Gated-SCNN: Gated Shape CNNs for Semantic Segmentation

Towaki Takikawa[1,2*] David Acuna[1,3,4*] Varun Jampani[1†] Sanja Fidler[1,3,4]
[1]NVIDIA [2]University of Waterloo [3] University of Toronto [4]Vector Institute
ttakikaw@edu.uwaterloo.ca, davidj@cs.toronto.edu, {vjampani, sfidler}@nvidia.com

Abstract

Current state-of-the-art methods for image segmentation form a dense image representation where the color, shape and texture information are all processed together inside a deep CNN. This however may not be ideal as they contain very different type of information relevant for recognition. Here, we propose a new two-stream CNN architecture for semantic segmentation that explicitly wires shape information as a separate processing branch, i.e. shape stream*, that processes information in parallel to the classical stream. Key to this architecture is a new type of gates that connect the intermediate layers of the two streams. Specifically, we use the higher-level activations in the classical stream to gate the lower-level activations in the shape stream, effectively removing noise and helping the shape stream to only focus on processing the relevant boundary-related information. This enables us to use a very shallow architecture for the shape stream that operates on the image-level resolution. Our experiments show that this leads to a highly effective architecture that produces sharper predictions around object boundaries and significantly boosts performance on thinner and smaller objects. Our method achieves state-of-the-art performance on the Cityscapes benchmark, in terms of both mask (mIoU) and boundary (F-score) quality, improving by 2% and 4% over strong baselines.*

1. Introduction

Semantic image segmentation is one of the most widely studied problems in computer vision with applications in autonomous driving [43, 17, 58], 3D reconstruction [38, 30] and image generation [22, 48] to name a few. In recent years, Convolutional Neural Networks (CNNs) have led to dramatic improvements in accuracy in almost all the major segmentation benchmarks. A standard practice is to adapt an image classification CNN architecture for the task of semantic segmentation by converting fully-connected layers into convolutional layers [37]. However, using classification architectures for dense pixel prediction has several drawbacks [52, 37, 59, 11]. One eminent drawback is the loss in spatial resolution of the output due to the use of pooling layers. This prompted several works [52, 59, 15, 35, 21] to propose specialized CNN modules that help restore the spatial resolution of the network output.

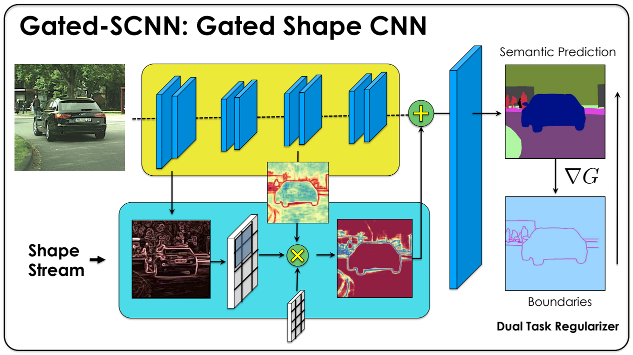

Figure 1: We introduce *Gated-SCNN* (GSCNN), a new two-stream CNN architecture for semantic segmentation that explicitly wires shape information as a separate processing stream. GSCNN uses a new gating mechanism to connect the intermediate layers. Fusion of information between streams is done at the very end through a fusion module. To predict high-quality boundaries, we exploit a new loss function that encourages the predicted semantic segmentation masks to align with ground-truth boundaries.

We argue here that there is also an inherent inefficacy in the architecture design since color, shape and texture information are all processed together inside one deep CNN. Note that these likely contain very different amounts of information that are relevant for recognition. For example, one may need to look at the complete and detailed object boundary to get a discriminative encoding of shape [2, 33], while color and texture contain fairly low-level information. This may also provide an insight of why residual [19], skip [19, 53] or even dense connections [21] lead to the most prominent performance gains. Incorporating additional connectivity helps the different types of information to flow across different scales of network depth. Disentangling these representations by design may, however, lead to a more natural and effective recognition pipeline.

In this work, we propose a new two-stream CNN architecture for semantic segmentation that explicitly wires shape information as a separate processing branch. In par-

[*] authors contributed equally
[†] work done at NVIDIA, now at Google Research

ticular, we keep the classical CNN in one stream, and add a so-called *shape stream* that processes information in parallel. We explicitly do not allow fusion of information between the two streams until the very top layers.

Key to our architecture are a new type of gates that allow the two branches to interact. In particular, we exploit the higher-level information contained in the classical stream to denoise activations in the shape stream in its very early stages of processing. By doing so, the shape stream focuses on processing only the relevant information. This allows the shape stream to adopt a very effective shallow architecture that operates on the full image resolution. To achieve that the shape information gets directed to the desired stream, we supervise it with a semantic boundary loss. We further exploit a new loss function that encourages the predicted semantic segmentation to correctly align with the ground-truth semantic boundaries, which further encourages the fusion layer to exploit information coming from the shape stream. We call our new architecture *GSCNN*.

We perform extensive evaluation on the Cityscapes benchmark [13]. Note that our GSCNN can be used as plug-and-play on top of any classical CNN backbone. In our experiments, we explore ResNet-50 [19], ResNet-101 [19] and WideResnet [57] and show significant improvements in all. We outperform the state-of-the-art DeepLab-v3+[11] by more than 1.5 % in terms of mIoU and 4% in F-boundary score. Our gains are particularly significant for the thinner and smaller objects (*i.e*. poles, traffic light, traffic signs), where we get up to 7% improvement in terms of IoU.

We further evaluate performance at varying distances from the camera, using a prior as proxy for distance. Experiments show that we consistently outperform the state-of-the-art baseline achieving up to 6% improvement in terms of mIoU at the largest distance (*i.e*. further away objects).

2. Related Work

Semantic Segmentation. State-of-the-art approaches for semantic segmentation are predominantly based on CNNs. Earlier approaches [37, 9] convert classification networks into fully convolutional networks (FCNs) for efficient end-to-end training for semantic segmentation. Several works [8, 32, 60, 44, 20, 3, 36, 23, 5] propose to use structured prediction modules such as conditional random fields (CRFs) on network output for improving the segmentation performance, especially around object boundaries. To avoid costly DenseCRF [29], the work of [6] uses fast domain transform [16] filtering on network output while also predicting edge maps from intermediate CNN layers. We also predict boundary maps to improve segmentation performance. Contrary to [6], which uses edge information to refine network output, we inject the learned boundary information into intermediate CNN layers. Moreover, we propose specialized network architecture and a dual-task regularizer to obtain high-quality boundaries.

