×3     | 36.34 | 0.9330 | 36.50 | 0.9351 |
|           | 10           | ×1     | 40.75 | 0.9668 | 40.83 | 0.9671 |
|           |              | ×2     | 40.84 | 0.9671 | 40.94 | 0.9677 |
|           |              | ×3     | 40.88 | 0.9672 | 41.06 | 0.9683 |
| OutdoorSeg| 50           | ×1     | 27.97 | 0.8106 | 28.01 | 0.8115 |
|           |              | ×2     | 28.06 | 0.8136 | 28.10 | 0.8152 |
|           |              | ×3     | 28.10 | 0.8147 | 28.14 | 0.8159 |
|           | 25           | ×1     | 31.35 | 0.8966 | 31.39 | 0.8973 |
|           |              | ×2     | 31.42 | 0.8988 | 31.47 | 0.8995 |
|           |              | ×3     | 31.43 | 0.8990 | 31.50 | 0.9002 |

The rows with \( \sigma = 0 \) are the results of the segmentation module trained and tested on clean images.

### Table 8. Boosting Results of SDBs for Segmentation

| Dataset         | \( \sigma \) | #Units | mIoU | Pix. Acc. | Mean Acc. |
|-----------------|--------------|--------|------|-----------|------------|
| Cityscapes      | 50           | ×1     | 63.9 | 93.41     | 74.0       |
|                 |              | ×2     | 66.1 | 94.05     | 76.3       |
|                 |              | ×3     | 66.5 | 94.06     | 75.6       |
|                 | 30           | ×1     | 67.1 | 94.42     | 76.1       |
|                 |              | ×2     | 68.5 | 94.77     | 77.8       |
|                 |              | ×3     | 69.0 | 94.80     | 78.6       |
|                 | 10           | ×1     | 73.4 | 95.47     | 82.3       |
|                 |              | ×2     | 74.4 | 95.65     | 82.9       |
|                 |              | ×3     | 74.8 | 95.73     | 83.3       |
| OutdoorSeg      | 50           | ×1     | 75.2 | 88.25     | 85.8       |
|                 |              | ×2     | 76.4 | 88.5      | 86.0       |
|                 |              | ×3     | 76.7 | 89.11     | 86.5       |
|                 | 25           | ×1     | 77.4 | 89.32     | 86.9       |
|                 |              | ×2     | 78.0 | 89.57     | 87.4       |
|                 |              | ×3     | 78.2 | 89.68     | 87.5       |

4.4 Boosting Results

In this section, we present the boosting results of SDABN on Cityscapes and OutdoorSeg. The quantitative results of image denoising and semantic segmentation are shown in Tables 7 and 8, respectively. In Table 8, note that the segmentation results of SDB×1 are also those of the
Fig. 7. Segmentation results of BM3D\cite{14}\+Seg., DnCNN\cite{70}\+Seg., DDFN×1\cite{9}\+Seg., DMS\cite{35}, and our method (SDB×1 and ×3) within noise level $\sigma = 50$. The original images come from OutdoorSeg\cite{60}.

segmentation module, since SDABN is beginning with $S_1$. As reference, the rows with $\sigma = 0$ are the results of the segmentation module (trained and tested) on clean images.

In Table 7, we also list the denoising results of DDFNs\cite{9}, because DDFN is the backbone of our denoiser. Both DDFN and SDB can be cascaded with different numbers of basic units, which are denoted as ×1, ×2, and ×3. From the results of PSNR/SSIM, we can observe that SDBs always have better performance than DDFN when the numbers of basic units are the same. These results verify that segmentation can help denoising, which has been claimed in Section 3.1.

From the results of DDFNs, we know that denoising is boosted as the number of denoising modules increases. There is the same conclusion of SDBs if we consider SDABN as a cascade of segmentation-aware denoisers. Hence, the denoising capacity of SDABN is improved by not only segmentation prior but also the former denoising results.

