LEARNING TO DOWNSAMPLE FOR SEGMENTATION OF ULTRA-HIGH RESOLUTION IMAGES

Anonymous authors
Paper under double-blind review

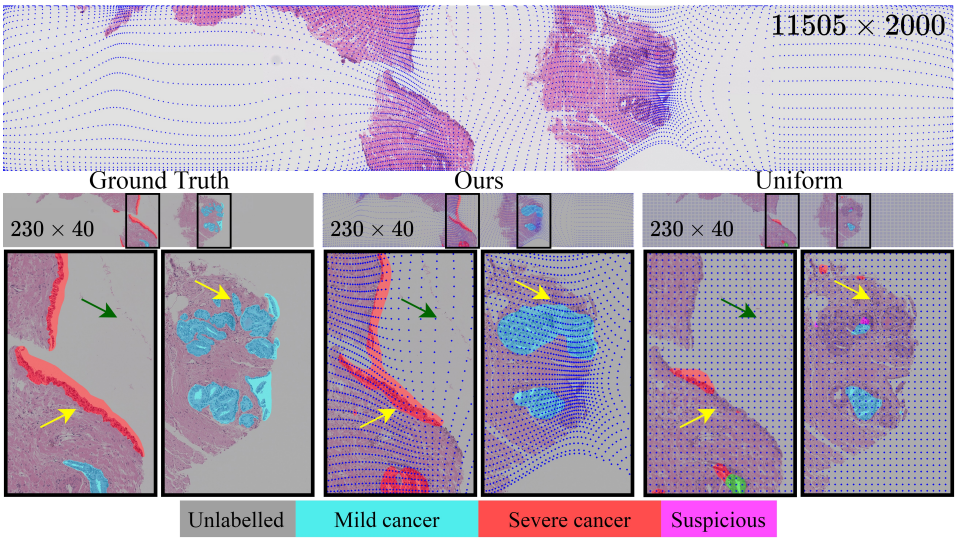


Figure 1: **Semantic segmentation with learnt deformed downsampling on cancerous histology image.** We propose a method for learning to downsample ultra high-resolution images (top-row) that reflects the importance of each location. The aim is to learn non-uniform downsampling (blue dots on the images) to preserve information that guides segmentation and ignore image content that does not contribute (e.g. dark green arrows) thereby enabling low-cost semantic segmentation on low-resolution downsampled images. This strategy helps identify small regions such as the cancerous regions with abnormal tissues (yellow arrows); see masked ground truth and predictions in middle-row. Compared to baseline standard uniform downsampling (referred to as "Uniform") our method (referred to as "Ours") samples more densely at important semantic regions (yellow arrows) and leads to more accurate segmentation (see bottom-row).

ABSTRACT

Segmentation of ultra-high resolution images with deep learning is challenging because of their enormous size, often consisting of millions or even billions of pixels. Common solutions drastically downsample the image uniformly to meet memory constraints, implicitly *assuming all pixels are equally important*. However this assumption is not true and compromises the performance of deep learning techniques that have proved powerful on standard-sized images. For example with uniform downsampling, see the arrow pointed regions in Fig. 1, the unhealthy cells (yellow arrows) do not have enough corresponding samples while the less informative "background" regions (green arrows) are oversampled, and lead to a negative effect on the segmentation prediction from the low-resolution downsampled image. In this work we show that learning the spatially varying downsampling strategy jointly with segmentation offers advantages in segmenting large images with limited computational budget. Fig. 1 shows that our method adapts the sampling density over different locations so that more samples are collected from the small important regions (yellow arrows) and less from the others (green arrows), which in turn leads to better segmentation accuracy. We show on two public and one local high-resolution datasets across different domains that our method consistently learns sampling locations preserving more information and leads to 1) boosted 3% to 10% absolute **mIoU** (up to 25% in class-wise **IoU**) and 2) better cost-performance trade-off saving up to 90% calculations over baseline methods. Code will be available here.

1 INTRODUCTION

Many automated processes such as auto-piloting, geospatial analysis and medical image processing rely on good semantic segmentation over ultra-high resolution images with millions or even billions of pixels. Exemplary applications include urban scene analysis with camera array images ($>25000\times14000$ pixels)(Wang et al., 2020), geospatial analysis with satellite images ($>5000\times5000$ pixels) (Maggiori et al., 2017) and histopathological analysis with whole slide images ($>10,000\times10,000$ pixels) (Srinidhi et al., 2019). Challenges arise when applying deep learning (DL) methods on those ultra-high resolution images, like those in Srinidhi et al. (2019); Chen et al. (2019); Marin et al. (2019), due to GPU memory constraints.

To speed up performance and adapt to memory requirements or data transmission latency, it is common to uniformly downsample the whole input images, then train the segmentation network and make predictions at lower resolution (locally or remotely), and finally upsample low-resolution prediction back to the original resolution (as summarised in Fig. 2 (a)). However, the uniform downsampling approach, which we refer to as "uniform", treats all pixels as equally important. This generally leads to reduced performance when directly applying DL approaches, which perform best without downsampling on datasets where images have more manageable size like in Chen et al. (2016); Sun et al. (2019). For example, in segmentation problems that require finding "needles in a haystack" — *a certain small positive foreground is more important than the rest large negative background* (e.g., like tumours in histology images as in Fig. 1), more samples should be drawn from foreground than background and uniform downsampling may be suboptimal.

Based on a similar intuition, Marin et al. (2019) recently postulated that for better segmentation quality more pixels should be picked near semantic boundaries and thus proposed to separately train an "edge-based" network from ground truth semantic boundaries for content-adaptive sampling, as summarised in Fig. 2 (b), and achieved state-of-the-art (SOTA) result. However the sampling density around boundaries depends on fixed hyper-parameters and the effectiveness of using a single "best fit" configuration is highly questionable given the large pattern variation in ultra-high resolution images. Secondly, it is unknown how well the "edge-based" (Marin et al., 2019) approach would perform when label boundaries are less informative. For example, cancerous regions in Fig. 1 or forest regions in satellite images (Demir et al., 2018) are annotated based on higher-level semantic information rather than nuclei or tree edges, respectively.

In this work, we start with a motivational experiment to investigate the barriers of current SOTA. We demonstrate that the segmentation performance of the "edge-based" method (Marin et al., 2019) with a fixed sampling density around edges is sub-optimal, as the optimal sampling density varies spatially according to local patterns in different areas. Secondly, motivated by this finding, we then introduce *deformation module*, a learnable downsampler, that is trained jointly in an end-to-end fashion with any *segmentation model* (Fig. 2 (c)) learns to adapt sampling density according to the local image contents. The downsampled image guided by the *deformation module* results in a deformed low-resolution image and hence the name. The *deformation module* processes a downsampled version of the given ultra high-resolution image to estimate a sampling density distribution hence lightweight and can be used to augment a wide range of existing segmentation architectures. We demonstrate the general utility of our approach using three datasets: DeepGlobe (Demir et al., 2018) and a local Prostate Cancer Histology dataset ("PCa-Histo") are datasets consisting of ultra high-resolution aerial and whole slide pathological images. We also use the popular Cityscapes dataset (Cordts et al., 2016) since it provides a familiar platform for demonstration and evaluation. In all datasets we show that a lightweight implementation of the *deformation module* can more flexibly adjust the sampling density according to local content than baseline methods. This can consistently boost the segmentation performance and save computational costs.