More recently, dramatic improvements in performance and inference speed have been driven by new architectural designs. For example, PSPNet [59] and DeepLab [8, 11] proposed a feature pyramid pooling module that incorporates multiscale context by aggregating features at multiples scales. Similar to us, [43] proposed a two stream network, however, in their case, the main purpose of the second stream is to recover high-resolution features that are lost with pooling layers. Here, we explicitly specialize the second stream to process shape related information. Some works [15, 35, 49] propose modules that use learned pixel affinities for structured information propagation across intermediate CNN representations. Instead of learning specialized information propagation modules, we propose to learn high-quality shape information through carefully designed network and loss functions. Since we simply concatenate shape information with segmentation CNN features, our approach can be easily incorporated into existing networks for performance improvements.

Multitask Learning. Several works have also explored the idea of combining networks for complementary tasks to improve learning efficiency, prediction accuracy and generalization across computer vision tasks. For example, the works of [46, 39, 27, 26, 28], proposed unified architectures that learn a shared representation using multi-task losses. Our main goal is not to train a multi-task network, but to enforce a structured representation that exploits the duality between the segmentation and boundary prediction tasks. [12, 4] simultaneously learned segmentation and boundary detection network, while [31, 41] learned boundaries as an intermediate representation to aid segmentation. Contrary to these works, where semantics and boundary information interact only at the loss functions, we explicitly inject boundary information into segmentation CNN and also propose a dual-task loss function that refines both semantic masks and boundary predictions.

Gated Convolutions. Recent work on language modeling have also proposed the idea of using gating mechanisms in convolutions. For instance, [14] proposed to replace the recurrent connections typically used in recurrent networks with gated temporal convolutions. [54], on the other hand, proposed the use of convolutions with a soft-gating mechanism for Free-Form Image Inpainting and [47] proposed Gated PixelCNN for conditional image generation. In our case, we use a gated convolution operator for the task of semantic segmentation and to define the information flow between the shape and regular streams.

3. Gated Shape CNN

In this section, we present our Gated-Shape CNN architecture for semantic segmentation. As depicted in Fig. 2,

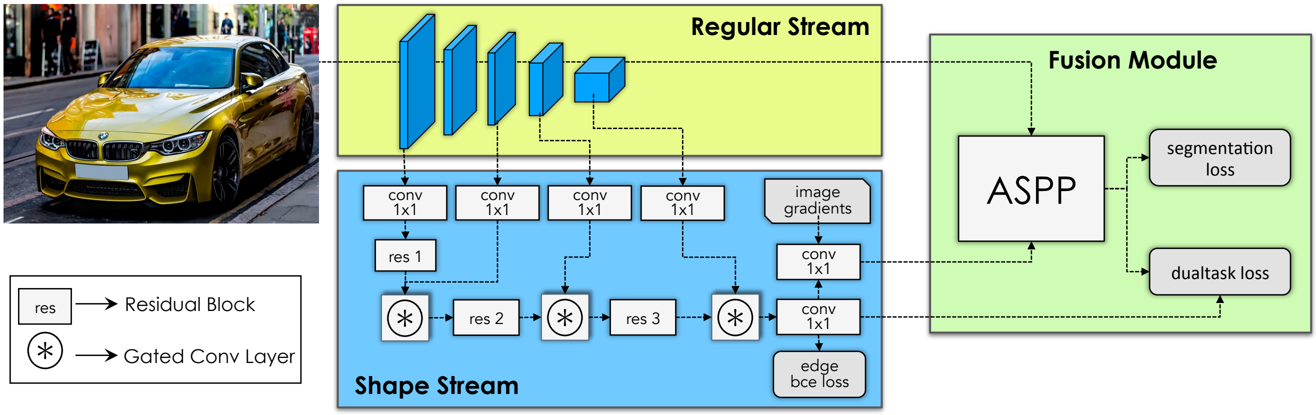

Figure 2: **GSCNN architecture.** Our architecture constitutes of two main streams. The regular stream and the shape stream. The regular stream can be any backbone architecture. The shape stream focuses on shape processing through a set of residual blocks, Gated Convolutional Layers (GCL) and supervision. A fusion module later combines information from the two streams in a multi-scale fashion using an Atrous Spatial Pyramid Pooling module (ASPP). High quality boundaries on the segmentation masks are ensured through a Dual Task Regularizer .

our network consists of two streams of networks followed by a fusion module. The first stream of the network ("regular stream") is a standard segmentation CNN, and the second stream ("shape stream") processes shape information in the form of semantic boundaries. We enforce shape stream to only process boundary-related information by our carefully designed Gated Convolution Layer (GCL) and local supervision. We then fuse semantic-region features from the regular stream and boundary features from the shape stream to produce a refined segmentation result, especially around boundaries. Next, we describe, in detail, each of the modules in our framework followed by our novel GCL.

Regular Stream. This stream, denoted as $\mathcal{R}_\theta(I)$, with parameters θ, takes image $I \in \mathbb{R}^{3\times H\times W}$ with height H and width W as input and produces dense pixel features. The regular stream can be any feedforward fully-convoutional network such as ResNet [19] based or VGG [45] based semantic segmentation network. Since ResNets are the recent state-of-the-art for semantic segmentation, we make use of ResNet-like architecture such as ResNet-101 [19] and WideResNet [57] for the regular stream. We denote the output feature representation of the regular stream as $r \in \mathbb{R}^{C\times\frac{H}{m}\times\frac{W}{m}}$ where m is the stride of the regular stream.

Shape Stream. This stream, denoted as $\mathcal{S}_\phi$, with parameters ϕ, takes image gradients ∇I as well as output of the first convolutional layer of the regular stream as input and produces semantic boundaries as output. The network architecture is composed of a few residual blocks interleaved with gated convolution layers (GCL). GCL, explained below, ensures that the shape stream only processes boundary-relevant information. We denote the output boundary map of the shape stream as $s \in \mathbb{R}^{H\times W}$. Since we can obtain ground-truth (GT) binary edges from GT semantic segmentation masks, we use supervised binary cross entropy loss on output boundaries to supervise the shape stream.

