In International Conference on Computer Vision (ICCV), Seoul, Korea, 2019

Efficient Segmentation: Learning Downsampling Near Semantic Boundaries

Dmitrii Marin* Zijian He† Peter Vajda† Priyam Chatterjee†
Sam Tsai† Fei Yang‡ Yuri Boykov*

*University of Waterloo, Canada
{d2marin,yboykov}@uwaterloo.ca

*Vector Research Institute, Canada
{zijian,vajdap,priyamc,sstsai}@fb.com

†Facebook Inc, USA

‡TAL Education, China
yang.fei@100tal.com

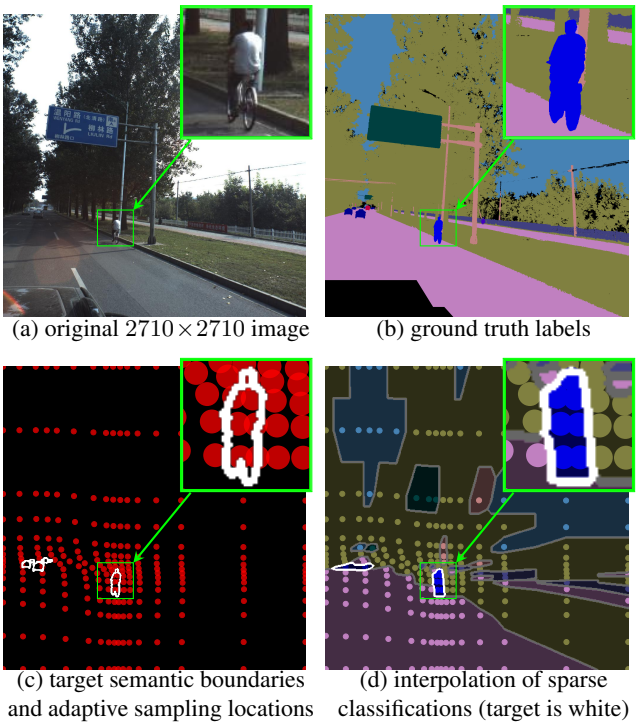
Abstract

Many automated processes such as auto-piloting rely on a good semantic segmentation as a critical component. To speed up performance, it is common to downsample the input frame. However, this comes at the cost of missed small objects and reduced accuracy at semantic boundaries. To address this problem, we propose a new content-adaptive downsampling technique that learns to favor sampling locations near semantic boundaries of target classes. Cost-performance analysis shows that our method consistently outperforms the uniform sampling improving balance between accuracy and computational efficiency. Our adaptive sampling gives segmentation with better quality of boundaries and more reliable support for smaller-size objects.

1. Introduction

Recent progress in hardware technology has made running efficient deep learning models on mobile devices possible. This has enabled many on-device experiences relying on deep learning-based computer vision systems. However, many tasks including semantic segmentation still require downsampling of the input image trading off accuracy in finer details for better inference speed [25, 55]. We show that uniform downsampling is sub-optimal and propose an alternative content-aware adaptive downsampling technique driven by semantic boundaries. We hypothesize that for better segmentation quality more pixels should be picked near semantic boundaries. With this intuition, we formulate a neural network model for learning content-adaptive sampling from ground truth semantic boundaries, see Fig. 1.

The advantages of our non-uniform downsampling over the uniform one are two-fold. First, the common uniform downsampling complicates accurate localization of boundaries in the original image. Indeed, assuming N uniformly sampled points over an image of diameter D, the distance between neighboring points gives a bound for the segmentation boundary localization errors $\mathcal{O}(\frac{D}{\sqrt{N}})$. In contrast, analysis in Appendix A shows that the error bound de-



(a) original 2710×2710 image (b) ground truth labels

(c) target semantic boundaries and adaptive sampling locations (d) interpolation of sparse classifications (target is white)

Figure 1: Illustration of our content-adaptive downsampling method on a high-resolution image (a). Given ground truth (b), we compute a non-uniform grid of sampling locations (red in (c)) pulled towards semantic boundaries of target classes (white in (c)). We use these points for training our *auxiliary network* to automatically produce such sparsely sampled locations. Their classification (colored dots in (d)) can be produced by a separately trained efficient low-res segmentation CNN. Concentration of sparse classifications near boundaries of target classes improves accuracy of interpolation (d) compared to uniform sampling, see Fig. 7.

creases significantly faster with respect to the number of sample points $\mathcal{O}(\frac{\kappa l^2}{N^2})$ assuming they are uniformly distributed near the segment boundary of max curvature κ and length l. Our non-uniform boundary-aware sampling approach selects more pixels around semantic boundaries reducing quantization errors on the boundaries.

Second, our non-uniform sampling implicitly accounts for scale variation via reducing the portion of the downsampled image occupied by larger segments and increasing that of smaller segments. It is well-known that presence of the same object class at different scales complicates automatic image understanding [6–8, 18, 19, 23, 45, 54, 56]. Thus, the scale equalizing effect of our adaptive downsampling simplifies learning. As shown in Fig. 1(c,d), our approach samples many pixels inside the cyclist, while the uniform downsampling may miss that person all together.

With the proposed content-adaptive sampling, a semantic segmentation system consists of three parts, see Fig.2. The first is our non-uniform downsampling block trained to sample pixels near semantic boundaries of target classes. The second part segments the downsampled image and can be based on practically any existing segmentation model. The last part upsamples the segmentation result producing a segmentation map at the original (or any given) resolution. Since we need to invert the non-uniform sampling, standard CNN interpolation techniques are not applicable.

Our contributions in this paper are as follows:

- We propose adaptive downsampling aiming at accurate representation of targeted semantic boundaries. We use an efficient CNN to reproduce such downsampling.
- Most segmentation architectures can benefit from non-uniform downsampling by incorporating our content-adaptive sampling and interpolation components.
- We apply our framework to semantic segmentation and show consistent improvements on many architectures and datasets. Our cost-performance analysis accounts for the computational overhead. We also analyze improvements from our adaptive downsampling at semantic boundaries and on objects of different sizes.

Sec. 2 provides an overview of prior works. Sec. 3 describes our approach in details. Sec. 4 compares many state-of-the-art semantic segmentation architectures with uniform and our adaptive downsampling on multiple datasets.