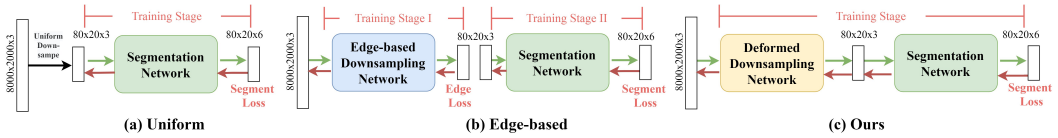


Figure 2: Comparing training pipelines of (a) "uniform" (b) "edge-based" (Marin et al., 2019) and (c) our proposed "deformation" approaches. Low resolution prediction are uniform/non-uniform upsampled to original resolution at inference.

2 RELATED WORKS

Different sampling strategies have been developed to improve the performance of CNNs. We group these methods into three main categories: feature aggregating and sampling, image sampling as pre-processing for classification, image sampling as pre-processing for segmentation.

2.1 FEATURE AGGREGATING AND SAMPLING

One key limitation of traditional CNNs is the fix-sized sampling window (i.e. convolutional kernel) restricted information "visible" to the network. Attempts have been made through either aggregating multiple scaled samplings or shifting sampling locations directly. With the former approach, various multi-scale architectures (Chen et al., 2014; 2016; Hariharan et al., 2015) have shown the importance of sampling multiple-scale information, but the aggregation process is less efficient because inputs/features need to be re-scaled multiple times. Dilated convolution (Yu & Koltun, 2015) shift sampling locations at multiple distances to collect multiscale information therefore avoided inefficient re-scaling. Such shifting is however limited to fixed geometric structures of sampling window, therefore deformable convolutional networks (DCNs) (Dai et al., 2017) propose to learn sampling offsets to augment each sampling location at feature level can improve segmentation performance. Although DCNs share similarities with our approach, they are complementary to ours because we focus on learning a better image downsampling as pre-processing so that 1) maximal information can be kept at minimal computation cost at segmentation and 2) it can be flexibly plug-in to existing segmentation networks.

2.2 IMAGE SAMPLING AS PRE-PROCESSING FOR CLASSIFICATION

With the intuition that per-image labelling can be predicted by an improved sampling of the image to reduce computation cost, several learnt image sampling methods have been developed. Spatial Transformer Networks (STN) (Jaderberg et al., 2015) introduce a layer that estimates a parametrized affine, projective and splines transformation from an input image to recover data distortions and thereby improve image classification accuracy. Recasens et al. (2018) proposed to jointly learn a saliency-based network and "zoom-in" to salient regions when downsampling an input image for classification. Similarly, Katharopoulos & Fleuret (2019) uses attention weights to guide sub-sampling an informative subset to reduce the computational burden. However, an end-to-end trained downsampling network has not been proposed so far for per-pixel segmentation. More importantly joint optimising the downsampling network for segmentation is more challenging than the per-image classification, due to potential oversampling at trivial locations, as we will show in this work, and hence we propose an edge-loss inspired by Marin et al. (2019) to regularise the training.

2.3 IMAGE SAMPLING AS PRE-PROCESSING FOR SEGMENTATION

On learning the sampling for image segmentation, the patch-based or region-proposal approaches aiming to learn optimal crop at each location to reduce the computational cost and improve accuracy, either separately (Girshick et al., 2014; He et al., 2017) or jointly (Jin et al., 2020). However this approach is less computational efficient at inference because each cropped region needs to be segmented individually. Non-uniform grid representation modifying the entire input images to alleviate the memory cost, such as meshes (Wang et al., 2018; Gkioxari et al., 2019), signed distance functions (Mescheder et al., 2019), and octrees (Tatarchenko et al., 2017). However none of those approaches is optimised specifically for segmentation. The closest work to ours is by Marin et al. (2019), who proposed reducing the computational cost by **separately training** an "edge-based" downsampling network, prior to segmentation, to encourage denser sampling around object edges, achieving SOTA results. However the "edge-based" approach is constrained by fixed sampling density around edges and is therefore less flexible when label edges are less informative. Here in this work we 1) first demonstrate empirically that the choice of sampling density around edges indeed considerably influences the segmentation performance across different locations and hence the barrier of SOTA; 2) Secondly, motivated by this finding, we then propose and demonstrate a **jointly trained** downsampling system that can adjust sampling more flexibly and boost the performance robustly, despite the variation of edge informativeness across different datasets.

3 METHODS

In this section, we first formulate the basic components associated with this study in section 3.1. We then perform a motivation experiment in section 3.2 to illustrate the impact of fixed sampling density around the edge on segmentation performance and its spatial variation across the image. Motivated by the finding, we propose the *deformation module*, the jointly trained downsampler for segmentation,

in section 3.3. To ease the joint optimisation challenge of oversampling at the trivial locations, a regularisation term is proposed in section 3.4.

3.1 PROBLEM FORMULATION

Sampling method: Let $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ be a high-resolution image of an arbitrary size H, W, C. A sampler $\mathbf{G}$ takes $\mathbf{I}$ as input and computes a downsampled image $\mathbf{J} = \mathbf{G}(\mathbf{I})$, where $\mathbf{J} \in \mathbb{R}^{h \times w \times C}$. Consider a relative coordinate system such that $\mathbf{I}[u, v]$ is the pixel value of $\mathbf{I}$ where $\boldsymbol{u}, \boldsymbol{v} \in [0,1]$. And an absolute coordinate system such that $\mathbf{J}[i, j]$ is the pixel value of $\mathbf{J}$ at coordinates $(\boldsymbol{i}, \boldsymbol{j})$ for $\boldsymbol{i} \in \{1,2,...h\}, \boldsymbol{j} \in \{1,2,...w\}$. Essentially, the sampler $\mathbf{G}$ computes a mapping between $(\boldsymbol{i}, \boldsymbol{j})$ and $(\boldsymbol{u}, \boldsymbol{v})$. Practically, sampler $\mathbf{G}$ contains two functions $\{\mathbf{g}^0, \mathbf{g}^1\}$, such that for a downsampled image $\mathbf{J} = \{\mathbf{J}[i, j]\}$ we have sampling operator[1]:

$$\mathbf{J}[i, j] := \mathbf{I}[\mathbf{g}^0(i, j), \mathbf{g}^1(i, j)] \tag{1}$$

Segmentation and non-uniform upsampling: In this work we discuss model agnostic downsampling and upsampling methods, and therefore any existing segmentation model can be applied. Here we denote the segmentation network $\mathbf{S}_\phi$, parameterised by ϕ, that takes as an input a downsampled image $\mathbf{J}$ and makes a prediction $\mathbf{S}_\phi(\mathbf{J})$ in the low-resolution space. The upsampling process consists of a reverse sampler $\mathbf{G^{-1}}()$ reverses the mapping of each pixel at coordinates $(\boldsymbol{i}, \boldsymbol{j})$ of the sampled image $\mathbf{J}$ back to coordinates $(\boldsymbol{u}, \boldsymbol{v})$ in the high-resolution domain, and an interpolator $\mathbf{Int}()$ calculates the missing pixels to get final prediction $\mathbf{Int}(\mathbf{G^{-1}}(\mathbf{J}))$. In this work, we use the nearest neighbour as $\mathbf{Int}()$.