Fusion Module. This module, denoted as $\mathcal{F}_\gamma$, with parameters γ, takes as input the dense feature representation r coming from the regular branch and fuses it with the boundary map s output by the shape branch in a way that multi-scale contextual information is preserved. It combines region features with boundary features and outputs a refined semantic segmentation output. More formally, for a segmentation prediction of K semantic classes, it outputs a categorical distribution $f = p(y|s,r) = \mathcal{F}_\gamma(s,r) \in \mathbb{R}^{K\times H\times W}$, which represents the probability that pixels belong to each of the K classes. Specifically, we merge the boundary map s with r using an Atrous Spatial Pyramid Pooling [11]. This allows us to preserve the multi-scale contextual information and is proven to be an essential component in state-of-the-art semantic segmentation networks.

3.1. Gated Convolutional Layer

Since the tasks of estimating semantic segmentation and semantic boundaries are closely related, we devise a novel GCL layer that facilitates flow of information from the regular stream to the shape stream. GCL is a core component of our architecture and helps the shape stream to only process relevant information by filtering out the rest. Note that the shape stream does not incorporate features from the regular stream. Rather, it uses GCL to deactivate its own activations that are not deemed relevant by the higher-level information contained in the regular stream. One can think of this as a collaboration between two streams, where the more powerful one, which has formed a higher-level semantic understanding of the scene, helps the other stream to focus only on the relevant parts since start. This enables the shape stream to adopt an effective shallow architecture that processes the image at a very high resolution.

We use GCL in multiple locations between the two streams. Let m denote the number of locations, and let $t \in 0, 1, \cdots, m$ be a running index where r_t and s_t denote intermediate representations of the corresponding regular and shape streams that we process using a GCL. To apply GCL, we first obtain an attention map $\alpha_t \in \mathbb{R}^{H \times W}$ by concatenating r_t and s_t followed by a normalized 1×1 convolutional layer $C_{1\times 1}$ which in turn is followed by a sigmoid function σ :

$$\alpha_t = \sigma(C_{1\times 1}(s_t || r_t)), \tag{1}$$

where $||$ denotes concatenation of feature maps. Given the attention map α_t, GCL is applied on s_t as an element-wise product $\odot$ with attention map α followed by a residual connection and channel-wise weighting with kernel w_t. At each pixel (i, j), GCL $\circledast$ is computed as

$$\begin{aligned} \hat{s}_t^{(i,j)} &= (s_t \circledast w_t)_{(i,j)} \\ &= ((s_{t_{(i,j)}} \odot \alpha_{t_{(i,j)}}) + s_{t_{(i,j)}})^T w_t. \end{aligned} \tag{2}$$

$\hat{s}_t$ is then passed on to the next layer in the shape stream for further processing. Note that both the attention map computation and gated convolution are differentiable and therefore backpropagation can be performed end-to-end. Intuitively, α can also be seen as an attention map that weights more heavily areas with important boundary information. In our experiments, we use three GCLs and connect them to the third, fourth and last layer of the regular stream. Bilinear interpolation, if needed, is used to upsample the feature maps coming from the regular stream.

3.2. Joint Multi-Task Learning

We jointly learn the regular and shape streams together with the fusion module in an end-to-end fashion. We jointly supervise segmentation and boundary map prediction during training. Here, the boundary map is a binary representation of all the outlines of objects and stuff classes in the scene (Fig 6). We use standard binary cross-entropy (BCE) loss on predicted boundary maps s and use standard cross-entropy (CE) loss on predicted semantic segmentation f:

$$\mathcal{L}^{\theta\,\phi,\gamma} = \lambda_1 \mathcal{L}^{\theta,\phi}_{BCE}(s, \hat{s}) + \lambda_2 \mathcal{L}^{\theta\,\phi,\gamma}_{CE}(\hat{y}, f) \tag{3}$$

where $\hat{s} \in \mathbb{R}^{H \times W}$ denotes GT boundaries and $\hat{y} \in \mathbb{R}^{H \times W}$ denotes GT semantic labels. Here, λ_1, λ_2 are two hyper-parameters that control the weighting between the losses.

As depicted in Fig. 2, the BCE supervision on boundary maps s is performed before feeding them into the fusion module. Thus the BCE loss $\mathcal{L}^{\theta,\phi}_{BCE}$ updates the parameters of both the regular and shape streams. The final categorical distribution f of semantic classes is supervised with CE loss $\mathcal{L}^{\theta\,\phi,\gamma}_{CE}$ at the end as in standard semantic segmentation networks, updating all the network parameters. In the case of BCE on boundaries, we follow [51, 55] and use a coefficient β to account for the high imbalance between boundary/non-boundary pixels.

3.3. Dual Task Regularizer

As mentioned above, $p(y|r, s) \in R^{K \times H \times W}$ denotes a categorical distribution output of the fusion module. Let $\zeta \in R^{H \times W}$ be a potential that represents whether a particular pixel belongs to a semantic boundary in the input image I. It is computed by taking a spatial derivative on segmentation output as follows:

$$\zeta = \frac{1}{\sqrt{2}} ||\nabla(G * \arg\max_k p(y^k|r, s))|| \tag{4}$$

where G denotes Gaussian filter. If we assume $\hat{\zeta}$ is a GT binary mask computed in the same way from the GT semantic labels $\hat{f}$, we can write the following loss function:

$$\mathcal{L}^{\theta\,\phi,\gamma}_{reg_{\rightarrow}} = \lambda_3 \sum_{p^+} |\zeta(p^+) - \hat{\zeta}(p^+)| \tag{5}$$

where p^+ contains the set of all non-zero pixel coordinates in both ζ and $\hat{\zeta}$. Intuitively, we want to ensure that boundary pixels are penalized when there is a mismatch with GT boundaries, and to avoid non-boundary pixels to dominate the loss function. Note that the above regularization loss function exploits the duality between boundary prediction and semantic segmentation in the boundary space.