Besides, the results in Table 8 verify that the performance of segmentation on noisy images is negatively correlated to the noise intensity in two ways. First, if we only consider the single segmentation module with the different noise levels, then we can find that the performance on the images with a lower noise level is closer to that on clean images. Second, because the denoisers provide cleaner and cleaner images as the number of units increases, the segmentation performance is boosted and closer to the results of clean images. The inputs of SDB×1 and SDB×2 are
Table 9. Ablation Study about Training Steps

|                | PSNR  | SSIM  | mIoU  | Pix. Acc. | Mean Acc. |
|----------------|-------|-------|-------|-----------|-----------|
| SDB×1          | 34.23 | 0.9100| 63.9  | 93.4      | 74.0      |
| SDB×1-2T       | 34.28 | 0.9106| 64.2  | 93.6      | 74.0      |
| SDB×2          | 34.34 | 0.9115| 66.1  | 94.1      | 76.3      |

SDB×1-2T denotes training SDB×1 with 2× number of steps.

Table 10. Ablation Study of Segmentation-aware Design for the Later Segmentation Module (cf. Section 4.5)

| Denoising | Seg.-Aware | PSNR  | SSIM  | Segmentation | mIoU  | Pix. Acc. | Mean Acc. |
|-----------|------------|-------|-------|--------------|-------|-----------|-----------|
| DDFN×1    | ×          | 34.157| 0.9082| DDFN×1 + S   | 64.7  | 93.8      | 74.7      |
| DDFN×2    | ×          | 34.280| 0.9099| DDFN×2 + S   | 65.7  | 93.9      | 75.6      |
| S₁D₁      | ✓          | 34.231| **0.9100**| S₁D₁ + S     | 66.1  | **94.1**  | **76.3**  |
| GtD₁      | ✓          | 34.234| 0.9099| GtD₁ + S     | 87.0  | 98.4      | 92.3      |

DDFN refers to [9]. GtD₁ is using ground truth of the segmentation map as the condition to D₁.

noisy images and denoised images, respectively, the comparison verifies that denoising helps the segmentation, which is the same as the observation in Section 3.1.

From the visual comparison between SDB×1 and SDB×3 in Figures 4–7, we can observe that the results of not only denoising but also segmentation are boosted as the unit number increases. The boosting reflected by the visual results is more significant than that reflected by the quantitative results listed in Tables 7 and 8.

It is worth noting that more SDBs require more training time, but the gain is not attributed to the increased training time. To demonstrate this, we train SDB×1 with 2× number of steps and compare with SDB×2 that is trained with the same number of steps. As shown in Table 9, the performance gain obtained by increasing training time is marginal.

4.5 Segmentation Helps Segmentation

Denoising results help the further denoising [9]. In this section, we will verify that segmentation results help the further segmentation, too. That is to verify that the segmentation-aware design is helpful for the later segmentation module. The related results are on Cityscapes with noise level 50 and shown in Table 10. We list four different denoisers in this table, where GtD₁ stands for D₁ trained (and tested) with ground truth of the segmentation as a condition, rather than the estimated one from S₁. Correspondingly, there are four segmentation modules, denoted as X + S, where X ∈ {DDFN×1, DDFN×2, S₁D₁, GtD₁}, and the inputs of X + S are from the denoiser X. Note that DDFN×1 is the denoising backbone of S₁D₁.

First, S₁D₁ is with the segmentation-aware design, but DDFN×1 not. From Table 10, S₁D₁ + S is better than DDFN×1 + S on segmentation. So, we claim that our segmentation-aware design is helpful for the further segmentation. In other words, the segmentation module S₁ is beneficial to the later S₁₊₁, which is bridged by the middle denoiser D₁.

Furthermore, we compare S₁D₁ with DDFN×2, which is a cascade of two DDFN×1. DDFN×2 is with better performance than S₁D₁ on the denoising metrics. But because of the segmentation-aware design in S₁D₁, S₁D₁ + S is better than DDFN×2 + S on the segmentation metrics. This observation demonstrates that the denoising results of the segmentation-aware denoiser for the following segmentation module may achieve higher segmentation accuracy, although its denoising performance is worse. Besides, better segmentation prior is more helpful to the following
4.6 Effect of Alternate Boosting for Segmentation

In this section, we will verify the necessity of the alternate boosting for segmentation on noisy images. From Table 8, we know that the segmentation performance is boosted as the basic unit number increases. But the learnable parameters of \( S_1D_1S_2 \) are more than those of \( S_1 \), so we wonder if \( S_1D_1S_2 \) is better than the common single segmentation networks with comparable number of parameters.

