No boundary left behind in semantic segmentation

Jefferson Fontinele da Silva
Universidade Federal do Maranhão, Campus Balsas
Balsas, Maranhão, Brazil
Email: jefferson.fs@ufma.br

Bernardo Silva, Luciano Oliveira
IvisionLab, Institute of Computing
Federal University of Bahia, Salvador, Bahia, Brazil
Emails: bernardo.peters@ufba.br, lrebouca@ufba.br

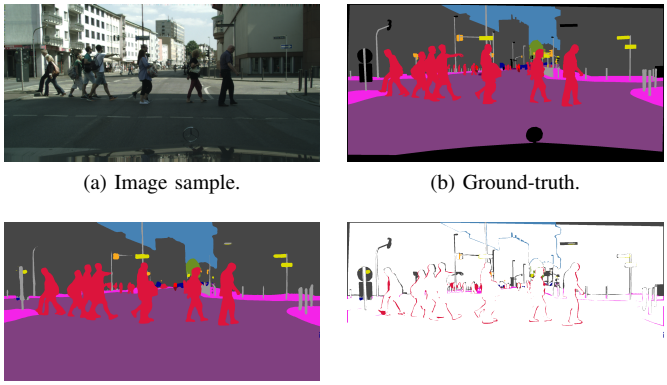
***Abstract*—This paper proposes a novel network architecture for image semantic segmentation based on attention mechanisms placed on specific points inside a convolutional neural network. Attention is explored across our network to integrate information from object boundary and a baseline semantic segmenter (inner segmentation). We call our novel network Attention-fitted Fusion of boundary and Inner Segmentation (AFIS), which combines the two streams through a set of attention gates, forming an end-to-end network. We performed an extensive evaluation of our method over four public challenging data sets (Cityscapes, CamVid, Pascal Context, and Mapillary Vistas), finding superior results when compared with other twelve state-of-the-art segmenters, considering the same training conditions.**

I. Introduction

Commonly segmentation models usually face problems in segmenting the boundaries. Figure 1 shows the result of the semantic segmentation obtained by a DeepLabV3 [1] network compared to the ground truth. The difference between ground truth (Fig. 1(b)) and predicted segmentation (Fig. 1(c)) depicted in Fig. 1(d) shows that a considerable part of the error attributed to the model is related to the boundaries. To mitigate this problem, we propose to explore a novel way to integrate boundary and inner semantic segmentation. Across a Fully Convolutional Network (FCN) segmentation architecture, a new relevant challenge is introduced: *How to integrate adequate pixel context from boundary and segmentation information?* Trying to answer this question, we investigate whether attention can contribute to better results in the final segmentation. The rationale is that it is possible to separately learn from different contexts by using attention models, also jointly training the model in an end-to-end network. We claim that pixels, which belong to the object boundaries own different features when compared to the other inner image pixels; this difference could be mainly related to the context of the outer border pixels, which ultimately contain features from the objects separated by that border.

We call our novel network architecture Attention-fitted Fusion of boundary and Inner Segmentation (AFIS), which uses two streams containing a semantic boundary detection (SBD) and an inner semantic segmenter, both combined through a set of attention gates. We carried out experiments to assess the AFIS performance on four publicly available data sets: Cityscapes [2], Mapillary Vistas [3], CamVid [4], and Pascal Context [5]. Twelve other methods and ours were benckmarked on these data sets, while showing superior results of AFIS.

978-1-6654-5385-1/22/$31.00 ©2022 IEEE



(a) Image sample. (b) Ground-truth. (c) Mask by DeepLabV3 [1]. (d) Difference between (b) and (c).

Figure 1. From the Cityscapes data set [2]: The impact of the boundaries in the image segmentation result.