Similarly, we can use the boundary prediction from the shape stream $s \in \mathbb{R}^{H \times W}$ to ensure consistency between the binary boundary prediction s and the predicted semantics $p(y|r, s)$:

$$\mathcal{L}^{\theta\,\phi,\gamma}_{reg_{\leftarrow}} = \lambda_4 \sum_{k,p} \mathbb{1}_{s_p} [-\hat{y}^k_p \log p(y^k_p | r, s)], \tag{6}$$

where p and k runs over all image pixels and semantic classes, respectively. $\mathbb{1}_s = \{1 : s > \textit{thrs}\}$ corresponds to the indicator function and *thrs* is a confidence threshold, we use 0.8 in our experiments. The total dual task regularizer loss function can be written as:

$$\mathcal{L}^{\theta\,\phi,\gamma} = \mathcal{L}^{\theta\,\phi,\gamma}_{reg_{\rightarrow}} + \mathcal{L}^{\theta\,\phi,\gamma}_{reg_{\leftarrow}} \tag{7}$$

Here, λ_3 and λ_4 are two hyper-parameters that control the weighting of the regularizer.

3.3.1 Gradient Propagation during Training

In order to back-propagate through Eq 7, we need to compute the gradients of Eq 4. Letting $g = ||.||$, the partial derivatives with respect to a given parameter η can be computed as follows:

$$\frac{\partial L}{\partial \eta_i} = \sum_{j,l} \nabla G * \frac{\partial L}{\partial \zeta_j} \frac{\partial \zeta_j}{\partial g_l} \frac{\partial \arg\max_k p(y^k)_l}{\partial \eta_i} \tag{8}$$

Since $\arg\max$ is not a differentiable function we use the Gumbel softmax trick [24]. During the backward pass, we

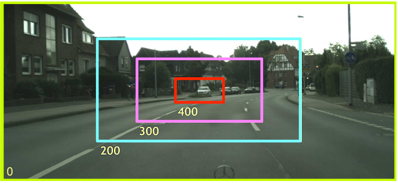

Figure 3: Illustration of the crops used for the distance-based evaluation.

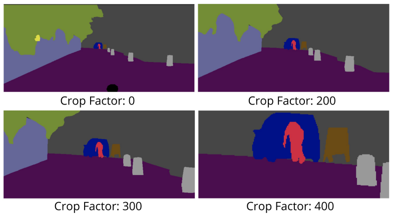
Crop Factor: 0
Crop Factor: 200
Crop Factor: 300
Crop Factor: 400

Figure 4: Predictions at diff. crop factors.

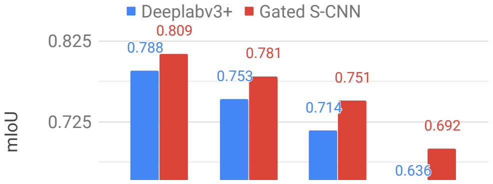

Figure 5: **Distance-based evaluation**: Comparison of mIoU at different crop factors.

Method	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
LRR [18]	97.7	79.9	90.7	44.4	48.6	58.6	68.2	72.0	92.5	69.3	94.7	81.6	60.0	94.0	43.6	56.8	47.2	54.8	69.7	69.7
DeepLabV2 [8]	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
Piecewise [32]	98.0	82.6	90.6	44.0	50.7	51.1	65.0	71.7	92.0	72.0	94.1	81.5	61.1	94.3	61.1	65.1	53.8	61.6	70.6	71.6
PSP-Net [59]	98.2	85.8	92.8	57.5	65.9	62.6	71.8	80.7	92.4	64.5	94.8	82.1	61.5	95.1	78.6	88.3	77.9	68.1	78.0	78.8
DeepLabV3+ [11]	98.2	84.9	92.7	**57.3**	62.1	65.2	68.6	78.9	92.7	63.5	95.3	82.3	62.8	95.4	**85.3**	89.1	80.9	64.6	77.3	78.8
Ours (GSCNN)	**98.3**	**86.3**	**93.3**	55.8	**64.0**	**70.8**	**75.9**	**83.1**	**93.0**	**65.1**	**95.2**	**85.3**	**67.9**	**96.0**	80.8	**91.2**	**83.3**	**69.6**	**80.4**	**80.8**

Table 1: Comparison in terms of IoU vs state-of-the-art baselines on the Cityscapes val set.

Thrs	Method	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
12px	DeepLabV3+	**92.3**	80.4	87.2	59.6	53.7	83.8	75.2	81.2	90.2	60.8	90.4	76.6	78.7	91.6	**81.0**	87.1	92.6	**81.8**	78.0	80.1
	Ours	92.2	**81.7**	**87.9**	**59.6**	**54.3**	**87.1**	**82.3**	**84.4**	**90.9**	**61.1**	**91.9**	**80.4**	**82.8**	**92.6**	78.5	**90.0**	**94.6**	79.1	**82.2**	**81.8**
9px	DeepLabV3+	91.2	78.3	84.8	58.1	52.4	82.1	73.7	79.5	87.9	59.4	89.5	74.7	76.8	90.0	**80.5**	86.6	92.5	**81.0**	75.4	78.7
	Ours	**91.3**	**80.1**	**86.0**	**58.5**	**52.9**	**86.1**	**81.5**	**83.3**	**89.0**	**59.8**	**91.1**	**79.1**	**81.5**	**91.5**	78.1	**89.7**	**94.4**	78.5	**80.4**	**80.7**
5px	DeepLabV3+	88.1	72.6	78.1	55.0	49.1	77.9	69.0	74.7	81.0	55.8	86.4	69.0	71.9	85.4	**79.4**	85.4	92.1	**79.4**	68.4	74.7
	Ours	**88.7**	**75.3**	**80.9**	**55.9**	**49.9**	**83.6**	**78.6**	**80.4**	**83.4**	**56.6**	**88.4**	**75.4**	**77.8**	**88.3**	77.0	**88.9**	**94.2**	76.9	**75.1**	**77.6**
3px	DeepLabV3+	83.7	65.1	69.7	52.2	46.2	72.0	62.8	67.7	71.8	52.0	80.9	61.5	66.4	78.8	**78.2**	83.9	91.7	**77.9**	60.9	69.7
	Ours	**85.0**	**68.8**	**74.1**	**53.3**	**47.0**	**79.6**	**74.3**	**76.2**	**75.3**	**53.1**	**83.5**	**69.8**	**73.1**	**83.4**	75.8	**88.0**	**93.9**	75.1	**68.5**	**73.6**

Table 2: Comparison vs baselines at different thresholds in terms of boundary F-score on the Cityscapes val set.

approximate the argmax operator with a softmax with temperature τ:

$$\frac{\partial \arg\max_k p(y^k)}{\partial \eta_i} = \nabla_{\eta_i} \frac{\exp((\log p(y_k) + g_k)/\tau)}{\sum_j \exp((\log p(y_j) + g_j)/\tau)} \tag{9}$$

where $g_j \sim$ Gumbel(0,I) and τ a hyper-parameter. The operator $\nabla G*$ can be computed by filtering with Sobel kernel.