2. Prior work

Semantic segmentation requires a class assignment for each pixel in an image. This problem is important for many automated navigational applications. We first review some related literature on this topic. We then provide a brief review of some relevant non-uniform sampling methods.

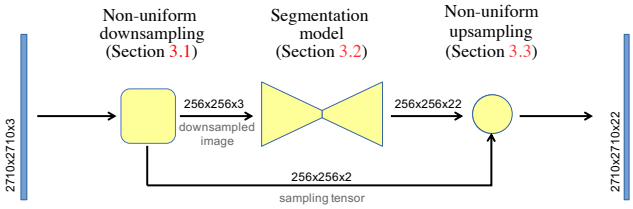


Figure 2: Proposed efficient segmentation architecture with adaptive downsampling. The first block (detailed in Fig.4) takes a high-res image and outputs sampling locations (*sampling tensor*) and a downsampled image. The resolutioin of 256×256 is an example, in the experiments we tested resolutions in the range 32×32 to 512×512. The downsampled image is then segmented by some standard model. Finally, the result is upsampled to the original resolution.

Many segmentation networks are built upon basic image classification networks, *e.g.* [6–8, 35, 52, 56]. These approaches modify the base model to produce dense higher resolution features maps. For example, Long *et al.* [35] used *fully convolutional network* [32] and trainable deconvolution layers. They also note that *algorithme à trous* [24] is a way to increase resolution of the feature maps. This idea was studied in [6] where *dilated convolutions* allow removal of max pooling layers from a trained model producing higher resolution feature maps with larger field of view without the need to retrain the model.

Segmentation models built upon classification models inherit one limiting property, that is the base classification models [22, 31, 48] tend to have many features in the deeper layers. That results in an extensive resources consumption when increasing the resolution of later feature maps (using for example *algorithme à trous* [24]). As a result, the final output is typically chosen to be of lower resolution, with an interpolation employed to upscale the final score map.

The alternative direction for segmentation (and more generally for pixel-level prediction) is based on "hourglass models" that first produce low resolution deep features and then gradually upsample the features employing common network operations and skip connections [5, 38, 39, 43].

The need and advantage of aggregating information from different scales have been long recognized in the computer vision literature [6–8, 18, 19, 23, 45, 53, 54, 56]. One way to tackle the multiscale challenge is to first detect the location of objects and then segment the image using either a cropped original image [19, 53] or cropped feature maps [18, 21]. These two-stage approaches separate the problems of scale learning and segmentation, making the latter task easier. As a result the accuracy of segmentation improves and instance level segmentation is straightforward. However, such an approach comes with a significant computational cost when many objects are present since each object

	two-stage [18, 19, 21]	ours Sec. 3	single-stage [6, 35, 43]
accuracy	++	+	-
speed	-	+	++
multi-object speed	- -	+	++
simplicity	-	+	++
multi-scale	++	+	-
boundary precision	++	+	-

Table 1: Segmentation approaches with pros & cons (+/-).

needs to be segmented individually. Our method improves upon the single-stage approach for a small computational cost and, thus, is positioned in between of these two approaches. Table 1 outlines pros and cons of our approach compared with two-stage and single-stage methods.

Spatial Transformer Networks [27, 41] learn spatial transformations (warping) of the CNN input. They explore different parameterizations for spatial transformation including affine, projective, splines [27] or specially designed saliency-based layers [41]. Their focus is to undo different data distortions or to "zoom-in" on salient regions, while our approach is focused on efficient downsampling retaining as much information around semantic boundaries as possible. They do not use their approach in the context of pixel-level predictions (*e.g.* segmentation) and do not consider the inverse transformations (Sec. 3.3 in our case).

Deformable convolutions [11, 28] augment the spatial sampling locations in the standard convolutions with additional adaptive offsets. In their experiments the deformable convolutions replace traditional convolutions in the last few layers of the network making their approach complementary to ours. The goal is to allow the new convolution to pick the features from the best locations in the previous layer. Our approach focuses on choosing the best locations in the original image and thus has access to more information.

Other complementary approaches include skipping some layers at some pixels [17] and early stopping of network computation for some spatial regions of the image [16, 33]. Similarly, these methods modify computation at deeper network layers and do not concern image downsampling.

3. Boundary Driven Adaptive Downsampling

Fig. 2 shows three main stages of our system: content-adaptive downsampling, segmentation and upsampling. The downsampler, described in Sec. 3.1, determines non-uniform sampling locations and produces a downsampled image. The segmentation model then processes this (non-uniformly) downsampled image. We can use any existing segmentation model for this purpose. The results are treated as sparsely classified locations in the original image. The third part, described in Sec. 3.3, uses interpolation to recover segmentation at the original resolution, see Fig. 1(d).

Let us introduce notation. Consider a high-resolution image $I = \{I_{ij}\}$ of size $H \times W$ with C channels. Assuming relative coordinate system, all pixels have spatial coordinates that form a uniform grid covering square $[0,1]^2$. Let $I[u,v]$ be the value of the pixel that has spatial coordinates closest to (u,v) for $u,v \in [0,1]$. Consider tensor $\phi \in [0,1]^{h\times w\times 2}$. We denote elements of ϕ by ϕ^c_{ij} for $c \in \{0,1\}$, $i \in \{1,2,\dots,h\}$, $j \in \{1,2,\dots,w\}$. We refer to such tensors as *sampling tensors*. Let ϕ_{ij} be the point $(\phi^0_{ij}, \phi^1_{ij})$. Fig. 1(c) shows an example of such points.

The sampling operator

$$\mathbb{R}^{H\times W\times C} \times [0,1]^{h\times w\times 2} \;\rightarrow\; \mathbb{R}^{h\times w\times C}$$

maps a pair of image I and sampling tensor ϕ to the corresponding sampled image $J = \{J_{ij}\}$ such that

$$J_{ij} := I[\phi^0_{ij}, \phi^1_{ij}]. \tag{1}$$

The uniform downsampling can be defined by a sampling tensor $u \in [0,1]^{h\times w\times 2}$ such that $u^0_{ij} = (i-1)/(h-1)$ and $u^1_{ij} = (j-1)/(w-1)$.

3.1. Sampling Model

Our non-uniform sampling model should balance between two competing objectives. On one hand we want our model to produce finer sampling in the vicinity of semantic boundaries. On the other hand, the distortions due to the non-uniformity should not preclude successful segmentation of the non-uniformly downsampled image.