II. Related work

FCN-based semantic segmentation methods: One significant issue of FCN-based approaches is related to the spatial representation of the extracted features. The downsampling stages of the classification network reduce the spatial resolution of the features, effectively attenuating the object boundaries in the input image. Hence, FCN-based segmentation networks usually produce results with less detailed boundaries when compared to the ground truth. The rationale is that features from the early stages have better spatial representation since they are not as downsampled as features from later stages. Models such as FCN-8 [6] and DeeplabV3+ [7] are examples of the networks that improve their overall results by exploiting features of the initial stages of a ResNet network [8]. However, even when combining several stages of the backbone, a significant part of the classification errors in segmentation models still occur due to failures in boundary classification, as illustrated in Fig. 1, where the difference between the segmentation result and the ground truth is mostly defined by incorrectly-classified boundaries. These results indicate that networks trained solely for semantic segmentation have difficulty in segmenting the borders of the objects. Simply using various stages of the backbone is not enough to

completely overcome the problem found on the boundaries. In AFIS, the backbone relies in a ResNet-based network along with the decoder part of the DeeplabV3+ [7] (see Fig. 2).

Boundary-based methods: Methods that detect boundaries for each semantic class [9]–[12] have obtained better results than those that perform general boundary detection. This motivated us to use semantic borders as part of our method. Inspired by the work in [11], AFIS detects semantic boundaries by combining several stages of a classification network (ResNet [8]). Those networks are considered mainly because their architecture choices lead to a just slightly increase of the number of network parameters. However, differently from them, we add spatial attention modules to allow the integration of the boundary stream with the inner semantic segmentation.

III. Fusion of boundary and inner semantic segmentation through attention gates

Figure 2 shows our complete proposed architecture, which is comprised of two main streams and a fusion module: (i) The **inner semantic segmentation stream** is represented by the *backbone* (encoder) and the *semantic multi-scale context* modules, while (ii) the **boundary stream** is represented by the *boundary detection* and *semantic boundary detection* modules; both streams are processed in parallel, and their outputs are combined to produce the final output via the **semantic fusion gate**.

A. Inner semantic segmentation stream

The inner semantic segmentation stream is responsible for carrying out a preliminary semantic segmentation step on the input image. According to the three modules found in FCN-based network, we have: The first is a feature extractor, which uses an image classification network as backbone (the one-by-one convolution, usually placed at the end of the network for class prediction, was excluded); the second module improves the context representation of each pixel by using dilated convolutions [1], [13], [14], large convolutions [15], and image pooling [16], [17]; the last and third module is the semantic multi-scale context, which is is responsible for combining the extracted features and performing pixel classification to produce the segmentation map. The backbone used by our proposed model relies in five stages as illustrated on the top-left part of Fig. 2. A **coarse semantic segmentation** (**CSS**) map contains the class prediction scores for each pixel in the image, which is ultimately refined by the output of the boundary stream.

As the base feature extractor for the inner semantic segmentation and boundary streams, we used a Dilated ResNet [18] (with 50 and 101 layers) and a WideResNet [19] (38 layers), in our experiments. For the Dilated ResNet, we used the default dilation rates of 2 and 4 in blocks 4 and 5 of our network, respectively, producing an output with stride 8. This design choice follows the suggestions in [18]. For the WideResNet, we used a dilation rate of 2 for block 3, and a dilation rate of 4 for the subsequent blocks. This final output is achieved with stride 4. We used a multi-scale representation for the context. This is done by way of an atrous spatial pyramid pooling (ASPP) [1] module, which captures multi-scale contextual information using dilated convolutions with different dilation rates. For both training and testing, we used an output stride of 8.

B. Boundary stream

The **boundary stream** is essentially in charge to detect the image object boundaries in our proposed model. We extract features from different stages of the backbone to improve the prediction accuracy on the boundaries. Features from four different stages, denoted as F_1, F_3, F_4, F_5, feed the SBD module, as depicted in Figure 2. The **boundary detection** (BD) module combines the features F_1, F_3, F_4, F_5 across the **attention gates** G (see the boundary detection module in Fig. 2) in order to output a set of features $S_n \in \{S_1, S_2, S_3\}$, representing the different processing stages on the boundary that will be taken to the SBD module. The use of these attention gates G are necessary to facilitate the flow of information from the **inner semantic segmentation stream** to the **boundary stream**. These local attention gates G consists of an attention map A_g followed by a residual basic module. The attention map, A_g, is given by

$$A_g = \sigma(\circledast_{1\times1}(F_n||B)), \tag{1}$$

where F_n represents the backbone features from the n_{th} stage used by the boundary detection module, $\mathbf{B}$ represents $\mathbf{F}_n$ features after the 1×1 convolution and batch normalization operations in the boundary detection module, $||$ is the concatenation function, $\circledast_{1\times1}$ is a convolutional layer, and σ is a sigmoid function.