In SDABN, \( S_1 \) and \( S_2 \) are both trained with segmentation losses, and \( D_1 \) is trained with a signal reconstruction loss. We train another \( S_1D_1S_2 \) jointly only with a segmentation loss, denoted as \( S_1D_1S_2 \)-joint, which is a single segmentation network with the same structure as \( S_1D_1S_2 \). The results are on Cityscapes with noise level 50 and shown in Table 11. We can observe that \( S_1D_1S_2 \)-joint is inferior to \( S_1D_1S_2 \). Besides, \( S_1D_1S_2 \)-joint is even worse than \( S_1 \), which verifies the structure of \( S_1D_1S_2 \) is not suitable for segmentation without the losses for \( S_1 \) and \( D_1 \), and the gain from \( S_1 \) to \( S_1D_1S_2 \) is from the design of the alternate boosting rather than the more parameters.

Besides, we train a wider and deeper \( S_1 \) with segmentation loss, denoted as \( S_1+ \) in Table 11, whose parameter number is comparable to that of \( S_1D_1S_2 \). The comparison between \( S_1D_1S_2 \) and \( S_1+ \) on segmentation demonstrates that our alternate boosting network outperforms the state-of-the-art segmentation network on noisy images. In the meantime, our method can provide the denoised images, which are unavailable in other segmentation networks.

### 4.7 DMS Variants

In Table 3, we provide the results of both DMS and DMS*, where the latter is a modified version of DMS [35]. There are three differences between DMS and DMS* detailed in Table 12: denoising networks, segmentation networks (used to calculate the segmentation losses), and training losses. DMS* does not use the perceptual loss, because in DMS the perceptual loss is calculated by the
VGG backbone in DeepLab-LargeFOV, but DMS* does not have this network. These differences are intentionally made because DMS* is more aligned, than the original DMS, to our proposed SDB. Then, the comparison between DMS* and SDB is more informative.

We conduct more comparative studies on the DMS variants in Table 12. From the results, we observe that: (1) U-Net seems performing better than DDFN×1, but the number of parameters of U-Net is also larger than that of DDFN×1. (2) Using perceptual loss would harm PSNR/SSIM a little, but would benefit the segmentation metrics. (3) Using HRNetV2-W18-Small-v2 instead of DeepLab-LargeFOV would improve the segmentation results significantly. (4) If considering signal quality, then the results of DMS/DMS* are worse than the results of their variants that are trained with only MSE (denoted as DMS-MSE/DMS*-MSE); this is reasonable, because the perceptual and segmentation losses deviate the optimization from signal quality. Based on these observations, we think our method can be improved twofold in the future. First, we may replace DDFN×1 with advanced denoising networks; note that our method is generic to encompass any denoising or segmentation networks. Second, we may reintroduce the perceptual loss into the training of denoising networks.

4.8 Discussion on $\text{ImgD}$ and $\text{CstD}$

As shown in Table 1, $\text{ImgD}$ and $\text{CstD}$ have more parameters but perform worse than $D_{\text{base}}$. This is somewhat surprising.

In $\text{ImgD}$ (respectively, $\text{CstD}$), we use the noisy image (respectively, the constant) as input to the SFT layers to obtain $\gamma$ and $\beta$ to modulate the features in the denoising backbone (see Figure 3). If $\gamma$ and $\beta$ are learned to be 0, then the SFT layer is equivalent to the identical mapping, and $\text{ImgD}$ (respectively, $\text{CstD}$) shall perform as well as $D_{\text{base}}$. It appears a trivial solution that $\gamma$ and $\beta$ are learned to be 0, but from the experimental results, this trivial solution is not found by the network training process. Therefore, we believe the reason may be attributed to the network training. Comparing $\text{ImgD}$ and $\text{CstD}$, it is intuitive that the latter has a better chance to find the trivial solution, because the latter is not interfered by the noisy image when learning $\gamma$ and $\beta$. Then, we focus on $\text{CstD}$.