Assume for image I (Fig. 1(a)) we have the ground truth semantic labels (Fig. 1(b)). We compute a boundary map (white in Fig. 1(c)) from the semantic labels. Then for each pixel we compute the closest pixel on the boundary. Let $b(u_{ij})$ be the spatial coordinates of a pixel on the semantic boundary that is the closest to coordinates u_{ij} (distance transform). We define our content-adaptive non-uniform downsampling as sampling tensor ϕ minimizing the energy

$$E(\phi) = \sum_{i,j} \|\phi_{ij} - b(u_{ij})\|^2 + \lambda \sum_{\substack{|i-i'|+\\ |j-j'|=1}} \|\phi_{ij} - \phi_{i'j'}\|^2 \tag{2}$$

subject to *covering constraints*

$$\begin{aligned} &\phi \in [0,1]^{h\times w\times 2} \\ &\phi^0_{1\,j} = 0 \quad \& \quad \phi^0_{hj} = 1, \quad 1 \le j \le w, \\ &\phi^1_{i\,1} = 0 \quad \& \quad \phi^1_{iw} = 1, \quad 1 \le i \le h. \end{aligned} \tag{3}$$

The first term in (2) ensures that sampling locations are close to semantic boundaries, while the second term ensures that the spatial structure of the sampling locations is not distorted excessively. The constraints provide that the sampling locations cover the entire image. This least squares problem with convex constraints can be efficiently solved globally via a set of sparse linear equations. Red dots in Figs. 1(c) and 3 illustrate solutions for different values of λ.

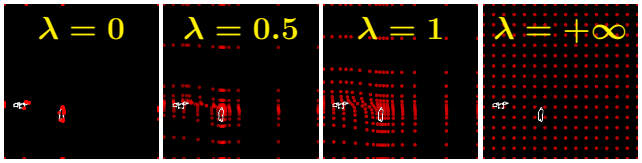


Figure 3: Boundary driven sampling for different λ in (2). Extreme λ sample either semantic boundaries (left) or uniformly (right). Middle-range λ yield in-between sampling.

We train a relatively small auxiliary network to predict the sampling tensor without boundaries. The auxiliary network can be significantly smaller than the base segmentation model as it solves a simpler problem. It learns cues indicating presence of the semantic boundaries. For example, the vicinity of vanishing points is more likely to contain many small objects (and their boundaries). Also, small mistakes in the sampling locations are not critical as the final classification decision is left for the segmentation network.

As an auxiliary network, we propose two U-Net [43] sub-networks stacked together (Fig. 6). The motivation for stacking sub-networks is to model the sequential processes of boundary computation and sampling points selection. We train this network with squared L2 loss between the network prediction and a tensor "proposal" $\tilde{\phi} = \arg\min_{\phi} E(\phi)$ minimizing (2) subject to (3)[1]. Alternatively, one can directly use objective (2) as a regularized loss function [50, 51]. Our proposal generation approach can be seen as a one step of ADM procedure for such a loss [37].

Once the sampling tensor is computed the original image is downsampled via sampling operator (1). Application of sampling tensor ϕ of size $(h, w, 2)$ yields sampled image of size $h \times w$. If this is not the desired size $h' \times w'$ of downsampled image, we still can employ ϕ for sampling. To that end, we obtain a new sampling tensor ϕ' of shape $(h', w', 2)$ by resizing ϕ using bilinear interpolation, see example in Fig.5.

Fig.4 shows the architecture of our downsampling block.

3.2. Segmentation Model

Our adaptive downsampling can be used with any off-the-shelf segmentation model as it does not place any constraints on the base segmentation model. Our improved results with base multiple models (U-Net [43], PSP-Net [56] and Deeplabv3+ [8]) in Sec. 4 showcase this versatility.

3.3. Upsampling

In keeping with prior work, we assume that the base segmentation model produces a final score map of the same size as its downsampled input. Thus, we need to upsample the output to match the original input resolution. In case of standard downsampling this step is a simple upscaling, commonly performed via bilinear interpolation. In our case, we need to "invert" the non-uniform transformation. Covering constraints (3) ensure that the convex hall of the sampling locations covers the entire image, thus we can use interpolation to recover the score map at the original resolution. We use Scipy [2] to interpolate the unstructured multi-dimensional data, which employs Delaunay [12] triangulation and barycentric interpolation within triangles [47].

An important aspect of our content-adaptive downsampling method in Sec. 3.1 is that it preserves the grid topology. Thus, an efficient implementation can skip the triangulation step and use the original grid structure. The interpolation problem reduces to a computer graphics problem of rendering a filled triangle, which can be efficiently solved by Bresenham's algorithm [47].

[1]The network prediction is projected onto constraints (3) during testing.

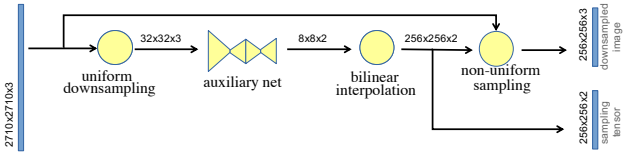


Figure 4: Architecture of non-uniform downsampling block in Fig. 2. A high-resolution image (*e.g.* 2710×2710) is uniformly downsampled to a small image (*e.g.* 32×32) and then processed by an auxiliary network producing sampling locations stored in a *sampling tensor*. This tensor is resized (see Fig. 5) to a desired resolution (*e.g.* 256×256) and then used for non-uniform downsampling.

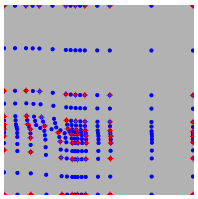


Figure 5: An example of 8×8 sampling locations (red crosses) produced by an auxiliary network and the result of resizing the corresponding *sampling tensor* by the factor of 2 (blue points) via bilinear interpolation, see Fig. 4.

4. Experiments

In this section we describe several experiments with our adaptive downsampling for semantic segmentation on many high-resolution datasets and state-of-the-art approaches. Figure 7 shows a few qualitative examples.