The attention gate, G, is applied to F'_n (the next features after the 1×1 convolution) and it is given by

$$\mathrm{G} = \circledast_{1\times1}(F'_n \odot (A_g + k)), \tag{2}$$

where $\odot$ is an element-wise product. The resulting G is passed to a residual module, followed by a 1×1 convolution. The final output of each gate is an S_n map. k is a constant to reinforce the borders avoiding close-to-zero values on boundary elements.

The SBD is inspired by [11]. The processing of low-level feature fusion is improved by more semantic features. Unlike [11], we used the ∇F_5 as an input, in addition to the set of low level features S_n in the **boundary fusion** box. In our fusion model, first we weigh the feature maps from different levels of the network to then perform the 1×1 convolution that ultimately combines the features from multiple stages. ∇F_5 is defined as an approximation of the gradient, and is given by

$$\nabla F_5 \approx \sigma(F_5 - \mathrm{MaxPool}_{3\times3}(F_5)), \tag{3}$$

where $\mathrm{MaxPool}_{3\times3}$ represents a maximum-pooling operation in the 2D space with stride of 3 in both dimensions. The use of ∇F_5 allows to obtain the most prominent features

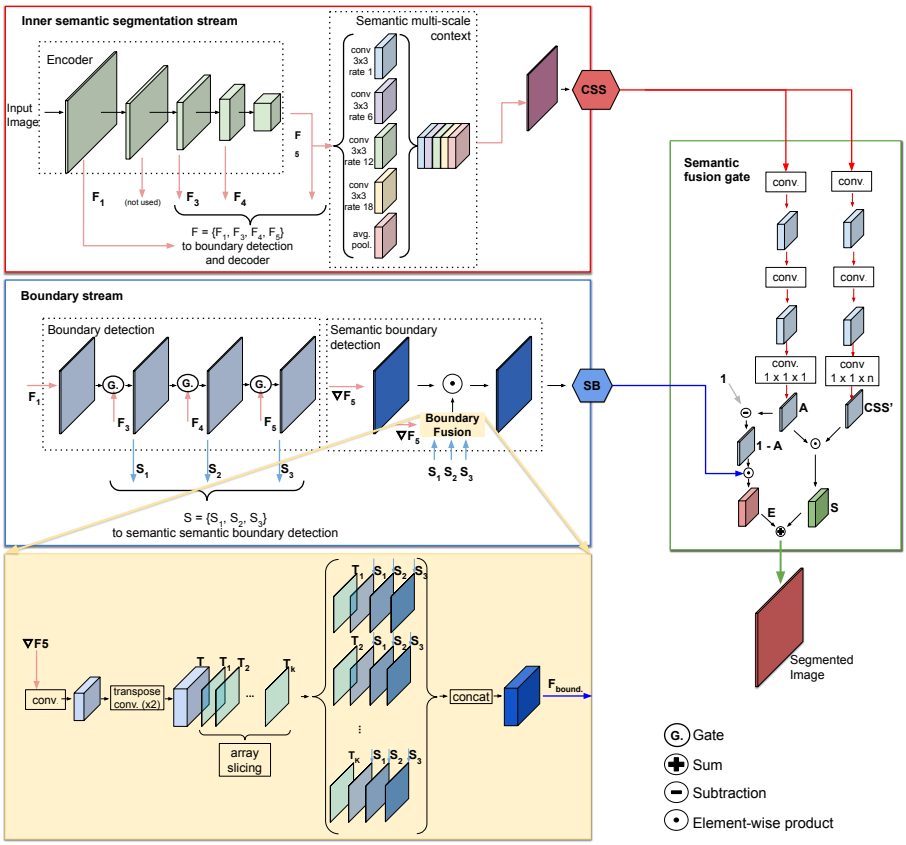


Figure 2. Detailed architecture of AFIS. The input image feeds the encoder that generates a set of feature maps, **F**, in different scales. **F** goes to the boundary stream to generate a set of features, **S**, through attention gates, **G**. In the semantic boundary detection module, the **boundary fusion** box combines the pseudo gradient, ∇F_5, obtained by a max-pool operation over F_5, and **S** to define a set of semantic boundaries, **SB**. To capture the global segmentation context, coarse semantic segmentation features (**CSS**), are achieved from the semantic multi-scale context module. The output image is found by the combination of features in **SB** and **CSS** through the semantic fusion gate, which takes as input the CSS features, using an attention gate model by generating two spatial maps of attention **A** and (**1** − **A**); these maps are used to highlight the features in **CSS'** and **SB**. Conv. 1×1×1 refers to a 1×1 convolution by 1 channel, while conv. 1×1×n is a 1×1 convolution by n classes.

from F_5, which improves the flow of information during back-propagation.