4. Experimental Results

In this section, we provide an extensive evaluation of each component of our framework on the challenging Cityscapes dataset [13]. We further show the effectiveness of our approach for several backbone architectures and provide qualitative results of our method.

Baselines. We use DeepLabV3+ [11], as our main baseline. This consitutes the state-of-the-art architecture for semantic segmentation and pretrained models are available. In most of our experiments, we use our own PyTorch implementation of DeeplabV3+ which differs from [11] in the choice of the backbone architecture. Specifically, we use ResNet-50, ResNet-101 and WideResNet as the backbone architecture for our version of DeeplabV3+. For a fair comparison, when applicable, we refer to this as *Baseline* in our tables. Additionally, we also compare against published state-of-the-art-methods on the validation set and in the Cityscapes benchmark (test set).

Dataset. All of our experiments are conducted on the Cityscapes dataset. This dataset contains images from 27 cities in Germany and neighboring countries. It contains 2975 training, 500 validation and 1525 test images. Cityscapes additionally includes 20,000 additional coarse annotations (i.e., coarse polygons covering individual objects). Notice that we supervise our shape stream with boundary ground-truth, and thus the coarse subset is not ideal for our setting. We thus do not use coarse data in our experiments. The dense pixel annotations include 30 classes which frequently occur in urban street scenes, out of which 19 are used for the actual training and evaluation. We follow [55, 56, 1] to generate the ground truth boundaries and supervise our shape stream.

Evaluation Metrics. We use three quantitative measures to evaluate the performance of our approach. **1)** We use the widely used intersection over union (IoU) to evaluate whether the network accurately predicts regions. **2)** Since our method aims to predict high-quality boundaries, we include another metric for evaluation. Specifically, we follow the boundary metric proposed in [42] to evaluate the quality of our semantics boundaries. This metric computes the F-score along the boundary of the predicted mask, given a small slack in distance. In our experiments, we use thresholds 0.00088, 0.001875, 0.00375, and 0.005 which correspond to 3, 5, 9, and 12 pixels respectively. Similarly to the IoU calculation, we also remove void areas during the computation of the F-score. Since boundaries are not pro-

Metric	Method	ResNet-50	ResNet-101	Wide-ResNet
mIoU	Baseline	71.3	72.7	79.2
	+ GCL	72.9	74.3	79.8
	+ Gradients	**73.0**	**74.7**	**80.1**
F-Score	Baseline	68.5	69.8	73.0
	+ GCL	71.7	**73.3**	**75.9**
	+ Gradients	**71.7**	73.0	75.6

Table 3: Comparison of the shape stream, GCL, and additional image gradient features (Canny) for different regular streams. Score on Cityscapes val (%) represents mean over all classes and F-Score represents boundary alignment (th=5px).

Method	th=3px	th=5px	th=9px	th=12px
Baseline	64.1	69.8	74.8	76.7
GCL	65.0	70.8	75.9	77.8
+ Dual Task	**68.0**	**73.0**	**77.2**	**78.8**

Table 4: Effect of the Dual Task Loss at difference thresholds in terms of boundary quality (F-score). ResNet-101 used in regular stream.

Base Network	Param Δ (%)	Perf Δ (mIoU)	Perf Δ (mF)
ResNet-50	+0.43	+1.7	+3.2
ResNet-101	+0.29	+2.0	+3.5
WideResNet38	+0.13	+0.9	+2.1

Table 5: Performance improvements and the percentage increase in the number of parameters due to the shape stream on different base networks.

vided for the test-set, we use the validation set to compute F-Scores as a metric for boundary alignment. **3)** We use distance-based evaluation in terms of IoU, explained below, in order to evaluate the performance of the segmentation models at varying distances from the camera.

Distance-based Evaluation. We argue that high accuracy is also important for small (distant) objects, where however, the global IoU metric does not well reflect this. Thus, we take crops of varying size around an approximate (fixed) vanishing point as a proxy for distance. In our case, this is performed by cropping 100 pixels along each image side except for the top, and the center of the resulting crop is our approximate vanishing point. Then, given a predefined cropping factor c , crops are applied such that: we crop c from the top and bottom and $c \times 2$ from the left and right. Intuitively, a smaller centered crop puts a larger weighting on the smaller objects farther away from the camera. An illustration of the procedure is shown in Fig 3. Fig 4 shows example predictions in each of the crops, illustrating how the metrics can focus on evaluating object at different sizes.

Implementation Details. In most of our experiments, we follow the methodology of Deeplab v3+ [11] but use simpler encoders as described in the experiments. All our networks are implemented in PyTorch. We use 800×800 as the training resolution and synchronized batch norm. Training is done on an NVIDIA DGX Station using 8 GPUs with a total batch size of 16. For Cityscapes, we use a learning rate of 1e-2 with a polynomial decay policy. We run the training for 100 epochs for the ablation purposes, and showcase our best results in Table 1 at 230 epochs. For our joint loss, we set $\lambda_1 = 20$, $\lambda_2 = 1$, $\lambda_3 = 1$ and $\lambda_4 = 1$. We set $\tau = 1$ for the Gumbel softmax. All our experiments are conducted in the Cityscapes fine set.