4.1. Experimental Setup

Dataset and evaluation. We evaluate and compare the proposed method on several public semantic segmentation datasets. Computational requirements of the contemporaneous approaches and the cost of annotations conditioned the low resolution of images or imprecise (rough) annotations in popular semantic segmentation datasets, such as Caltech [15], [3], Pascal VOC [13, 14, 20], COCO [34]. With rapid development of autonomous driving, a number of new semantic segmentation datasets focusing on road

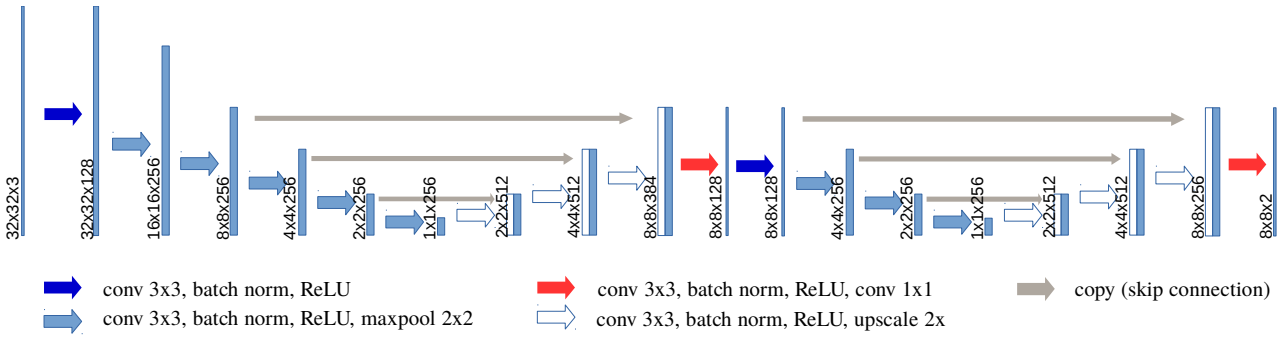


Figure 6: Double U-Net model for predicting sampling parameters. The depth of the first sub-network can vary (depending on the input resolution). The structure of the second sub-network is kept fixed. To improve efficiency, we use only one convolution (instead of two in [43]) in each block. The number of features is 256 in all layers except the first and the last one. We also use padded convolutions to avoid shrinking of feature maps, and we add batch normalization after each convolution.

scenes [10,26] or synthetic datasets [42,44] have been made available. These recent datasets provide high-resolution data and high quality annotations. In our experiments, we mainly focus on datasets with high-resolution images, namely ApolloScapes [26], CityScapes [10], Synthia [44] and Supervisely (person segmentation) [49] datasets.

The main evaluation metric is mean *Intersection over Union* (mIoU). The metric is always evaluated on segmentation results at the original resolution. We compare performance at various downsampling resolutions to emulate different operating requirements. Occasionally we use other metrics to demonstrate different features of our approach.

Implementation details: Our main implementation is in Caffe2 [1]. For both the non-uniform sampler network and segmentation network, we use Adam [29] optimization method with (base learning rate, #epochs) of (10^{-5}, 33), (10^{-4}, 1000), (10^{-4}, 500) for datasets ApolloScape, Supervisely, and Synthia, respectively. We employ exponential learning rate policy. The batch size is as follows:

input resolution	16	32	64	128	256	512
batch size	128	128	128	32	24	12

Experiments with PSP-Net [56] and Deeplabv3+ [8] use public implementations with the default parameters. MobileNetV2 [46] results are reported in [36].

In all experiments, we consider segmentation networks fed with uniformly downsampled images as our baseline. We replace the uniform downsampling with adaptive one as described in Sec. 3.1. The interpolation of the predictions follows Sec. 3.3 in both cases. The auxiliary network is separately trained with ground truth produced by (2) where we set $\lambda = 1$. The auxiliary network predicts a sampling tensor of size $(8, 8, 2)$, which is then resized to a required downsampling resolution. During training of the segmentation network we do not include upsampling stage (for both baseline and proposed models) but instead downsample the label map. We use the softmax-entropy loss.

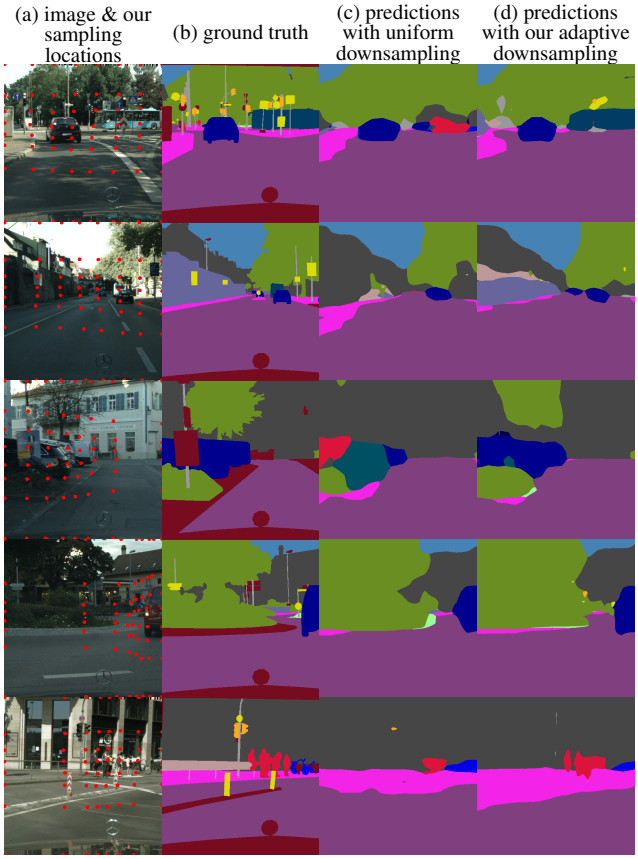


Figure 7: Examples from Cityscapes [10] *val* set. **(a)**: original images and non-uniform 8×8 sampling tensor produced by our trained auxiliary net in Fig. 4 (to avoid clutter, 128×128 tensor interpolation, as in Fig. 5, is not shown). **(c)**: results of PSP-Net [56] with uniform 128×128 downsampling. **(d)**: results of the same network *with* our adaptive 128×128 downsampling based on (a). High-res segmentation results in (c,d) are interpolated (Sec. 3.3) classifications for uniformly or adaptively downsampled pixels.