The **SBD** module is divided into two components: the **boundary fusion** box and the attention step. The former takes as inputs ∇F_5 and S_n. First, ∇F_5 is upsampled by two convolution transpose layers with stride of 8 pixels. The result is then split into k slices. Each slice, T_k, is concatenated with the S_n sets and linearly combined by a $\circledast_{1\times1}$. The boundary fusion outputs the F_{edges}, combining the sequential application of two $\circledast_{1\times1}$ with batch normalization and a ReLU activation function, L_{adapt}, through a semantic boundary attention gate, SBatt, given by

$$\text{SBatt} = L_{adapt}(\nabla F_5) \odot F_{edges}, \tag{4}$$

The rationale here is to reinforce the boundaries formed by both low-level and high-level features with more semantic information. Finally, the resulting SBatt is convoluted by a $\circledast_{1\times1}$, resulting in the semantic boundary, $\mathbf{SB} \in \mathbb{R}^{H\times W\times L}$.

C. Semantic fusion gate

The semantic fusion gate proposed in this work to separate the inner semantic segmentation and the boundary information. Given a $\text{CSS} \in \mathbb{R}^{H\times W\times L}$ and a semantic boundary $\text{SB} \in \mathbb{R}^{H\times W\times L}$, we first feed CSS into two convolutional sets. The output of the first set is denoted as $\text{CSS}' \in \mathbb{R}^{H\times W\times L}$ features, while the second set outputs a mask, $\text{A} \in \mathbb{R}^{H\times W\times L}$, that works as an attention gate. A $\mathrm{softmax}$ function is multiplied by CSS′, obtaining the segmentation, S. The input SB is multiplied by $1-\text{A}$ to get the output E. The final segmentation is finally obtained by $\text{S} + \text{E}$.

D. Multi-task loss

To train AFIS, we used different loss functions for each sub-task. We used a weighted binary cross-entropy loss (WBCE) for the boundary stream, and a binary cross-entropy (BCE) loss for the semantic segmentation (non-boundary) stream. Our multi-task loss is modeled as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{WBCE}(s, \hat{s}) + \lambda_2 \mathcal{L}_{BCE}(y, \hat{y}), \tag{5}$$

where $\hat{s} \in \mathbb{R}^{H\times W}$ denotes the ground truth of the semantic boundaries, while $\hat{y} \in \mathbb{R}^{H\times W}$ denotes the ground truth of the semantic segmentation. s and y represent the predicted values of the boundary and segmentation, respectively. λ_1 and λ_2 are

two hyper-parameters that control the weighting between the losses.