First, we conjecture that the trained $\gamma$’s and $\beta$’s are not 0. Our conjecture is confirmed by the results shown in Figure 8. Next, we want to ask ourselves why the network cannot learn $\gamma$ and $\beta$ to be 0. We find that the SFT layer would change the initial status of the network, which may cause the network to have difficulty in learning. Specifically, let us analyze the variance of the features after the SFT layer than those before the SFT layer. Given our implementation (in Figure 3), we have $F’ = (\gamma + 1)F + \beta = \alpha F + \beta$, where $\alpha = \gamma + 1$, and for simplicity of denotation, we consider scalars. After random initialization of the network parameters, we can regard $F$, $\alpha$, $\beta$ as random variables.
Table 13. Quantitative Results of Denoising on Poisson Noise, Compared with the Denoising Method DDFN [9]

| Dataset      | #Units | DDFN     | SDB       |
|--------------|--------|----------|-----------|
|              |        | PSNR/SSIM| PSNR/SSIM |
| Cityscapes   | ×1     | 40.83/0.9691 | 40.92/0.9695 |
|              | ×2     | 40.94/0.9695 | 41.15/0.9705 |
|              | ×3     | 41.05/0.9701 | 41.25/0.9710 |
| OutdoorSeg   | ×1     | 36.61/0.9628 | 36.64/0.9630 |
|              | ×2     | 36.71/0.9637 | 36.74/0.9640 |
|              | ×3     | 36.74/0.9640 | 36.77/0.9643 |

and safely assume that they are independent from each other. Thus, let $D(x)$ be the variance of $x$, $E(x)$ be the expectation of $x$, we have

$$D(\alpha F) = E[(\alpha F)^2] - E^2(\alpha F)$$

$$= E(\alpha^2)E(F^2) - E^2(\alpha)E^2(F)$$

$$= [D(\alpha) + E^2(\alpha)][D(F) + E^2(F)] - E^2(\alpha)E^2(F)$$

$$= D(\alpha)D(F) + D(\alpha)E^2(F) + D(F)E^2(\alpha).$$

If using proper initialization, like Kaiming’s, we have $E(\gamma) = 0$, so $E(\alpha) = 1$, then

$$D(\alpha F) = D(\alpha)D(F) + D(\alpha)E^2(F) + D(F) > D(F).$$

Then

$$D(F') = D(\alpha F + \beta) = D(\alpha F) + D(\beta) > D(\alpha F) > D(F).$$

That is to say, after inserting the SFT layers, the variance of the features would be increased. Since we have used multiple SFT layers, the feature variance would be larger and larger, which is reported harmful for training networks [17, 20, 47].

However, if we use a meaningful guidance, like a good segmentation map, as the input to the SFT layers, then the network can utilize the provided guidance and may overcome the negative effect of less proper initialization. This is evidenced by the results provided in Table 1.

We shall remark that the above analyses are qualitative, because the network training dynamics remain largely unexplored in the literature. In addition, there may be other unknown factors that cause the networks to have not been trained to reject the meaningless input to the SFT layers.

### 4.9 Poisson Noise-related Results

In the former experiments, we use AWGN to generate the noisy images. Here, we conduct a set of experiments using Poisson noise to validate the generalization ability. Different from AWGN, Poisson noise is non-additive and pixel-dependent, whose parameter $\lambda$ on each pixel is different and equal to the corresponding pixel value of clean images. From the related results in Table 13, we can observe that the denoising performance of SDB is still superior to that of the baseline DDFN [9], which confirms the generalization ability of our denoising method on Poisson noise.