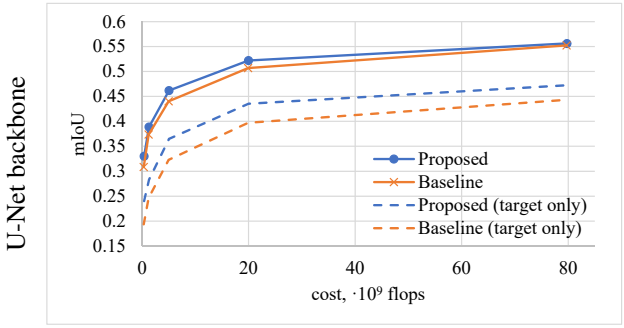


Figure 8: Cost-performance analysis on **ApolloScape** dataset. Proposed method performs better than the baseline method. With the same cost we can achieve higher quality.

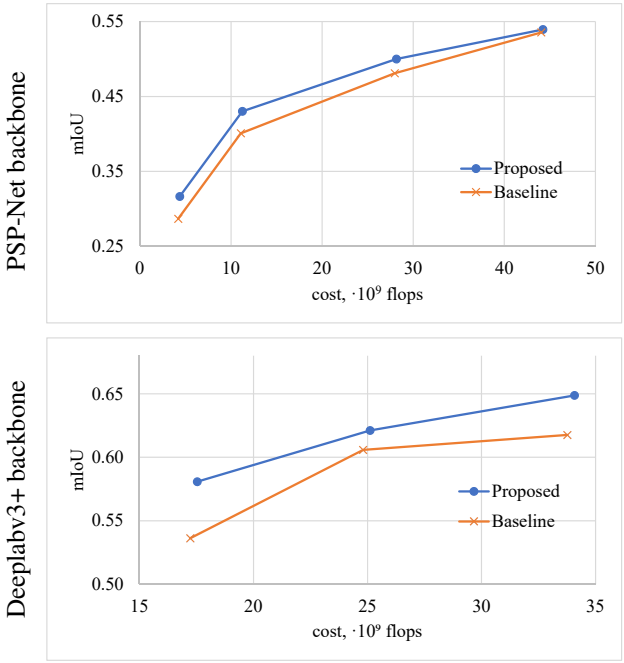


Figure 9: Cost-performance analysis on **CityScapes** with PSP-Net and Deeplabv3+ baselines for varying downsampling size, see Tab. 3. Our content-adaptive downsampling gives better results with the same computational cost.

During training we randomly crop largest square from an image. For example, if the original image is 3384×2710 we select a patch of size 2710×2710. During testing we crop the central largest square. Additionally, during training we augment data by random left-right flipping, adjusting the contrast, brightness and adding salt-and-pepper noise.

4.2. Cost-performance Analysis

ApolloScape [26] is an open dataset for autonomous driving. The dataset consists of approximately 105K training and 8K validation images of size 3384×2710. The annotations contain 22 classes for evaluation. The annotations of some classes (cars, motorbikes, bicycles, persons, riders, trucks, buses and tricycles) are of high quality. These occupy 26% of pixels in evaluation set. We refer to these as *target classes*. Other classes annotations are noisy. Since the noise in pixel labels greatly magnifies the noise of segments boundaries, we chose to define our sampling model based on the *target* classes boundaries. This exploits an important aspect of our method, *i.e.* an ability to focus on boundaries of specific semantic classes of interest. Following [26] we give separate metrics for these classes.

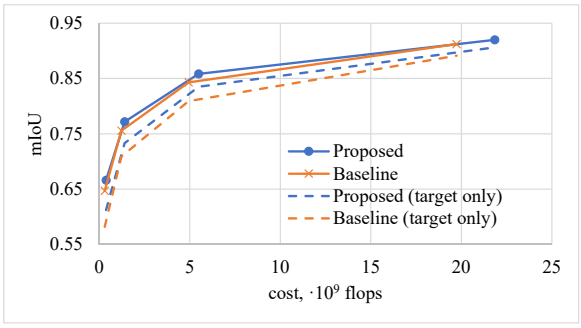


Figure 10: Cost-performance analysis on **Synthia** dataset. Our approach performs better for target classes (with a tie on all classes).

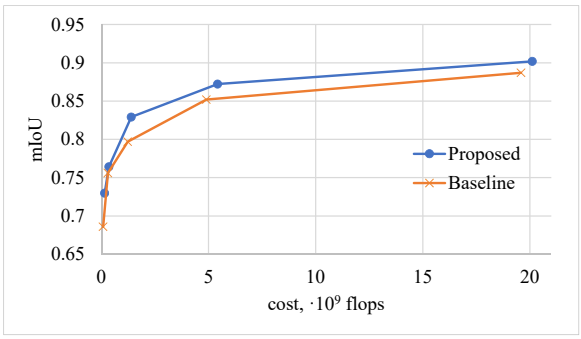


Figure 11: Cost-performance analysis on **Supervisely** dataset. Our approach improves quality of segmentation.

Our adaptive downsampling based on semantic boundaries improves segmentation, see Tab. 2. Our approach achieves a mIoU gain of 3-5% for target classes and up to 2% overall. This improvement comes at negligible computational cost. Our approach consistently produces better results even under fixed computational budgets, see Fig. 8.

Focusing on quality of the boundaries for some target classes may lower performance on other classes. This gives one a flexibility of reflecting importance of certain classes over the others depending on the application.

CityScapes [10] is another commonly used open road scene dataset providing 5K annotated images of size 1024×2048 with 19 classes in evaluation. Following the same test