IV. Materials and methods

We carried out experiments in four publicly available data sets: Cityscapes [2], Mapillary Vistas [3], CamVid [4], and Pascal Context [5]. These data sets were chosen by considering the need for annotations with fine granularity and well-defined boundaries, so that the training of the boundary-detection portion of AFIS was carried out adequately. None of these data sets has annotations for semantic boundaries, demanding us to derive the ground truth of the boundaries from the semantic segmentation annotations. We used a distance transform to select pixels, which belong to the boundaries of each class of the semantic segmentation annotations. As data augmentation strategies, we applied random flips, random scaling in a range of 0.5-2, and crops with a fixed size. We also used photometric distortions such as Gaussian blur and variations to the brightness, hue, and saturation.

We used a Dilated ResNet [18] backbone, which was pre-trained on ImageNet 1k [20], with dilation in the last two stages and output size of 1/8. We followed the work in [1], [16] to establish the learning rate schedule. All the training stage was done on a V100 NVIDIA DGX Station with 8 GPUs. The batch size was 8 for Cityscapes, Mapillary Vistas and CamVid, and 16 for Pascal Context. We used fixed crop with values equals to 800×800 for Cityscapes, Mapillary Vistas, and CamVid; for Pascal Context, we used 512×512.

V. Experimental evaluation

In an ablative study on the Cityscapes validation set, we first analyzed the influence of using the boundary stream and the semantic fusion gate model compared with a baseline model. The effectiveness of the boundary detection was evaluated by varying the border thickness of the image boundaries, as well. Finally, we compared our best setup with twelve other state-of-the-art methods in the literature.

A. Ablation study

We compared the performance of four baselines against AFIS: DeeplabV3, DeeplabV3+ (as individual segmentation streams), and a simple fusion approach considering an add operation of the resulting images separately obtained by the boundary and inner segmentation (considering DeeplabV3 and DeeplabV3+ as the semantic segmenters) streams. The backbone considered was ResNet101 for all architectures. The results are summarized in Table I. The results showed that the use of both streams within AFIS improved all the considered baselines over the Cityscapes validation set, having the best results when using DeeplabV3+. This leads us to conclude that the performance of semantic segmentation suffers when using the boundary stream in simple fusion mechanism (+Operation) without any other mechanism to avoid noise values originated from one of the two streams. Table II derives from Table I and show the results found per object class. When AFIS is based on DeepLabV3+, it outperforms the two baseline models for all classes but wall, terrain, rider, and motor.

Table I
Results on Cityscapes **validation set** comparing individual segmenters and a simple add fusion approach (+Operation) between the resulting images obtained by the boundary and inner semantic segmentation (DeeplabV3, DeeplabV3+) streams. The backbone is ResNet101 for all architectures.

Method	mIoU
DeeplabV3	75.0
+Operation (DeeplabV3)	75.3
AFIS (DeeplabV3)	**78.2**
DeeplabV3+	77.0
+Operation (DeeplabV3+)	76.6
AFIS (DeeplabV3+)	**78.9**

B. Evaluating the semantic boundary detection

By evaluating the semantic boundary detection, we aim to demonstrate that AFIS performs better than the baselines. The results of this experiment are summarized in Table III. In this case, the f1-boundary score represents the contour matching between the predicted segmentation and the ground truth. By using that score, we could determine the thickness of the boundary (**B. Thick**) that was considered in the other experiments henceforth. AFIS showed superior results in terms of boundary detection even in comparison with strong baselines such as DeepLabV3 and DeepLabV3+.

C. Comparison with the state-of-the-art

We compared our method with twelve other methods [7], [13], [15], [16], [18], [21]–[25], considered closely related to ours. For the Cityscapes, Mapillary Vistas, and CamVid data sets, we used the WideResNet network with 38 layers [19] and with dilation in the last 3 residual blocks. We used the extended ResNet network with 101 layers [18] as a backbone for our network on the Pascal Context data sets. The results found over the selected data sets are presented in Tables IV, V, VI, and VII for the Cityscapes, Mapillary Vistas, CamVid, and Pascal Context data sets, respectively.

Results on Cityscapes: We obtained the scores with 90k training steps. We did not use coarse annotation because the loss used in AFIS needs fine boundary annotation. Both the training set and the validation set were used for fine-tuning. For prediction, we adopted a multi-scale strategy with values of 0.5, 1, and 2 for all the methods. The best result was achieved by AFIS with a mIoU of 80.6%.