3.2 MOTIVATIONAL STUDY: INVESTIGATING THE BARRIER OF THE SOTA

Disentangle the intrinsic interpolation error: Marin et al. (2019) assume sampling optimised by an "edge-based" objective can reduce segmentation error. While the evaluation (e.g. with $\mathbf{IoU}$) is applied only after interpolation involves a compound of segmentation and interpolation error makes their relationship unclear. We propose to disentangle the intrinsic interpolation error independent of the segmentation error by assuming a perfect segmentation at downsampled space. In specific we calculate $\mathbf{IoU}(\mathbf{Y}, \mathbf{Y}')$, where $\mathbf{Y}' = \mathbf{Int}(\mathbf{G^{-1}}(\mathbf{G}(\mathbf{Y})))$ indicting the same non-uniform downsampling/upsampling process is applied to the label $\mathbf{Y}$.

Barrier of the SOTA: The "edge-based" approach (Marin et al., 2019), separately train a non-uniform down-sampler $\mathbf{G_e}$ minimizing two competing energy terms: "sampling distance to semantic boundaries" and "sampling distance to uniform sampling locations", as the first and second term in equation 2[2]. **A temperature term λ is used to balance the two energies, whose value is empirically recommend to 1 by Marin et al. (2019) and decided the sampling density around edge**.

$$\mathbf{E}(\mathbf{G_e}(i, j))] = \sum_{i,j} \|\mathbf{G_e}(i, j) - \mathbf{b}[i, j]\| + \lambda \sum_{i,j} \|\mathbf{G_e}(i, j) - \mathbf{G_e}(i', j')\|^2 \tag{2}$$

The first part of our work performs an empirical analysis to investigate a key question: "How does the empirically tuned temperature term λ affect the segmentation performance?". To simplify the problem we perform binary segmentation from the 19-classes Cityscapes dataset (Cordts et al., 2016). We generate a set of simulated "edge-based" samplers each at different sampling density around edge (i.e. λ)[3] and directly calculate $\mathbf{IoU}(\mathbf{Y}, \mathbf{Y}')$ to investigate the interpolation error, best performed sampler is shown as each dot on Fig. 3 (a). A set of different segmentation networks can then be trained with each of the above "edge-based" samplers with $\mathbf{IoU}$ of final predictions shown in Fig. 3 (b). It is clear that there is no "one-size-fits-all" sampling density configuration that leads to the best performance for all individual classes, neither for intrinsic interpolation error ($\mathbf{IoU}(\mathbf{Y}, \mathbf{Y}')$) or segmentation error ($\mathbf{IoU}$). Fig. 3 (c) illustrates visually the variation of the segmentation performance. These observations altogether imply that the SOTA "edge-based" sampling scheme with a pre-set sampling density around the edge is sub-optimal, highlighting the potential benefits of a more intelligent strategy that can adaptively configure the most informative non-uniform sampling configuration to describe the local patterns at a given location.

3.3 DEFORMATION MODULE: LEARN NON-UNIFORM DOWNSAMPLING FOR SEGMENTATION

Motivated by the above finding, we introduce the *deformation module*, a data-driven sampling method that learns spatially varying non-uniform downsampling at each spatial location according to its importance to the downstream segmentation task. Fig.4 provides a schematic of the proposed method.

[1] The "uniform" approach will have sampler $\mathbf{G_u} = \{\mathbf{g_u}^0(i, j) = (i-1)/(h-1), \mathbf{g_u}^1(i, j) = (j-1)/(w-1)\}$.

[2] $\mathbf{b}[i, j]$ is the spatial coordinates of a pixel on the semantic boundary closest to $(\boldsymbol{u}[i, j], \boldsymbol{v}[i, j])$

[3] We simulate the impact of λ to sampler as shown by Marin et al. (2019), with more details in Appendix A.4. The reason is our experiments used original non-square inputs instead of the square input reported by Marin et al. (2019).

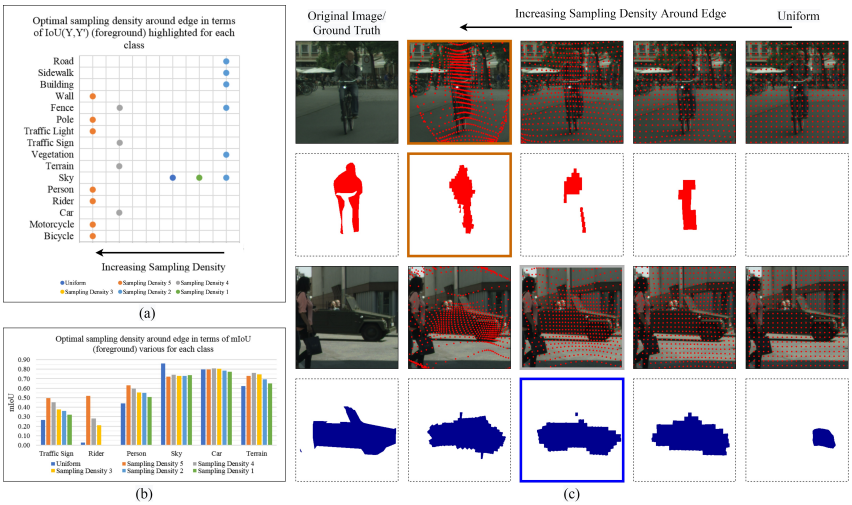


Figure 3: Quantitative (a-b) and qualitative (c) evidences that the optimal sampling density around edge varies over classes hence implies the variation over locations. (a): $\mathbf{IoU}(\mathbf{Y},\mathbf{Y}')$ with different "edge-based" samplers each at different sampling density around edge and best performed for each class is highlighted. (b): $\mathbf{IoU}$ of trained segmentation with above "edge-based" samplers. (c) Visual comparison of ground truth and predictions with various sampling density for two example classes. Sampling locations are masked on original image in red dots. All experiments performed on binary segmentation on 1/10th subset of Cityscapes (Cordts et al., 2016) with downsampled size of 64×128 given input of 1024×2048. Note three classes are under-represented thus not reported here.