	downsample resolution	flops, $\cdot 10^9$	non-target classes, IoU														**target classes**, IoU								mIoU	
			road	sidewalk	traffic cone	road pile	fence	traffic light	pole	traffic sign	wall	dustbin	billboard	building	vegatation	sky	**car**	**motor-bicycle**	**bicycle**	**person**	**rider**	**truck**	**bus**	**tricycle**	all classes	**target classes**
Ours	32	0.38	**0.92**	**0.38**	**0.17**	**0.00**	**0.49**	0.11	0.08	0.44	**0.28**	**0.03**	**0.00**	0.74	0.86	0.84	**0.66**	**0.07**	**0.27**	**0.02**	**0.03**	**0.34**	**0.52**	**0.01**	**0.24**	**0.24**
Baseline	32	**0.31**	0.92	0.29	0.13	0.00	0.43	**0.14**	**0.11**	**0.53**	0.18	0.00	0.00	**0.74**	**0.87**	**0.89**	0.59	0.04	0.26	0.01	0.02	0.20	0.44	0.00	0.19	0.19
Ours	64	1.31	0.94	0.39	**0.31**	**0.02**	**0.56**	0.25	0.17	0.61	**0.41**	**0.08**	**0.00**	0.78	0.89	0.87	**0.76**	**0.10**	**0.33**	**0.04**	**0.03**	**0.44**	**0.53**	**0.04**	**0.28**	**0.28**
Baseline	64	**1.24**	**0.94**	**0.40**	0.30	0.01	0.52	**0.30**	**0.22**	**0.64**	0.29	0.04	0.00	**0.79**	**0.90**	**0.91**	0.70	0.06	0.31	0.02	0.03	0.32	0.52	0.03	0.25	0.25
Ours	128	5.05	0.95	**0.51**	**0.43**	**0.07**	**0.61**	0.44	0.29	0.71	**0.47**	**0.13**	**0.01**	0.82	0.91	0.88	**0.83**	**0.16**	**0.41**	**0.08**	**0.05**	**0.57**	**0.76**	0.06	**0.36**	**0.36**
Baseline	128	**4.98**	**0.96**	0.39	0.43	0.05	0.59	**0.45**	**0.36**	**0.73**	0.37	0.11	0.00	**0.83**	**0.92**	**0.93**	0.80	0.10	0.38	0.06	0.03	0.44	0.70	**0.06**	0.32	0.32
Ours	256	19.99	0.96	0.44	**0.51**	**0.13**	**0.66**	**0.58**	0.42	0.78	**0.58**	**0.27**	**0.00**	0.84	0.92	0.89	**0.88**	**0.21**	**0.47**	**0.18**	0.04	**0.65**	0.80	**0.24**	**0.44**	**0.44**
Baseline	256	**19.92**	**0.97**	**0.48**	0.49	0.13	0.64	0.58	**0.46**	**0.79**	0.48	0.24	0.00	**0.85**	**0.94**	**0.94**	0.86	0.17	0.42	0.15	**0.04**	0.60	**0.83**	0.10	0.40	0.40
Ours	512	79.76	0.97	0.44	0.54	**0.21**	0.68	0.63	0.49	0.80	**0.67**	0.36	0.00	0.85	0.93	0.90	**0.91**	**0.24**	**0.52**	**0.30**	**0.06**	**0.75**	0.81	**0.19**	**0.47**	**0.47**
Baseline	512	**79.68**	**0.97**	**0.47**	**0.55**	0.20	**0.68**	**0.67**	**0.54**	**0.83**	0.59	**0.36**	**0.00**	**0.87**	**0.94**	**0.94**	0.90	0.21	0.49	0.26	0.03	0.68	**0.84**	0.13	0.44	0.44

Table 2: Per class results on the validation set of ApolloScape. Our adaptive sampling improves overall quality of segmentation. Target classes (bold font on the top row) consistently benefit for all resolutions.

	downsample resolution	**auxiliary net resolution**	**flops,** $\cdot 10^9$	**mIoU**	**downsample resolution**	**auxiliary net resolution**	**flops,** $\cdot 10^9$	**mIoU**
backbone	PSP-Net [56]				Deeplabv3+ [8]			
ours	64	32	4.37	**0.32**	160	32	17.54	**0.58**
baseline		-	4.20	0.29		-	17.23	0.54
ours	128	32	11.25	**0.43**	192	32	25.12	**0.62**
baseline		-	11.08	0.40		-	24.81	0.61
ours	256	32	44.22	**0.54**	224	32	34.08	**0.65**
baseline		-	44.05	0.54		-	33.77	0.62

Table 3: CityScapes results with different backbones.

	downsample resolution	**flops,** $\cdot 10^9$	**all classes**	**target classes**
ours	32	0.38	**0.67**	**0.61**
baseline		0.31	0.65	0.58
ours	64	1.40	**0.77**	**0.73**
baseline		1.23	0.76	0.71
ours	128	5.49	**0.86**	**0.83**
baseline		4.93	0.84	0.81
ours	256	21.85	**0.92**	**0.91**
baseline		19.74	0.91	0.89

Table 4: Synthia results (mIoU). With the same input resolution our approach improves the segmentation quality.

	downsample resolution	**flops,** $\cdot 10^9$	**mIoU**	**back-ground**	**person**
ours	16	0.15	**0.73**	**0.84**	**0.62**
baseline		0.07	0.69	0.81	0.56
ours	32	0.35	**0.76**	**0.86**	**0.67**
baseline		0.30	0.76	0.85	0.66
ours	64	1.39	**0.83**	**0.90**	**0.76**
baseline		1.22	0.80	0.88	0.71
ours	128	5.42	**0.87**	**0.93**	**0.82**
baseline		4.90	0.85	0.91	0.79
ours	256	20.11	**0.90**	**0.94**	**0.86**
baseline		19.59	0.89	0.93	0.84

Table 5: Supervisely results. With the same input resolution our approach improves the segmentation quality.

protocol, we evaluated our approach using PSP-Net [4, 56] (with ResNet50 [22] backbone) and Deeplabv3+ [8] (with Xception65 [9] backbone) as the base segmentation model. The mIoU results are shown in Tab. 3 and Fig. 9 where we again see consistent improvements of up to 4%.

Synthia [44] is a synthetic dataset of 13K HD images taken from an array of cameras moving randomly through a city. The results in Tab. 4 show that our approach improves upon the baseline model. The cost-performance analysis in Fig. 10 shows that our method improves segmentation quality of target classes by 1.5% to 3% at negligible cost.

Person segmentation The Supervisely Person Dataset [49] is a collection of 5711 high-resolution images with 6884 high-quality annotated person instances. The dataset set contains pictures of people taken in different conditions, including portraits, land- and cityscapes. We have randomly split the dataset into training (5140) and testing subsets (571). The dataset has only two labels: person and background. Segmentation results for this dataset are shown in Tab. 5 with a cost-performance analysis with respect to the baseline shown in Fig. 11. The experiment shows absolute mIoU increases up to 5.8%, confirming the advantages of non-uniform downsampling for person segmentation tasks as well.