Results on Mapillary Vistas: To train Mapillary Vistas, we used 110k training steps. We report our results in Mapillary Vistas validation set using a single scale for prediction. AFIS achieved the best result with a mIoU of 52.3%.

Results on CamVid: We first pre-trained AFIS in the Cityscapes training set. The pre-training with Cityscapes was obtained with 90k training steps. This procedure was necessary due to the CamVid data set provides only 369 images for training. After pre-training with Cityscapes, only the first layer of each stream was used in the training with CamVid data set. We used 10k training steps to fine-tuning the training

Table II
PER-CLASS RESULTS OF AFIS ON THE CITYSCAPES VALIDATION SET MEASURED BY F1-BOUNDARY SCORE.

Method	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bicycle	mean
DeepLabV3	**98.1**	84.1	**91.7**	**55.7**	60.3	53.1	67.3	75.2	91.2	54.0	93.3	78.2	57.9	94.2	77.0	83.5	**72.0**	62.7	74.9	75.0
+Operation (DeeplabV3)	98.0	84.0	91.1	44.5	53.9	53.3	**74.5**	77.3	**92.5**	51.4	**94.4**	76.3	62.2	93.7	**79.1**	85.4	51.4	62.1	75.7	75.3
AFIS (DeepLabV3)	98.0	**85.0**	91.0	53.0	**61.0**	**64.0**	72.0	**79.0**	92.0	**65.0**	93.0	**83.0**	**64.0**	**95.0**	73.0	**87.0**	70.0	**72.0**	**78.0**	78.2
DeepLabV3+	98.2	85.2	92.7	49.0	61.3	66.9	71.4	79.7	92.3	55.8	94.8	82.0	60.8	95.1	76.9	84.8	77.8	61.5	77.1	77.0
+Operation (DeeplabV3+)	97.2	83.2	92.4	44.6	54.8	67.1	73.6	78.3	92.5	53.5	94.4	81.0	63.9	94.3	78.6	86.3	78.6	64.2	77.4	76.6
AFIS (DeepLabV3+)	**98.4**	**86.4**	**93.1**	**51.5**	**64.9**	**68.8**	**73.6**	**81.4**	**92.7**	**58.9**	**95.0**	**83.4**	63.9	**95.5**	**80.9**	**87.8**	**80.9**	64.2	**78.1**	**78.9**

Table III
COMPARISON BETWEEN THE BASELINE (DEEPLABV3) AND AFIS ON THE CITYSCAPES VALIDATION SET, CONSIDERING DIFFERENT BOUNDARY THICKNESS (**B. THICK**) MEASURED BY F1-BOUNDARY SCORE.