The *deformation module* takes a low-resolution version of a high-resolution input image and estimates the deformed sampler (a non-uniform downsampler) that adapts the sampling density at different pixel locations. Then, the *segmentation network* processes the input downsampled image based on the output of *deformation module*, and estimates the corresponding segmentation probabilities.

For each high-resolution image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, we compute its lower resolution version $\mathbf{I}_{\text{lr}} \in \mathbb{R}^{h_d \times w_d \times C}$. The *deformation module*, D_θ, parametrised by θ, takes the low-resolution image $\mathbf{I}_{\text{lr}}$ as input and generates a deformation map $\mathbf{d} = D_\theta(\mathbf{I}_{\text{lr}})$, where $\mathbf{d} \in \mathbb{R}^{h_d \times w_d \times 1}$, that predicts the sampling density at each pixel location. Next, the deformed sampler $\mathbf{G_d}$ is constructed by taking both $\mathbf{I}$ and $\mathbf{d}$ as input and computes the downsampled image $\mathbf{J} = \mathbf{G_d}(\mathbf{I},\mathbf{d})$, where $\mathbf{J} \in \mathbb{R}^{h \times w \times C}$. Following the same downsampled segmentation pipeline defined earlier, the deformed sampled image $\mathbf{J}$ can be fed into the segmentation network, $\mathbf{S}_\phi(\mathbf{J})$, parametrised by ϕ to estimate the corresponding segmentation probabilities. We note here at training the segmentation loss is calculated in the low-resolution space without upsampling by getting the downsampled label $\mathbf{G_d}(\mathbf{Y},\mathbf{d})$ with the same deformed sampler.

Following earlier definitions in Equation 1, the deformed sampler $\mathbf{G_d}$ essentially computes a mapping between $(\boldsymbol{i},\boldsymbol{j})$ from sampled image $\mathbf{J}$ and $(\boldsymbol{u},\boldsymbol{v})$ from original image $\mathbf{I}$, such that $\boldsymbol{u}[i,j] = \mathbf{g}^0(i,j)$, $\boldsymbol{v}[i,j] = \mathbf{g}^1(i,j)$. As mentioned, we would like to train the deformed sampler $\mathbf{G_d}$ who samples $\mathbf{I}$ denser at high-importance regions based on the sampling density indicated in the deformation map $\mathbf{d}$. Inspired by Recasens et al. (2018) we consider the value at each pixel (i',j') in the deformation map $\mathbf{d}$ as an attractive force $\mathbf{d}(i',j')$ that is pulling surrounding pixels. The force acts on an expanded region defined by a distance kernel[4] $k((i,j),(i',j'))$ so that:

$$\mathbf{g}^0(i,j) = \frac{\sum_{i',j'} \mathbf{d}(i',j') k((i,j),(i',j')) i'}{\sum_{i',j'} \mathbf{d}(i',j') k((i,j),(i',j'))}, \qquad \mathbf{g}^1(i,j) = \frac{\sum_{i',j'} \mathbf{d}(i',j') k((i,j),(i',j')) j'}{\sum_{i',j'} \mathbf{d}(i',j') k((i,j),(i',j'))} \tag{3}$$

This formulation holds certain desirable properties that fit our goal: 1) the learnable *deformation module* produces a deformation map with varying sampling density at different regions, the need for this was illustrated in the motivational studies in section 3.2; 2) the *deformation module* can naturally fit CNN architecture and preserve differentiability for it to be jointly trained with downstream segmentation loss.

[4]Practically the distance kernel is a fixed Gaussian convolution with given radius. This radius decided the distance the attraction force can act on, while the degree of force is learnt though deformation map. Similar to Recasens et al. (2018), we set this radius of at least one fourth of the short axis of deformation map.

This way the sampling density at each location is optimised by segmentation performance rather than a one-size-fits-all solution hence break the barriers of the "edge-based" approach (Marin et al., 2019)).

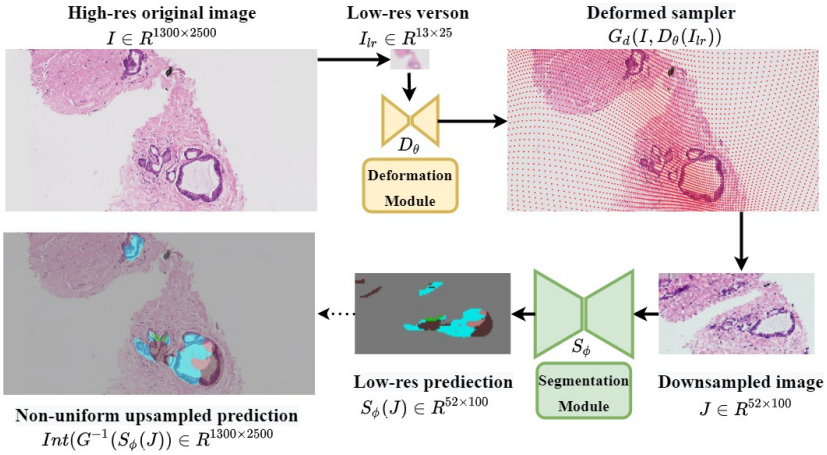


Figure 4: **Architecture schematic.** The deformation module produces the deformation map $\mathbf{d} = D_\theta(\mathbf{I}_{\text{lr}})$ that predicts sampling importance weights at different pixel locations. The deformed sampler $\mathbf{G_d}$ samples with different density at each location of the input image $\mathbf{I}$ according to the deformation map, where sampling locations are shown as red dots masked on the original image. The low-resolution downsampled image $\mathbf{J} = \mathbf{G_d}(\mathbf{I},\mathbf{d})$ is then fed into the segmentation module which makes a prediction $\mathbf{S}_\phi(\mathbf{J})$ in the low-resolution space. At inference time, the prediction is non-uniformly upsampled to the original space in a deterministic manner.

3.4 Single segmentation loss does not work: regularisation term is needed

Our method is similar to Recasens et al. (2018), which however is only explored for image classification tasks. Transferring the method proposed in Recasens et al. (2018) to segmentation not only needs reformulating the problem as described in section 3.1, optimising the joint system at the pixel level is not trivial due to potential oversampling at easy to segment regions like background. Such challenges have been verified in later experiments where the naive adaptation of the method from Recasens et al. (2018) to segmentation (red bars in Fig. 5) fails to improve both baselines on 2/3 datasets. To discourage the network learn trivial solutions, we add a regularisation term in addition to the segmentation loss. Inspired by the "edge-based" method (Marin et al., 2019) we add an additional edge-loss to encourage denser sampling around the edge. Different to the approach of Marin et al. (2019) 1) we directly simulate a target deformation map without separate training to carry its main message that edge information is useful and 2) fit into the joint training system excludes the heavy dependency to a specific parameter λ.