4.3. Boundary Accuracy

We design an experiment to show that our method improves boundary precision. We adopt a standard trimap approach [30] where we compute the classification accuracy within a band (called trimap) of varying width around boundaries of segments. We compute the trimap plots for two input resolutions in Fig. 13 for person segmentation dataset described above. Our methods improves mostly in the vicinity of semantic boundaries. Interestingly, for the input resolution of 64×64 the maximum accuracy improvement is reached around trimap width of 4 pixels. This may be attributed to the fact that downsampling model in

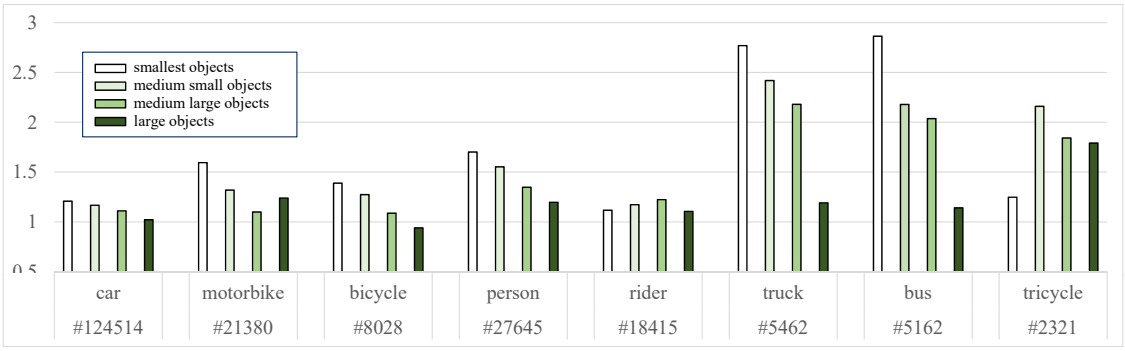


Figure 12: Average recall of objects broken down by object classes and sizes on the validation set of ApolloScapes. Values are expressed relative to the baseline. All objects of a class were split into 4 equally sized bins based on objects' area. Smaller bin number correspond to objects of smaller size. The total number of objects in each class is marked by "#". As well as in Fig. 14 there is negative correlation between object sizes and relative recall for all classes except rare "rider" and "tricycle".

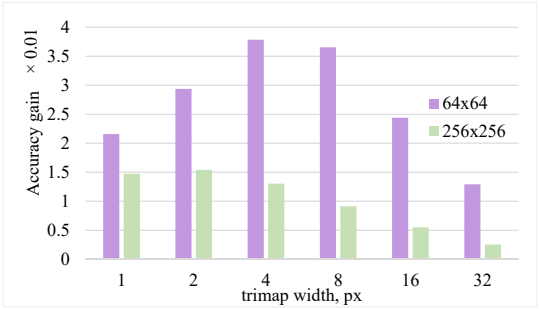


Figure 13: Absolute accuracy difference between our method and baseline near semantic boundaries on Supervisely data for sampling resolutions 64×64 and 256×256.

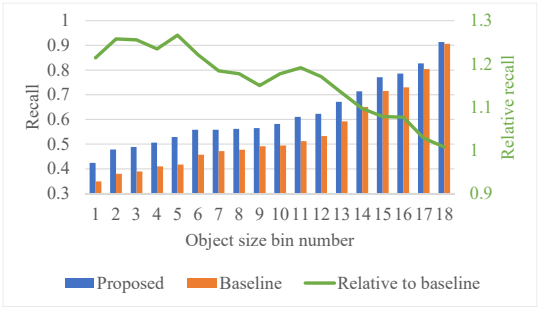


Figure 14: Average recall of objects of different sizes. All objects in the validation set of ApolloScapes were grouped into several equally sized bins by their area. A smaller bin number corresponds to smaller objects. Downsample resolution is 64×64. We improve baseline more on smaller objects. The green curve (right vertical axis) shows that the relative recall (the average recall of baseline is taken for 1) is negatively correlated with the object sizes.

Sec. 3.1 does not depend on downsampling resolution and essentially defines the same sampling tensor for all sizes of downsampled image. Thus, the distances between neighboring points for 64×64 sampling locations are approximately 4 times larger than the respective distances for 256×256 sampling locations. This leads to reduced gain of accuracy within narrow trimaps.

4.4. Effect of Object Size

Since our adaptive downsampling is trained to select more points around semantic boundaries, it implicitly provides larger support for small objects. This results in better performance of the overall system on these objects. Instance level annotations allow us to verify this by analyzing quality statistics with respect to individual objects. This is in contrast to usual pixel-centric segmentation metrics (mIoU or accuracy). *E.g.*, the *recall* of a segmentation of an object is defined as ratio of pixels classified correctly (pixel predicted to belong to the true object class) to the total number of pixels in the object[2]. Fig. 12 and 14 show the improvement of recall over baseline for objects of different sizes and categories. Our method degrades more gracefully than the uniform downsampling as the object size decreases.

[2]Recall usually comes together with *precision*. Since segmentation does not have instance labels, the object-level precision is undefined.

Conclusions

In this work, we described a novel method to perform non-uniform content-aware downsampling as an alternative method to uniform downsampling to reduce the computational cost for semantic segmentation systems. The adaptive downsampling parameters are computed by an auxiliary CNN that learns from a non-uniform sample geometric model driven by semantic boundaries. Although the auxiliary network requires additional computations, the experimental results show that the network improves segmentation performance while keeping the added cost low, providing a better cost-performance balance. Our method significantly improves performance on small objects and produces

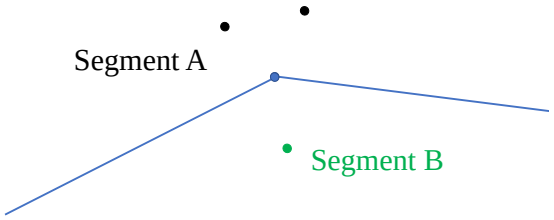


Figure 15: Three sampling points per corner is enough to build any piece-wise linear boundary.