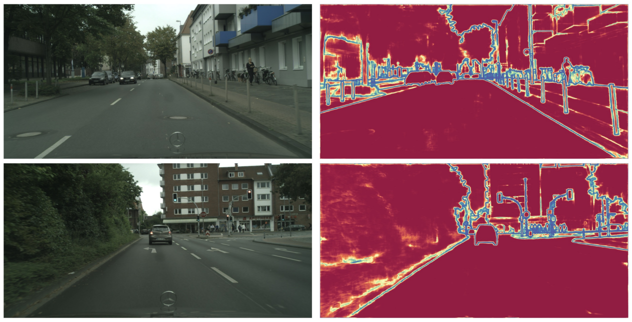

Figure 6: Example output of shape stream fed into the fusion module.

4.1. Quantitative Evaluation

In Table 1, we compare the performance of our GSCNN against the baselines in terms of region accuracy (measured by mIoU). The numbers are reported on the validation set, and computed on the full image (no cropping). In this metric, we achieve a 2% improvement, which is a significant result at this level of performance. In particular, we notice that we obtain significant improvements for small objects: motorcycles, traffic signs, traffic lights, and poles.

Table 2, on the other hand, compares the performance of our method against the baseline in terms of boundary accuracy (measured by F score). Similarly, our model performs considerably better, outperforming the baseline by close to 4% in the strictest regime. Note that, for fair comparison, we only report models trained on the Cityscapes fine set. Inference for all models is done on a single-scale.

In Fig 5, we show the performance of our method vs baseline following the proposed distance-based evaluation method. Here, we find that GSCNN performs increasingly better compared to DeeplabV3+ as the crop factor increases. The gap in performance between GSCNN and DeeplabV3+ increases from 2% at crop factor 0 (i.e. no cropping) to close to 6% at crop factor 400. This confirms that our network achieves significant improvements for smaller objects located further away from the camera.

Cityscapes Benchmark. To get optimal performance on the test set, we use our best model (*i.e.*, regular stream is WideResNet pretrained on the Mapillary dataset [40]). Training is done on an NVIDIA DGX Station using 8 GPUs with a total batch size of 16. We train this network with GCL and dual task loss for 175 epochs with a learning rate of 1e-2 with a polynomial decay policy. We also use a uniform sampling scheme to retrieve a 800×800 crop that uniformly samples from all classes. Additionally, we use a multi-scale inference scheme using scales 0.5, 1.0 and 2.0. We **do not use coarse data** during training, due to our boundary loss which requires fine boundary annotation.

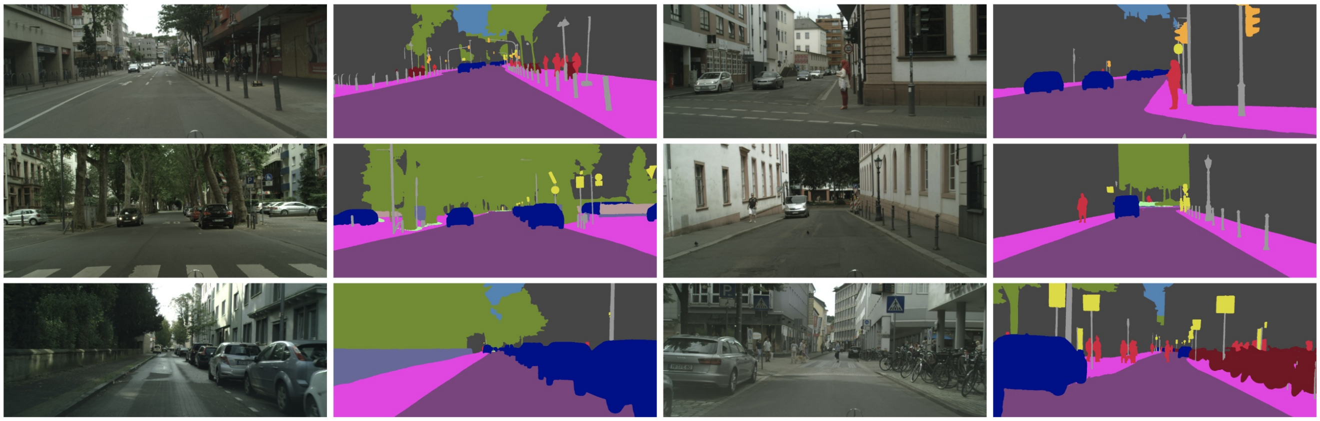

Figure 7: Qualitative results of our method on the Cityscapes **test set**. Figure shows the predicted segmentation masks.

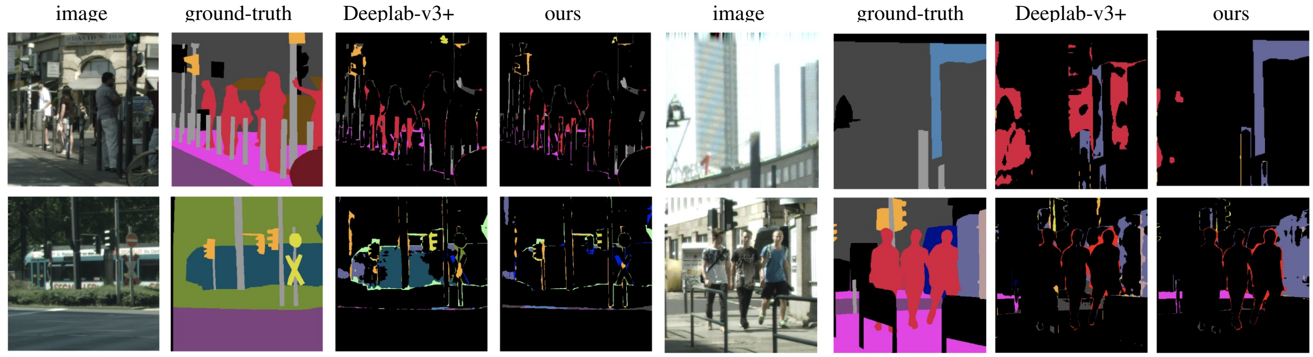

Figure 8: Qualitative comparison in terms of **errors** in predictions. Notice that our method produces more precise boundaries, particularly for smaller and thiner objects such as poles. Boundaries around people are also sharper.