In specific, let Y_{lr} be the uniformly downsampled segmentation label, the edge loss $\mathcal{L}_e(I_{lr}, Y_{lr}; \theta) = MSE(d, d_t)$, where d is the predicted deformation map $d = D_\theta(I_{lr})$, and d_t is the target deformation map $d_t = Edge(Gaus(Y_{lr}, \delta))$[5]. To this end, we jointly optimise the parameters $\{\theta, \phi\}$ of both the *deformation module* and the *segmentation module* to minimise the segmentation specific loss function, $\mathcal{L}_s(\mathbf{J};\theta,\phi)$, and the edge-loss, $\mathcal{L}_e(I_{lr},Y_{lr};\theta)$. We also add a weight term to make sure both losses have comparable magnitudes.

4 Experiments and Results

In this section, we evaluate the performance of our deformed downsampling approach on three datasets from different domains as summarised in Table 1, against baselines described in section 4.1. We evaluate the segmentation performance with $\mathbf{IoU}$ and verify the quality of the sampler by looking at intrinsic interpolation error with $\mathbf{IoU}(\mathbf{Y},\mathbf{Y}')$. We show quantitatively and qualitatively (section 4.2) that our method can learn the spatial variation of optimal sampling density and lead to a better cost-performance trade-off. We further demonstrate how our method generalise the edge-loss differently than similar work (Marin et al., 2019) in section 4.3. The *deformation module* is defined as a

[5] $Edge()$ is an edge detection filter by a convolution of a specific 3×3 kernel [[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]] and $Gaus()$ is Gaussian blur with radius (Standard deviation of the Gaussian kernel) $\delta = 1$ to encourage sampling close to the edge.

small CNN architecture comprised of 3 convolution layers. The segmentation network was defined as a deep CNN architecture, with HRNetV2-W48 (Sun et al., 2019) used in all datasets, and PSP-net (Zhao et al., 2017) as a sanity check in the Cityscapes dataset. We employ random initialisation like as in He et al. (2015), the same training scheme (Adam (Kingma & Ba, 2014), the focal loss as segmentation loss and MSE loss for the edge-loss with full details in Appendix A.6) unless otherwise stated.

Table 1: Dataset summary. More details in Appendix A.5

Dataset	Content	Resolution (pixels)	Number of Classes
Cityscape (Cordts et al., 2016)	Urban scenes	2048×1024	19
DeepGlobe (Demir et al., 2018)	Aerial scenes	2448×2448	6
PCa-Histo (local)	Histopathological	$1968 \pm 216 \times 9392 \pm 4794$	6

4.1 BASELINES

We compare our method either with single segmentation loss ("Ours-Single loss") or adding edge-loss ("Ours-Joint loss") against two baselines: 1) uniform downsampling of the input to train the segmentation network followed by uniformly upsampling the prediction ("uniform"); 2) the second "edge-based" baseline is an approximation of method from Marin et al. (2019), follow same reason and scheme as declared in motivational section 3.2, which is the best-performed result from a set segmentation networks each trained with a simulated "edge-based" samplers represents with different λ. In addition, for a fair comparison, we also perform experiments at the same experimental conditions with square input and compare reported results from Marin et al. (2019) in section 4.3.

4.2 QUANTITATIVE AND QUALITATIVE PERFORMANCE ANALYSIS

We plot: 1) $\mathbf{mIoU}$ and $\mathbf{mIoU}(\mathbf{Y},\mathbf{Y}')$ in Fig. 5 comparing our method against both baselines on all three datasets; 2) Fig. 6 further investigate $\mathbf{IoU}$ at per-class level, aiming to understand the impact of different object sizes reflected by different class frequency; 3) lastly, in Fig. 7, we examine the impact of downsampled size and ask could our method learn a downsampling strategy enable better trade-off between performance and computational cost.

In Fig. 5, looking at $\mathbf{mIoU}$, we see "Ours-Joint loss" consistently performs best with a **boost of 3% to 10%** over the "uniform" baseline. We also observe the improvements after adding the *edge loss* vary over datasets which supported our argument that **our jointly trained method learnt to generalise sampling better at different important regions** and leads to better overall segmentation accuracy. On the other hand, $\mathbf{mIoU}(\mathbf{Y},\mathbf{Y}')$ does not always agree with $\mathbf{mIoU}$, indicting **segmentation error and interpolation error are not tightly bounded therefore separately trained sampler may fail and jointly trained system is necessary**.

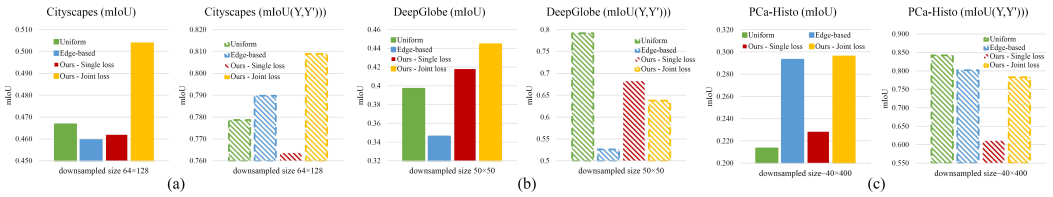


Figure 5: Quantitative comparison of segmentation performance with our learn to downsample strategy, either with single segmentation loss or adding the regularisation edge loss, versus two baseline downsampling methods, in terms of $\mathbf{mIoU}$ and $\mathbf{mIoU}(\mathbf{Y},\mathbf{Y}')$. Results are from three datasets: (a) Cityscapes (Cordts et al., 2016) (b) DeepGlobe (Demir et al., 2018) and a local prostate cancer histology dataset (PCa-Histo).

Class-wise "Ours-Joint loss" achieved **up to 25%** boost in absolute $\mathbf{IoU}$, as in Fig. 6, includes improving all low-frequency classes representing small objects, and significantly improved "edge-based" baseline in high-frequency classes representing large objects, either better or close to the "uniform" baseline. Such results verified **our joint trained method generalise "edge-based" sampling better to not only small but large objects**.