### 4.10 Real-world Image Denoising Results

The existing real-world image denoising datasets, such as SIDD [1] and DND [43], do not have segmentation labels. However, we still try to use our method as much as possible on these datasets to provide some information in this subsection.
Table 14. Denoising Results of DDFN×1 [9] and Our SDB×1 on the SIDD [1] and DND [43] Datasets

| Dataset  | DDFN×1   | SDB×1    |
|----------|----------|----------|
|          | PSNR/SSIM | PSNR/SSIM |
| SIDD     | 38.37/0.9480 | **38.61/0.9503** |
| DND      | 38.92/0.9468 | **39.00/0.9487** |

The models are trained on Cityscapes with heteroscedastic Gaussian noise.

Fig. 9. Visual comparison of segmentation results before and after denoising. The left two examples are from SIDD [1] and the right two are from DND [43].

Since the heteroscedastic Gaussian noise is closer (than AWGN) to real-world noise, the denoisers trained with heteroscedastic Gaussian noise show good denoising results on real-world noisy images [18, 25]. Following Reference [25], we train SDB×1 and DDFN×1 on Cityscapes with the heteroscedastic Gaussian noise, and then fine-tune the denoising modules using the SIDD training set. Note that the segmentation module in our SDB×1 is not fine-tuned because there is no segmentation label for the SIDD training set. The evaluation results are shown in Table 14. We can observe that SDB×1 consistently outperforms DDFN×1 on the two tested datasets, which demonstrates the effectiveness of our method.

Since there is no segmentation label for real-world noisy image datasets, we have to perform comparisons of segmentation results subjectively. As shown in Figure 9, the denoised images are better segmented than the noisy images.

5 CONCLUSION

We have explored the synergy between image denoising and semantic segmentation by using a holistic deep model SDABN, which is a cascade of SDBs. From our experiments, we find denoising and segmentation are both boosted as the unit number increases. In the boosting process, segmentation improves denoising performance, and the denoiser helps increase segmentation accuracy. Moreover, denoising assists the later denoising, and segmentation is beneficial to the further segmentation.

REFERENCES

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. 2018. A high-quality denoising dataset for smartphone cameras. In CVPR. IEEE, 1692–1700.
Synergy between Semantic Segmentation and Image Denoising via Alternate Boosting

[2] Pedram Abdolghader, Andrew Ridsdale, Tassos Grammatikopoulos, Gavin Resch, Francois Legare, Albert Stolow, Adrian F. Pegoraro, and Isaac Tamblyn. 2021. Unsupervised hyperspectral stimulated Raman microscopy image enhancement: Denoising and segmentation via one-shot deep learning. arXiv preprint arXiv:2104.08338 (2021).

[3] Saeed Anwar and Nick Barnes. 2019. Real image denoising with feature attention. In ICCV. IEEE, 3155–3164.

[4] Saeed Anwar, Fatih Porikli, and Cong Phuoc Huynh. 2017. Category-specific object image denoising. IEEE Trans. Image Process. 26, 11 (2017), 5506–5518.

[5] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. 2010. Contour detection and hierarchical image segmentation. IEEE Trans. Patt. Anal. Mach. Intell. 33 (2010), 898–916.

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Patt. Anal. Mach. Intell. 39, 12 (2017), 2481–2495.

[7] Tim-Oliver Buchholz, Mangal Prakash, Deborah Schmidt, Alexander Krull, and Florian Jug. 2020. DenoSeg: Joint denoising and segmentation. In ECCV. Springer, 324–337.

[8] Michael R. Charest, Michael Elad, and Peyman Milanfar. 2006. A general iterative regularization framework for image denoising. In CISS. IEEE, 452–457.

[9] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. 2019. Real-world image denoising with deep boosting. IEEE Trans. Patt. Anal. Mach. Intell. 42, 12 (2019), 3071–3087.

[10] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. 2021. HiNet: Half instance normalization network for image restoration. In CVPR. IEEE, 2018–2019.

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

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2017. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Patt. Anal. Mach. Intell. 40, 4 (2017), 834–848.

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In CVPR. IEEE, 3213–3223.

[14] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. 16, 8 (2007), 2080–2095.

[15] Weisheng Dong, Xin Li, Lei Zhang, and Guangming Shi. 2011. Sparsity-based image denoising via dictionary learning and structural clustering. In CVPR. IEEE, 457–464.