B. Thick.	Method	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bicycle	mean
3px	DeepLabV3	80.8	57.6	60.9	48.5	**48.1**	53.8	53.2	57.0	61.8	50.2	73.0	50.8	61.9	70.4	**74.1**	84.0	90.0	**74.8**	55.0	63.4
	AFIS (DeepLabV3)	**83.7**	**64.8**	**70.2**	**53.7**	46.0	**73.3**	**62.0**	**69.6**	**71.6**	**53.6**	**81.7**	**63.9**	**62.5**	**79.6**	73.8	**88.1**	**93.6**	72.2	**60.5**	**69.7**
	DeepLabV3+	83.7	64.8	**70.2**	53.7	46.0	73.3	62.0	69.6	71.6	53.6	**81.7**	63.9	62.5	79.6	73.8	**88.1**	93.6	72.2	60.5	69.7
	AFIS (DeepLabV3+)	**83.8**	**64.9**	69.5	**56.2**	**50.4**	**74.0**	**73.4**	**74.1**	**71.7**	**53.8**	81.2	**67.8**	**69.7**	**80.8**	**81.4**	87.7	**93.6**	**78.9**	**64.1**	**72.5**
5px	DeepLabV3	**86.2**	69.0	**73.5**	51.6	51.6	69.0	61.7	**70.5**	75.6	54.8	**82.6**	63.2	68.5	**82.2**	75.4	85.5	90.3	76.7	**64.6**	71.2
	AFIS (DeepLabV3)	86.0	**69.0**	73.2	**52.3**	**53.1**	**69.0**	**66.6**	69.4	**75.7**	**55.5**	81.5	**63.5**	**69.3**	82.1	**78.4**	**88.1**	**92.1**	**79.3**	63.9	**72.0**
	DeepLabV3+	**88.0**	71.9	**77.9**	56.2	49.0	78.2	67.8	75.7	80.4	57.4	**86.8**	70.5	67.5	85.5	74.8	**89.3**	93.8	73.8	67.5	74.3
	AFIS (DeepLabV3+)	87.9	**72.2**	76.9	**58.8**	**53.4**	**78.5**	**79.0**	**79.8**	**80.6**	**57.7**	86.6	**74.4**	**74.8**	**86.7**	**82.6**	88.8	**93.9**	**80.4**	**71.0**	**77.1**
9px	DeepLabV3	90.3	76.5	82.9	54.8	55.2	77.6	68.1	**78.4**	85.7	58.9	**88.0**	72.2	74.8	**89.3**	76.8	86.9	90.7	78.6	**73.4**	76.8
	AFIS (DeepLabV3)	**90.4**	**76.8**	**82.9**	**55.6**	**57.0**	**77.9**	**72.9**	78.0	**86.3**	**59.6**	87.6	**72.8**	**74.9**	89.3	**79.6**	**89.5**	**92.6**	**81.1**	73.1	**77.8**
	DeepLabV3+	90.9	77.2	**84.1**	58.9	52.1	81.5	71.9	79.5	86.9	60.8	89.6	75.5	72.0	89.6	75.9	**90.3**	**94.2**	75.1	73.7	77.9
	AFIS (DeepLabV3+)	**90.9**	**77.6**	82.8	**61.6**	**56.6**	**81.7**	**82.9**	**83.5**	**87.3**	**61.5**	**89.6**	**79.2**	**79.2**	**90.8**	**83.9**	89.7	94.1	**82.0**	**77.3**	**80.6**
12px	DeepLabV3	91.5	78.7	85.7	56.2	56.6	79.4	69.8	**80.2**	88.5	60.3	**89.3**	74.5	**76.9**	**91.1**	77.4	87.4	90.9	79.3	**76.5**	78.4
	AFIS (DeepLabV3)	**91.6**	**79.1**	**85.8**	**56.9**	**58.5**	**79.8**	**74.6**	80.0	**89.2**	**61.2**	89.0	**75.2**	76.9	91.0	**80.2**	**90.0**	**92.8**	**81.8**	76.5	**79.5**
	DeepLabV3+	91.9	79.1	**86.4**	60.2	53.4	83.0	73.1	80.8	89.1	62.2	**90.6**	77.2	73.6	91.1	76.4	**90.7**	**94.3**	75.8	76.0	79.2
	AFIS (DeepLabV3+)	**92.0**	**79.5**	85.1	**62.8**	**58.1**	**83.1**	**84.0**	**84.8**	**89.6**	**63.1**	90.5	**80.9**	**81.0**	**92.2**	**84.3**	90.0	94.2	**82.6**	**79.7**	**82.0**

Table IV
COMPARATIVE RESULTS IN THE CITYSCAPES TESTING SET.

Method	Multi-scale	**mIoU (%)**
DeepLabV3+ (dilated-ResNet-50) [7]		73.0
Dilated-ResNet-101 [18]		75.7
DeepLabV3+ (dilated-ResNet-101) [7]		76.5
Large Kernel Matters [15]		76.9
PSANet (dilated-ResNet-101) [21]	✓	80.1
PSPNet (dilated-ResNet-101) [16]	✓	80.2
AFIS (WideResNet-38)	✓	**80.6**

Table V
COMPARATIVE RESULTS IN THE MAPILLARY VISTAS VALIDATION SET.