One key motivation of proposing our learn to downsample method is **enable a better trade-off between performance and computational cost**, and this is verified in Fig. 7. Our method shows can consistently save **up to 90%** computational cost than the "uniform" baseline, across a set of different

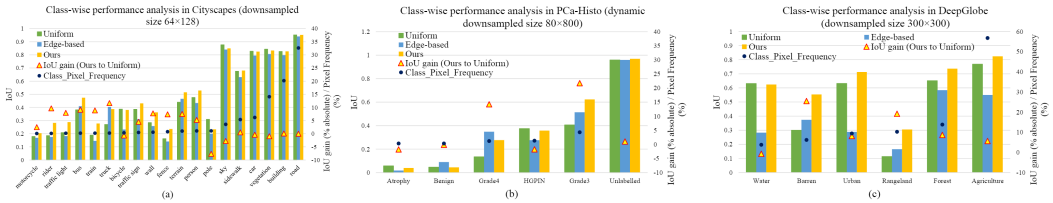


Figure 6: Class-wise IoU analysis on (a) Cityscapes (Cordts et al., 2016); (b) PCa-Histo; (c) DeepGlobe (Demir et al., 2018). Despite absolute $\mathbf{IoU}$, relative $\mathbf{mIoU}$ gain with our method over "uniform" baseline is shown. Classes are ordered with increasing pixel frequency which is also indicated in each plot.

downsampling sizes over all three datasets. We notice $\mathbf{mIoU}$ does not increase monotonically on DeepGlobe dataset (Demir et al., 2018) with increased cost in Fig. 7 (b), we stress this conveyed one important message that **high resolution is not always better but a data-dependent optimal cost-performance tradeoff exists and an end-to-end adjustable downsampler is needed**. It is also worth highlighting the PCa-Histo dataset at an extreme downsample rate (cost 0.0004), our method still performs significantly better. Looking at the interpolation error with $\mathbf{mIoU}(\mathbf{Y},\mathbf{Y}^{'})$, the results **shows our method can adjust optimisation priority depend on available computational resource**. For example in Fig. 7 (b) and (c) at the low-cost end (small downsampled size) our method learnt to compromise the interpolation error but focus on reducing segmentation error, while when budget increase (large downsampled size), the model quickly picks up $\mathbf{IoU}(\mathbf{Y},\mathbf{Y}^{'})$ while maintaining segmentation $\mathbf{mIoU}$. Putting quantitative evidence together, we showed **our method can efficiently learn where to "invest" the limited budget of pixels at downsampling to achieve the highest overall return in segmentation accuracy**.

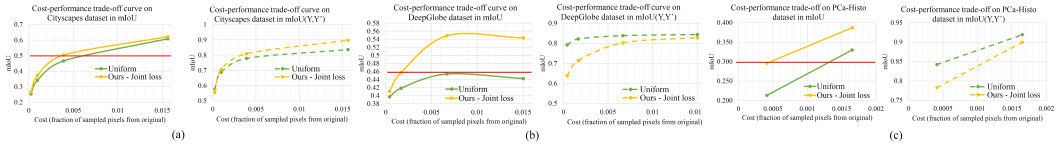


Figure 7: Cost-performance trade-offs in terms of $\mathbf{mIoU}$ and $\mathbf{mIoU}(\mathbf{Y},\mathbf{Y}^{'})$ on (a) Cityscapes dataset (Cordts et al., 2016); (b) DeepGlobe dataset (Demir et al., 2018) and (c) PCa-Histo. The red lines indicates at same performance our method can save **33%**, **90%** and **70%** from the "uniform" baseline on the three datasets respectively. Cost determines the amount of computation required at segmentation.

The visual overview (Fig. 1) of applying our method **confirmed that our method adapts the sampling density over different locations, which in turn leads to better segmentation accuracy**. Comparing with both baselines, visual results from Fig. 8 and Fig. 9 show our method **picked the advantage of the "edge-based" approach by sampling denser at small objects while further generalise with better sparse sampling on larger objects** (see caption for details). We also find visual evidence supporting the claim **object edges does not guarantee most informative locations** (see window edges in the wall class of Fig. 8 misguided the edge-based sampler under-sampling of surrounding textures and lead to miss-prediction, and similarly as bushes in forest region as Fig. 9). We therefore further discuss how boundary accuracy affects segmentation accuracy in the next section.

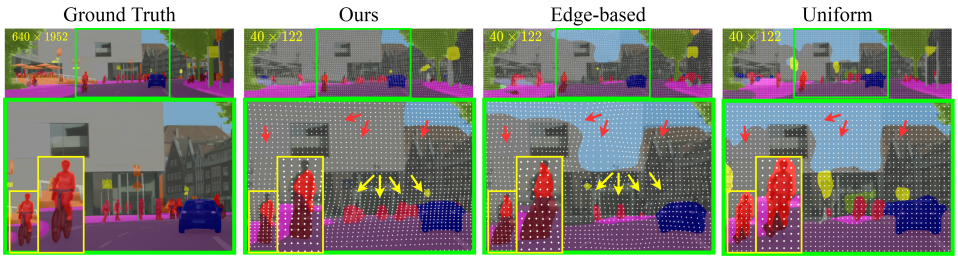


Figure 8: Examples on Cityscapes (Cordts et al., 2016) comparing our method against both baselines, where segmentation is performed on 16 times downsampled images (at each dimension). Predictions are masked over and sampling locations are shown in white dots. Yellow/ red arrows indicated regions denser/ sparser sampling helped to segment rider (red)/ sky (blue) classes, respectively.

4.3 A FAIR COMPARISON TO THE "EDGE-BASED" AND BOUNDARY ACCURACY ANALYSIS

Despite comparison with simulated "edge-based" baseline at non-square input, in this section we perform experiments at same experimental condition on Cityscapes dataset and compare to reported

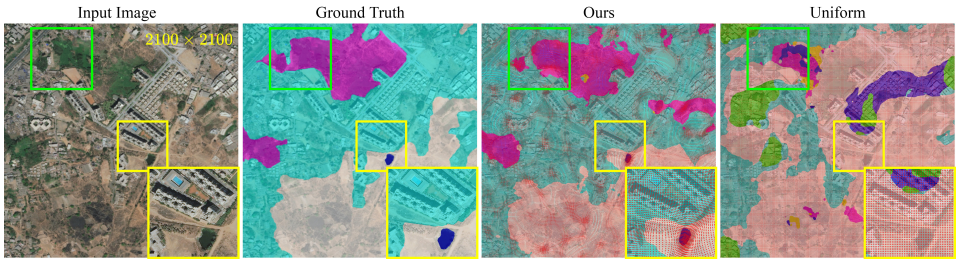
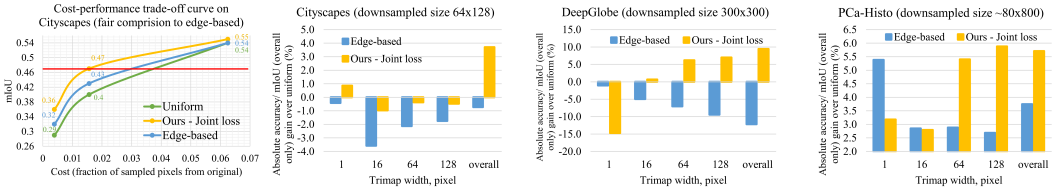


Figure 9: Qualitative example on DeepGlobe dataset (Demir et al., 2018) where segmentation performed on 8 times downsampled image (at each dimension). Predictions are masked over and sampling locations are shown in red dots. Yellow/ green boxed region indicated regions denser/ sparser sampling helped segmenting water (blue)/ forest (purple) classes, respectively.

results from Marin et al. (2019). As shown in Fig. 10 (a), our method outperforms the original "edge-based" (Marin et al., 2019) across all trade-off downsample sizes by **upto 4% in absolute mIoU** and achieves better cost-performance trade-off **saving upto 47% calculation**.