[16] Michael Elad and Michal Aharon. 2006. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. 15, 12 (2006), 3736–3745.

[17] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, 249–256.

[18] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2019. Toward convolutional blind denoising of real photographs. In CVPR. IEEE, 2018–2019.

[19] Shizhong Han, Zibo Meng, Ahmed-Shehab Khan, and Yan Tong. 2016. Incremental boosting convolutional neural network for facial action unit recognition. In NIPS. 108–117.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In ICCV. IEEE, 1026–1034.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In CVPR. IEEE, 770–778.

[22] Fumitaka Hosotani, Yuya Inuzuka, Masaya Hasegawa, Shigeki Hirobayashi, and Tadanobu Misawa. 2015. Image denoising with edge-preserving and segmentation based on mask NHA. IEEE Trans. Image Process. 24 (2015), 6025–6033.

[23] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML. 448–456.

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In ECCV. Springer, 694–711.

[25] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. 2020. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In CVPR. IEEE, 3482–3492.

[26] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[27] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. 2019. Noise2Void-learning denoising from single noisy images. In CVPR. IEEE, 2129–2137.
[28] Agostina J. Larrazabal, Cesar Martinez, and Enzo Ferrante. 2019. Anatomical priors for image segmentation via post-processing with denoising autoencoders. In MICCAI. Springer, 585–593.

[29] Ghazanfar Latif, D. A. Iskandar, Jaafar Alghazo, Mohsin Butt, and Adil H. Khan. 2018. Deep CNN based MR image denoising for tumor segmentation using watershed transform. Int. J. Eng. Technol. 7 (2018), 37–42.

[30] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. 2018. Noise2Noise: Learning image restoration without clean data. In ICML. 2965–2974.

[31] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. SwinIR: Image restoration using Swin transformer. In CVPR. 1833–1844.

[32] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. 2017. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In CVPR. IEEE, 1925–1934.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In ECCV. Springer, 740–755.

[34] Xiaoyu Lin. 2021. Learning degraded image classification with restoration data fidelity. arXiv preprint arXiv:2101.09606 (2021).

[35] Ding Liu, Bihan Wen, Jianbo Jiao, Xianming Liu, Zhangyang Wang, and Thomas S. Huang. 2020. Connecting image denoising and high-level vision tasks via deep learning. IEEE Trans. Image Process. 29 (2020), 5695–5706.

[36] Ding Liu, Bihan Wen, Xianming Liu, Zhangyang Wang, and Thomas S. Huang. 2018. When image denoising meets high-level vision tasks: A deep learning approach. In IJCAI. 842–848.

[37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In ICCV. 10012–10022.

[38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In CVPR. IEEE, 3431–3440.

[39] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. 2009. Non-local sparse models for image restoration. In ICCV. IEEE, 2272–2279.

[40] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In NIPS. 2802–2810.

[41] Mohammad Moghimi, Serge J. Belongie, Mohammad J. Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. 2016. Boosted convolutional neural networks. In BMVC. 1–6.

[42] Adam Paszke, Sam Gross, Francisco Massa, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In NeurIPS. 8024–8035.

[43] Tobias Plotz and Stefan Roth. 2017. Benchmarking denoising algorithms with real photographs. In CVPR. IEEE, 1586–1595.

[44] Tal Remez, Or Litany, Raja Giryes, and Alex M. Bronstein. 2018. Class-aware fully convolutional Gaussian and Poisson denoising. IEEE Trans. Image Process. 27, 11 (2018), 5707–5722.

[45] Wenqi Ren, Jinshian Pan, Xiaochun Cao, and Ming-Hsuan Yang. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In NIPS. 2802–2810.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In MICCAI. Springer, 234–241.

[47] Jie Shao, Kai Hu, Changhu Wang, Xiangyang Xue, and Bhiksha Raj. 2020. Is normalization indispensable for training deep neural network? In NeurIPS. 13434–13444.

[48] Vivek Sharma, Ali Diba, Davy Neven, Michael S. Brown, Luc Van Gool, and Rainer Stiefelhagen. 2018. Classification-driven dynamic image enhancement. In CVPR. IEEE, 4033–4041.