Method	**mIoU (%)**
FCN (WideResNet-38) [22]	41.1
FCN (WideResNet-38 + bcs) [22]	47.7
PSPNet (dilated-ResNet101) [16]	49.7
Seamless [23]	50.4
AFIS (WideResNet-38)	**52.3**

in CamVid. AFIS achieved the best result, 1 percentage point better than the second-best method, using a single-scale inference.

Results on Pascal Context: To evaluate our network in Pascal Context, we used a subset of 59 classes in a 90k training steps. AFIS reached the best result in this data set, 0.3 percentage point better than the second place, using a single-scale inference.

VI. CONCLUDING REMARKS

Each stream that composes AFIS is responsible for specializing in different types of information for image segmentation.

Table VI
COMPARATIVE RESULTS IN CAMVID TESTING SET.

Method	mIoU (%)
PSPNet (dilated-ResNet-101) [16]	69.1
BiSetNet [24]	68.7
Dilated ResNet-101 [18]	65.3
BFP [25]	74.1
AFIS (WideResNet-38)	**75.1**

Table VII
COMPARATIVE RESULTS IN PASCAL CONTEXT VALIDATION SET.

Method	mIoU (%)
Dilated-ResNet-101 [18]	42.6
RefineNet [13]	47.3
PSPNet (dilated-ResNet101) [16]	47.8
AFIS (dilated-ResNet101)	**48.1**

At the same time, these streams share enough information as to complement each other. This sharing is made possible through the semantic fusion gate, which we demonstrated to be an effective manner to fuse information from the two different sub-networks (streams). Our semantic fusion gate facilitated the learning of the two tasks simultaneously but separately, with each stream focusing on the appropriate information. Even though AFIS was trained on finely-annotated data sets, some of them provide coarse annotations to increment the amount of data for training. Developing a boundary stream model that could be pre-trained with coarse annotations could improve segmentation results, and a strategy for that could be worth it in a future work. Also, it is important to note that the straight representation of semantic boundaries increases the computational complexity of the model in memory requirements. So elaborating on more efficient forms of memory representation and computational cost for semantic boundary can also be an important path for future research.

ACKNOWLEDGEMENTS

The authors acknowledge the National Laboratory for Scientific Computing (LNCC/MCTI, Brazil) for providing HPC resources of the SDumont supercomputer, which have contributed to the research results reported here. Bernardo Silva receives a scholarship from Fundação de Apoio à Pesquisa do Estado da Bahia (FAPESB) under grant BOL0569/2020. Luciano Oliveira receives a research productivity award from the Brazilian Research Council (CNPq), grant no. 308580/2021-4.

REFERENCES

[1] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[3] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 4990–4999.

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European Conference on Computer Vision (ECCV)*, 2008, pp. 44–57.

[5] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 891–898.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[7] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[9] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "Casenet: Deep category-aware semantic edge detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5964–5973.

[10] D. Acuna, A. Kar, and S. Fidler, "Devil is in the edges: Learning semantic boundaries from noisy annotations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 075–11 083.

[11] Y. Hu, Y. Chen, X. Li, and J. Feng, "Dynamic feature fusion for semantic edge detection," *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019.

[12] W. Ma, C. Gong, S. Xu, and X. Zhang, "Multi-scale spatial context-based semantic edge detection," *Information Fusion*, vol. 64, pp. 238–251, 2020.

[13] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[14] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3684–3692.

[15] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4353–4361.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[17] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[18] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 472–480.

[19] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference (BMVC)*, E. R. H. Richard C. Wilson and W. A. P. Smith, Eds. BMVA Press, September 2016, pp. 87.1–87.12.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[21] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 267–283.

[22] Z. Wu, C. Shen, and A. v. d. Hengel, "High-performance semantic segmentation using very deep fully convolutional networks," *arXiv preprint arXiv:1604.04339*, 2016.

[23] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8277–8286.

[24] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European conference on computer vision (ECCV)*, 2018, pp. 325–341.

[25] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *IEEE International Conference on Computer Vision*, 2019, pp. 6819–6829.