We introduced the novel edge-loss to regularize the training of the joint system, inspired by Marin et al. (2019). Here we measure boundary precision and ask 1) how does it contribute to overall segmentation accuracy? and 2) how differently do the joint system balance boundary accuracy and overall segmentation accuracy than the separate system (Marin et al., 2019)? We adopt trimap following Kohli et al. (2009); Marin et al. (2019) computing the accuracy within a band of varying width around boundaries in addition to **mIoU**, for all three datasets in Fig. 10 (b). We found: 1) the **"edge-based"** baseline consistently obtain the **best accuracy closest to the boundary, but this does not consistently transfer to mIoU gain**; 2) **Our joint learnt downsampling** on the other hand **automatically identify the most beneficial sampling distance** to invest sampling thus **lead to the best overall mIoU improvement**; 3) The learnt **most beneficial sampling distance by our approach also turns out to be data-dependent** that can close to the boundary (i.e. Cityscapes) or away (i.e. DeepGlobe and PCa-Histo), while edge approach without this adjustable scheme will fail; 4) it also strengthened one key hypothesis that **our method is particularly useful when the labelling region is based on higher-level knowledge**, i.e. Deepglobe dataset and PCa-Histo dataset when optimal edge failed, which scenario is of the future research interests in the vision community.



(a) (b)

Figure 10: (a) Experiments performed at comparable condition to reported "edge-based" results by Marin et al. (2019) on Cityscapes dataset (Cordts et al., 2016), i.e. applying central 1024×1024 crop pre-processing and train with PSP-net (Zhao et al., 2017). The red line indicates to achieve same **mIoU**, our method saves **47% and 58%** calculation than the "edge-based" (Marin et al., 2019) and the "uniform" baselines respectively; (b) Absolute accuracy difference between our method and "edge-based" baseline near semantic boundaries and overall **mIoU** differences (shown as overall) on the three datasets.

5 Conclusion

We introduce an approach for learning to downsample ultra high-resolution images for segmentation tasks. The main motivation is to adapt the sampling budget to the difficulty of segmented pixels/regions. We empirically illustrate SOTA method (Marin et al., 2019) been limited by sampling location fixed constrained to manual designed edge objective, and hence motivate our end-to-end trained method. We illustrate simple extend the method of (Recasens et al., 2018) to segmentation does not work, and propose to avoid trivial solutions by incorporating an edge-loss to regularize the training. Although our edge-loss, despite a simpler approximation, share the same spirit with Marin et al. (2019) we demonstrate our jointly trained method generalises sampling more robustly especially when label edges are less informative and consistently leads to a better cost-performance trade-off. Our method is light weighted and can be flexibly combined with the existing segmentation method without modifying architecture.

REFERENCES

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3640–3649, 2016.

Wuyang Chen, Ziyu Jiang, Zhangyang Wang, Kexin Cui, and Xiaoning Qian. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8924–8933, 2019.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

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*, pp. 764–773, 2017.

Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9785–9795, 2019.

Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 447–456, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

Chen Jin, Ryutaro Tanno, Moucheng Xu, Thomy Mertzanidou, and Daniel C Alexander. Foveation for segmentation of mega-pixel histology images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 561–571. Springer, 2020.

Angelos Katharopoulos and François Fleuret. Processing megapixel images with deep attention-sampling models. *arXiv preprint arXiv:1905.03711*, 2019.

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

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

Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 3226–3229. IEEE, 2017.

Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, Fei Yang, and Yuri Boykov. Efficient segmentation: Learning downsampling near semantic boundaries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2131–2141, 2019.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

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)*, pp. 51–66, 2018.

Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep neural network models for computational histopathology: A survey. *arXiv preprint arXiv:1912.12378*, 2019.

Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2088–2096, 2017.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 52–67, 2018.

Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, Liuyu Xiang, Zerun Wang, Guiguang Ding, David Brady, Qionghai Dai, et al. Panda: A gigapixel-level human-centric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3268–3278, 2020.

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

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

A APPENDIX

Here we provide additional results and various ablation studies and implementation details that have not been presented in the main paper. We submit our code in an anonymous repository at the discussion.

CONTENTS

A.1 More visual examples at extreme downsampling rates 12

A.2 Investigate the learnt class distribution 13

A.3 Stage training: how much contribution from the deformation module and the edge loss 14