[49] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[50] Maneesh Singh, Prakash Ishwar, Krishna Ratakonda, and Narendra Ahuja. 1999. Segmentation based denoising using multiple compaction domains. In ICIP. IEEE, 372–375.

[51] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. 2021. Segmenter: Transformer for semantic segmentation. In ICCV. 7626–7627.

[52] Hossein Talebi, Xiang Zhu, and Peyman Milanfar. 2012. How to SAIF-ly boost denoising performance. IEEE Trans. Image Process. 22, 4 (2012), 1470–1485.

[53] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2018. Deep image prior. In CVPR. IEEE, 9446–9454.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Adv. Neural Inf. Process. 30 (2017).

[55] Mayank Vatsa, Richa Singh, and Afzel Noore. 2018. Denoising and segmentation of 3D brain images. Image Process. Comput. Vis. Patt. Recog. 9 (2009), 561–567.

[56] Jingdong Wang, Ke Sun, Tianheng Cheng, et al. 2020. Deep high-resolution representation learning for visual recognition. IEEE Trans. Patt. Anal. Mach. Intell. (2020). DOI: 10.1109/TPAMI.2020.2983686
Synergy between Semantic Segmentation and Image Denoising via Alternate Boosting

[57] Li Wang, Dong Li, Yousong Zhu, Lu Tian, and Yi Shan. 2020. Dual super-resolution learning for semantic segmentation. In CVPR. IEEE, 3774–3783.

[58] Sicheng Wang, Bihan Wen, Junru Wu, Dacheng Tao, and Zhangyang Wang. 2019. Segmentation-aware image denoising without knowing true segmentation. arXiv preprint arXiv:1905.08965 (2019).

[59] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In ICCV. 568–578.

[60] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. 2018. Recovering realistic texture in image super-resolution by deep spatial feature transform. In CVPR. IEEE, 606–615.

[61] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuan Liu. 2021. Uformer: A general u-shaped transformer for image restoration. arXiv preprint arXiv:2106.03106 (2021).

[62] Ziyue Xu, Ulas Bagci, Jurgen Seidel, David Thomasson, Jeff Solomon, and Daniel J. Mollura. 2014. Segmentation based denoising of PET images: An iterative approach via regional means and affinity propagation. In MICCAI. Springer, 698–705.

[63] Ziyue Xu, Mingchen Gao, Georgios Z. Papadakis, Brian Luna, Sanjay Jain, Daniel J. Mollura, and Ulas Bagci. 2018. Joint solution for PET image segmentation, denoising, and partial volume correction. Med. Image Anal. 46 (2018), 229–243.

[64] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. 2018. DenseASPP for semantic segmentation in street scenes. In CVPR. IEEE, 3684–3692.

[65] Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015).

[66] Yuhui Yuan, Xilin Chen, and Jingdong Wang. 2020. Object-contextual representations for semantic segmentation. In ECCV. Springer, 173–190.

[67] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2021. Restormer: Efficient transformer for high-resolution image restoration. arXiv preprint arXiv:2111.09881 (2021).

[68] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. 2021. Multi-stage progressive image restoration. In CVPR. 14821–14831.

[69] Haochen Zhang, Dong Liu, and Zhiwei Xiong. 2019. Two-stream action-recognition-oriented video super-resolution. In ICCV. IEEE, 8799–8808.

[70] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Trans. Image Process. 26, 7 (2017), 3142–3155.

[71] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. 2017. Learning deep CNN denoiser prior for image restoration. In CVPR. IEEE, 3929–3938.

[72] Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2018. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. IEEE Trans. Image Process. 27, 9 (2018), 4608–4622.

[73] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. 2018. Joint task-recursive learning for semantic segmentation and depth estimation. In ECCV. Springer, 235–251.

[74] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaoang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In CVPR. IEEE, 2881–2890.

[75] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In CVPR. 6881–6890.

[76] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ADE20K dataset. In CVPR. IEEE, 633–641.

Received 13 December 2021; revised 2 June 2022; accepted 7 July 2022