Method	Coarse	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
PSP-Net [59]	✓	98.7	86.9	93.5	58.4	63.7	67.7	76.1	80.5	93.6	72.2	95.3	86.8	71.9	96.2	77.7	91.5	83.6	70.8	77.5	81.2
DeepLabV3 [10]	✓	98.6	86.2	93.5	55.2	63.2	70.0	77.1	81.3	93.8	72.3	95.9	87.6	73.4	96.3	75.1	90.4	85.1	72.1	78.3	81.3
DeepLabV3+ [11]	✓	98.7	87.0	93.9	59.5	63.7	71.4	78.2	82.2	94.0	73.0	95.8	88.0	73.3	96.4	78.0	90.9	83.9	73.8	78.9	81.9
AutoDeepLab-L [34]	✓	98.8	87.6	93.8	61.4	64.4	71.2	77.6	80.9	94.1	72.7	96.0	87.8	72.8	96.5	78.2	90.9	88.4	69.0	77.6	82.1
DPC [7]	✓	98.7	87.1	93.8	57.7	63.5	71.0	78.0	82.1	94.0	73.3	95.4	88.2	**74.5**	**96.5**	**81.2**	**93.3**	**89.0**	**74.1**	79.0	82.7
AAF-PSP [25]		98.5	85.6	93.0	53.8	59.0	65.9	75.0	78.4	93.7	72.4	95.6	86.4	70.5	95.9	73.9	82.7	76.9	68.7	76.4	79.1
TKCN [50]		98.4	85.8	93.0	51.7	61.7	67.6	75.8	80.0	93.6	72.7	95.4	86.9	70.9	95.9	64.5	86.9	81.8	79.6	77.6	79.5
Ours (GSCNN)		**98.7**	**87.4**	**94.2**	**61.9**	**64.6**	**72.9**	**79.6**	**82.5**	**94.3**	**74.3**	**96.2**	**88.3**	74.2	96.0	77.2	90.1	87.7	72.6	**79.4**	**82.8**

Table 6: Comparison vs state-of-the-art methods (with/without coarse training) on the Cityscapes test set. We only include published methods.

In Table 6, we compare against published state-of-the-art methods on the Cityscapes benchmark, evaluated on the test set. It is important to stress that our model is not trained on coarse data. Impressively, we can see that our model consistently outperforms very strong baselines, some of which also use extra coarse training data. At the time of this writing, our approach is also ranked as first among the published methods that do not use coarse data.

4.2. Ablation

In Table 3, we evaluate the effectiveness of each component of our method using different encoder networks for the regular stream. For fairness, comparison in this table is performed with respect to our own implementation of the baseline (i.e DeepLabV3+ with different backbone architectures), trained from scratch using the same set of hyperparameters and ImageNet initialization. Specifically, we use ResNet-50, ResNet-101 and Wide-ResNet for the backbone architectures. Here, GCL denotes a network trained with the shape stream with dual task loss, and Gradients denotes the network that also adds image gradients before the fusion layer. In our network, we use a Canny edge detector to retrieve such gradients. We see from the table that we achieve between 1 to 2 % improvement in performance in terms of mIoU, and around 3 % for boundary alignment.

Table 4, on the other hand, showcases the effect of the Dual Task loss in terms of F-score for boundary alignment. We illustrate its effect at three different thresholds. Here, GCL denotes the network with the GCL shape stream trained without Dual Task Loss. With respect to the base-

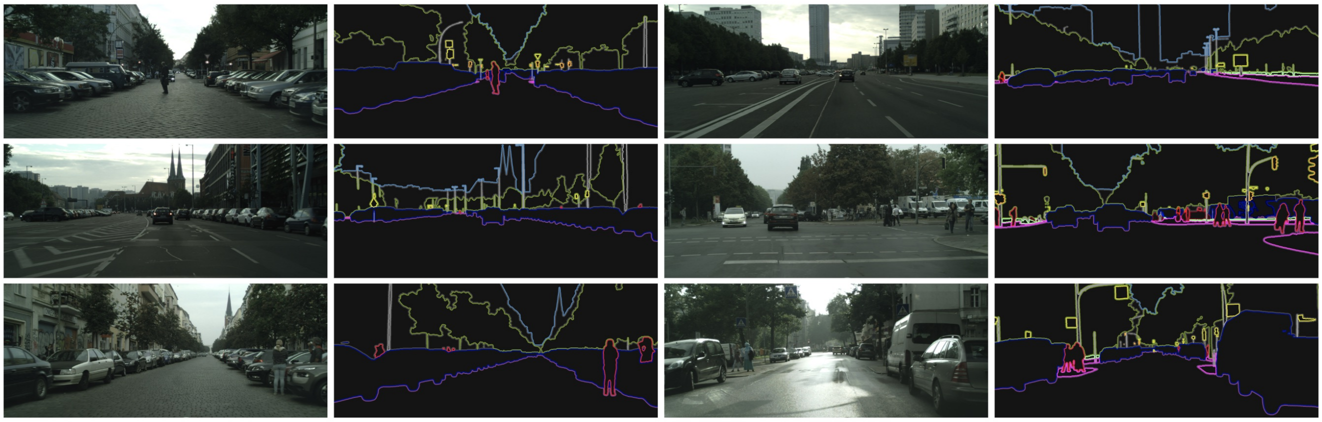

Figure 9: Qualitative results on the Cityscapes test set showing the high-quality boundaries of our predicted segmentation masks. Boundaries are obtained by finding the edges of the predicted segmentation masks.

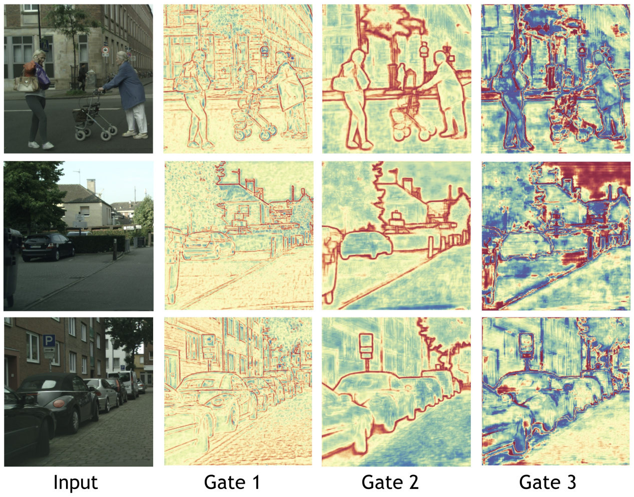

Figure 10: Visualization of the alpha channels from the GCLs.

line, we can observe that the dual loss significantly improves the performance of the model in terms of boundary accuracy. Concretely, by adding the Dual-Task loss, we see up to 3% improvement at the strictest regime.