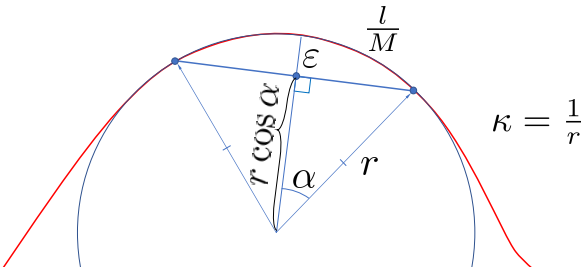


Figure 16: Illustration for the piece-wise linear approximation for a curve (red) where the ends of linear segments (blue) lie on the curve.

more precise boundaries. In addition, any off-the-shelf segmentation system can benefit from our approach as it is implemented as an additional block enclosing the system.

A potential future research direction is employing more advanced interpolation methods, similar to [40], which can further improve quality of the final result.

Finally, we note that our adaptive sampling may benefit other applications with pixel-level predictions where boundary accuracy is important and downsampling is used to reduce computational cost. This is left for future work.

Appendix A. Non-uniform Sampling Error

As stated in the submission the error bound decreases as $\mathcal{O}(\frac{\kappa l^2}{N^2})$ assuming N sampling points are uniformly distributed *near the segment boundary* where κ and l are the maximal curvature and length of the boundary respectively. To show the upper bound on the best approximation it suffices to show an example that provides $\mathcal{O}(\frac{\kappa l^2}{N^2})$ boundary approximation error.

Assuming commonly used linear interpolation method (which we use in the paper) the boundary of segments is piece-wise linear. Using N sampling points we can define any piece-wise linear curve with $M = N/3$ segments, see Fig. 15.

Let $f(s)$ be a curve of length l in $\mathbb{R}^2$ for $s_0 \leq s \leq s_1$ and $p(t)$ be its linear approximation with M segments. We define boundary approximation error ϵ as the maximal distance between the curve and its approximation:

$$\epsilon = \sup_t \inf_s \|p(t) - f(s)\|.$$

We place the ends of the segments of p exactly on curve f such that they are uniformly distributed. That is, each segment encloses a piece of the curve of length l/M. The error on each segments can be bounded by approximation error of an arc of radius $r = \frac{1}{\kappa}$ as shown in Fig. 16:

$$\varepsilon \leq \frac{1 - \cos\alpha}{\kappa} = \mathcal{O}\left(\frac{\kappa l^2}{M^2}\right)$$

where we used the facts

$$\alpha \cdot r = \frac{l}{2M} \quad \text{and} \quad \cos\alpha = 1 - \frac{\alpha^2}{2} + \mathcal{O}(\alpha^4).$$

This immediately leads to the bound $\epsilon = \mathcal{O}(\frac{\kappa l^2}{N^2})$.

References

[1] Caffe2: A New Lightweight, Modular, and Scalable Deep Learning Framework. https://caffe2.ai. 5

[2] SciPy is open-source software for mathematics, science, and engineering. https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.griddata.html. 4

[3] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(11):1475–1490, 2004. 4

[4] Oles Andrienko. ICNet and PSPNet-50 in Tensorflow for real-time semantic segmentation. https://github.com/oandrienko/fast-semantic-segmentation, 2018. 7

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, Dec 2017. 2

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2018. 2, 3

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. pages 801–818, 2018. 2, 4, 5, 7

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1251–1258, 2017. 7

[10] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 6

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 3

[12] Boris Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934. 4

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 4

[14] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. The 2005 pascal visual object classes challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 117–176. Springer, 2006. 4

[15] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007. 4

[16] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1039–1048, 2017. 3

[17] Mikhail Figurnov, Aizhan Ibraimova, Dmitry P Vetrov, and Pushmeet Kohli. Perforatedcnns: Acceleration through elimination of redundant convolutions. In *Advances in Neural Information Processing Systems*, pages 947–955, 2016. 3

[18] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2, 3

[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2, 3

[20] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*. IEEE, 2011. 4

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017. 2, 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016. 2, 7

[23] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Proceedings of the 2004 IEEE computer society conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–II. IEEE, 2004. 2

[24] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990. 2

[25] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1

[26] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *arXiv preprint arXiv:1803.06184*, 2018. 5, 6

[27] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 3

[28] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1846–1854. IEEE, 2017. 3

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[30] Pushmeet Kohli, Philip HS Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision (IJCV)*, 82(3):302–324, 2009. 7

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[32] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 2

[33] Xiaoxiao Li, Ziwei Liu, Ping Luo, Chen Change Loy, and Xiaoou Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3193–3202, 2017. 3

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 4

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3431–3440, 2015. 2, 3

[36] Dmitrii Marin. Efficient Segmentation: Learning Downsampling Near Semantic Boundaries (unofficial implementation). GitHub repo adaptive-sampling, August 2019. 5

[37] Dmitrii Marin, Meng Tang, Ismail Ben Ayed, and Yuri Boykov. Beyond gradient descent for regularized segmentation losses. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[38] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016. 2

[39] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1528, 2015. 2

[40] Pascal Peter, Sebastian Hoffmann, Frank Nedwed, Laurent Hoeltgen, and Joachim Weickert. From optimised inpainting with linear pdes towards competitive image compression codecs. In Thomas Bräunl, Brendan McCane, Mariano Rivera, and Xinguo Yu, editors, *Image and Video Technology*, pages 63–74, Cham, 2016. Springer International Publishing. 9

[41] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018. 3

[42] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2232–2241, 2017. 5

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3, 4, 5

[44] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes (SYNTHIA-Rand). In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5, 7

[45] Chris Russell, Pushmeet Kohli, Philip HS Torr, et al. Associative hierarchical crfs for object class image segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 739–746. IEEE, 2009. 2

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 5

[47] Peter Shirley, Michael Ashikhmin, and Steve Marschner. *Fundamentals of Computer Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2nd edition, 2005. 4

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[49] Supervise.ly. Releasing "Supervisely Person" dataset for teaching machines to segment humans. https://hackernoon.com/releasing-supervisely-person-dataset-for-teaching-machines-to-segment-humans-1f1fc1f28469, 2018. 5, 7

[50] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 507–522, 2018. 4

[51] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 4

[52] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016. 2

[53] Fangting Xia, Peng Wang, Liang-Chieh Chen, and Alan L Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *European Conference on Computer Vision (ECCV)*, pages 648–663. Springer, 2016. 2

[54] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2

[55] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. *arXiv preprint arXiv:1704.08545*, 2017. 1

[56] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. 2, 4, 5, 7