4.3. Qualitative Results

In Figure 7, we provide qualitative results of our method on the Cityscapes test set. We compare our method to the baseline by highlighting typical cases where our methods excels in Figure 8. Specifically, we visualize the prediction *errors* for both methods. In these zoomed images, we can see a group of people standing around an area densely populated by poles. Here, Deeplab v3+ fails to capture the poles and naively classifies them as humans. Conversely, we can see that in our model poles are properly classified, and the error boundaries for pedestrians also thin out. Additionally, objects such as traffic lights, which are typically predicted as an over compromising blob in Deeplab v3+ (especially at higher distances) retain their shape and structure in the output of our model.

Fig 10 provides a visualization of the alpha channels from the GCL. We can notice how the gates help to emphasize the difference between the boundary/region areas in the incoming feature map. For example, the first gate emphasized very low-level edges while the second and third focus on object-level boundaries. As the result of gating, we obtain a final boundary map in the shape stream which accurately outlines objects and stuff classes. This stream learns to produce high quality class-agnostic boundaries which are then fed to the fusion module. Qualitative results of the output of the shape stream are shown in Fig 6.

In Figure 9, on the other hand, we show the boundaries obtained from the final segmentation masks. Notice their accuracy on the thinner and smaller objects.

5. Conclusion

In this paper, we proposed Gated-SCNN (GSCNN), a new two-stream CNN architecture for semantic segmentation that wires shape into a separate parallel stream. We used a new gating mechanism to connect the intermediate layers and a new loss function that exploits the duality between the tasks of semantic segmentation and semantic boundary prediction. Our experiments show that this leads to a highly effective architecture that produces sharper predictions around object boundaries and significantly boosts performance on thinner and smaller objects. Our architecture achieves state-of-the-art results on the challenging Cityscapes dataset, significantly improving over strong baselines.

Acknowledgments. We thank Karan Sapra for sharing their DeepLabV3+ implementation.

References

[1] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *CVPR*, 2019. 5

[2] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018. 1

[3] Anurag Arnab and Philip H.S. Torr. Bottom-up instance segmentation using deep higher-order crfs. In *arXiv:1609.02583*, 2016. 2

[4] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Semantic segmentation with boundary neural fields. In *CVPR*, pages 3602–3610, 2016. 2

[5] Siddhartha Chandra and Iasonas Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *ECCV*, pages 402–418. Springer, 2016. 2

[6] Liang-Chieh Chen, Jonathan T Barron, George Papandreou, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *CVPR*, pages 4545–4554, 2016. 2

[7] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NIPS*, pages 8713–8724, 2018. 7

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *T-PAMI*, 40(4):834–848, April 2018. 2, 5

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015. 2

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 7

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1, 2, 3, 5, 6, 7

[12] Dongcai Cheng, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Fusionnet: Edge aware deep convolutional networks for semantic segmentation of remote sensing harbor images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(12):5769–5783, 2017. 2

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 5

[14] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, pages 933–941. JMLR. org, 2017. 2

[15] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. Superpixel convolutional networks using bilateral inceptions. In *ECCV*, pages 597–613. Springer, 2016. 1, 2

[16] Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (ToG)*, volume 30, page 69. ACM, 2011. 2

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 1

[18] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, pages 519–534. Springer, 2016. 5

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 3

[20] Xuming He and Stephen Gould. An Exemplar-based CRF for Multi-instance Object Segmentation. In *CVPR*, 2014. 2

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 1

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1

[23] Varun Jampani, Martin Kiefel, and Peter V Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *CVPR*, pages 4452–4461, 2016. 2

[24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[25] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X Yu. Adaptive affinity fields for semantic segmentation. In *ECCV*, pages 587–602, 2018. 7

[26] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018. 2

[27] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, pages 6129–6138, 2017. 2

[28] Shu Kong and Charless C Fowlkes. Recurrent scene parsing with perspective understanding in the loop. In *CVPR*, pages 956–965, 2018. 2

[29] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011. 2

[30] David C. Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. *CVPR*, pages 2136–2143, 2009. 1

[31] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, pages 1925–1934, 2017. 2

[32] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, pages 3194–3203, 2016. 2, 5

[33] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019. 1

[34] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. *arXiv preprint arXiv:1901.02985*, 2019. 7

[35] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NIPS*, pages 1520–1530, 2017. 1, 2

[36] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *ICCV*, pages 1377–1385, 2015. 2

[37] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, 2015. 1, 2

[38] Jitendra Malik and Dror E. Maydan. Recovering three-dimensional shape from a single image of curved objects. *T-PAMI*, 11(6):555–566, 1989. 1

[39] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016. 2

[40] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 6

[41] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, pages 4353–4361, 2017. 2

[42] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 5

[43] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. *CVPR*, 2017. 1, 2

[44] Alex Schwing and Raquel Urtasun. Fully Connected Deep Structured Networks. arXiv:1503.02351, 2015. 2

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[46] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018. 2

[47] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *NIPS*, pages 4790–4798, 2016. 2

[48] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 1

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 2

[50] Tianyi Wu, Sheng Tang, Rui Zhang, and Jintao Li. Tree-structured kronecker convolutional networks for semantic segmentation. *arXiv preprint arXiv:1812.04945*, 2018. 7

[51] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015. 4

[52] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016. 1

[53] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, pages 2403–2412, 2018. 1

[54] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2

[55] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. CASENet: Deep category-aware semantic edge detection. In *CVPR*, 2017. 4, 5

[56] Zhiding Yu, Weiyang Liu, Yang Zou, Chen Feng, Srikumar Ramalingam, BVK Vijaya Kumar, and Jan Kautz. Simultaneous edge alignment and learning. In *ECCV*, 2018. 5

[57] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 2, 3

[58] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *CVPR*, 2016. 1

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2, 5, 7

[60] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015. 2