A.4 Simulated "edge-based" results 15

A.5 Datasets 15

A.6 Network Architectures and Implementation Details 16

A.1 MORE VISUAL EXAMPLES AT EXTREME DOWNSAMPLING RATES

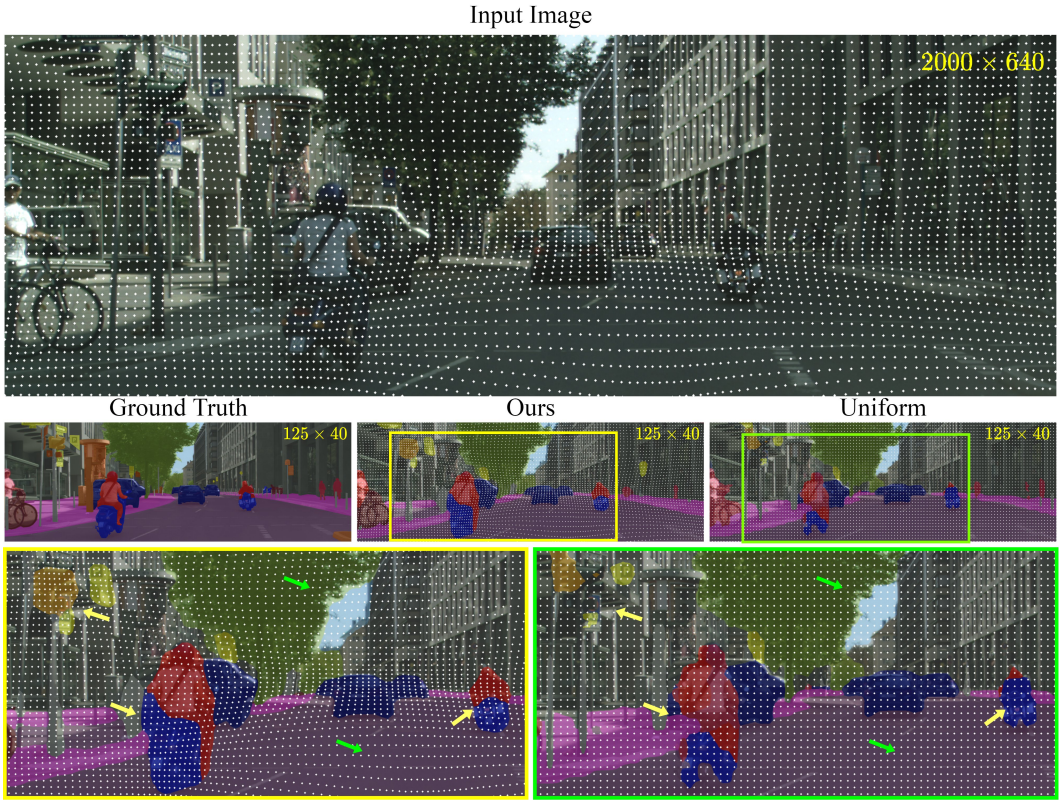


Figure 11: **Semantic segmentation with learnt deformed downsampling on Cityscapes.** Our method learns non-uniform downsampling (white dots on the images) a high-resolution image (top-row) to preserve information that guides segmentation and ignore image content that does not contribute (green arrows) thereby enabling low-cost semantic segmentation on low-resolution downsampled images. This strategy helps identify small regions such as the people and traffic signs in the figure; see masked ground truth and predictions in middle-row. Compared to "Uniform" baseline "Ours" samples are more densely at important semantic regions (yellow arrows) and leads to more accurate segmentation (see bottom-row).

A visual overview of applying our method on Cityscapes are given in Fig. 11. To further verify how our method performs at extreme downsampling on histology images, in Fig. 12, we show visual examples when a downsampling rate of 50 times at each dimension is applied to the PCa-Histo dataset. Consistent with quantitative results, **our method can still perform well comparing to ground truth**

and significantly better than "uniform" baseline at such extreme downsampling rate, which leads to better cost-performance trade-off. Our method performs **especially well for the most clinically important classes, Gleason Grade 3 and Gleason Grade 4**, the separation of these two classes is often referred to as the threshold from healthy to cancer. The sampling locations shown as blue dots in Fig. 12 also indicated our method learnt to sample densely at clinical important and challenge regions (e.g. cancerous nuclei and tissues) while sampling sparsely at less informative regions (e.g. unlabelled/background), which lead to consistent improvement over uniform sampling on segmentation accuracy.

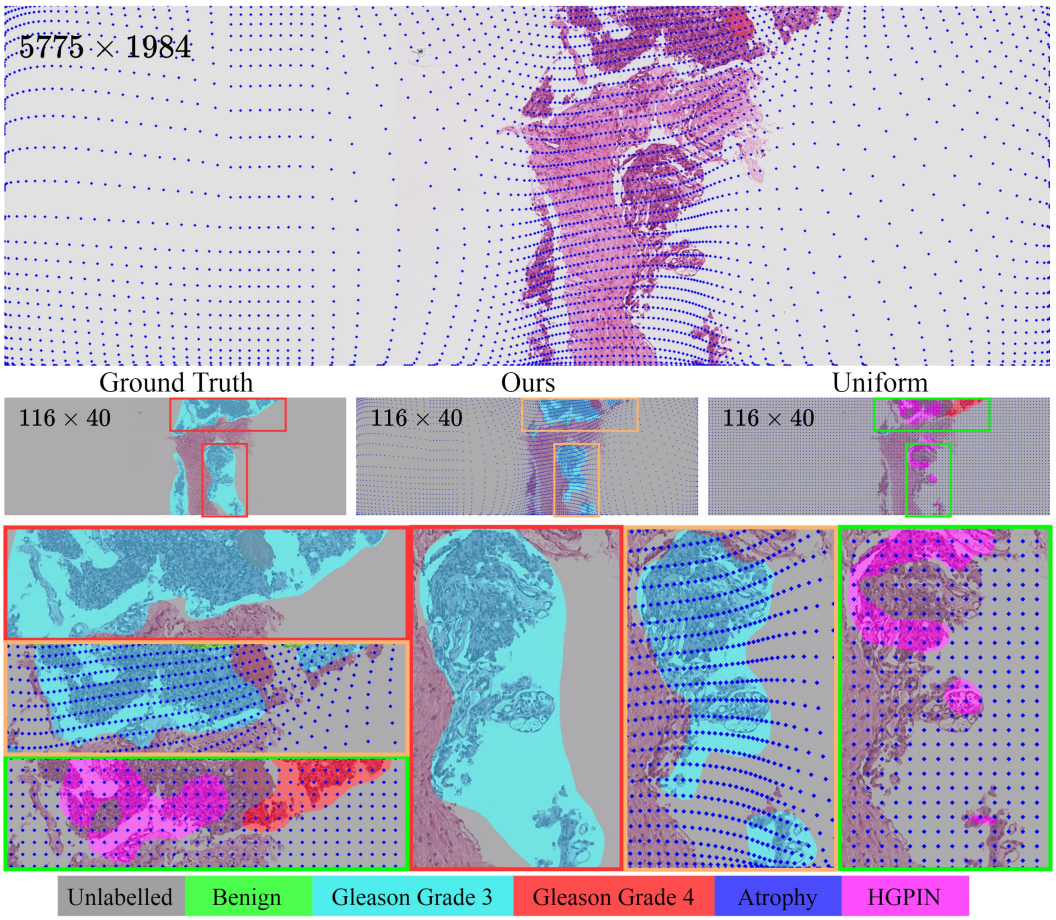


Figure 12: Qualitative example on PCa-Histo dataset (1) where segmentation performed on 50 times (at each dimension) downsampled image. Predictions are masked over and sampling locations are shown in blue dots.

A.2 Investigate the learnt class distribution

To verify our *deformation module* has learnt to downsample denser at important regions/classes effectively, **we monitor class frequency change after downsampling**, as shown in Fig. 13. To have a direct evaluation of whether a class has been sampled denser or sparser comparing to its original frequency, we calculate the class frequency change as the ratio between sampled and original image.

The results indicated **the *deformation module* learnt to adjust sampling strategy so that more pixels are "invested" to important regions/classes and less from less informative region/classes, and lead to optimal overall segmentation accuracy**. For example, the *deformation module* learnt to sample less (shown as negative sampling frequency ratio) in the "Background" class in the PCa-Histo dataset (Fig. 13 (a)) and "road" and "sky" classes in the cityscapes (Cordts et al., 2016) dataset (Fig. 13 (b)), in both cases those classes can be considered as potential "background" with common knowledge. However, **our method also shows ability to adjust sampling strategy on more complex dataset**

where "background class" is less obvious. For example, in DeepGlobe (Demir et al., 2018) dataset there is no obvious "background class", but the "Urban" class has been learnt to sample much denser than other classes (Fig. 13 (c)), and leads to boosted segmentation accuracy (see Fig. 5 and Fig. 7 in the main paper). The learnt distribution on the DeepGlobe dataset also **verified our argument that the manual designed *sample denser around object edge strategy* as the "edge-based" (Marin et al., 2019) method may fail when object boundary is less informative**, which is justified in the main results.

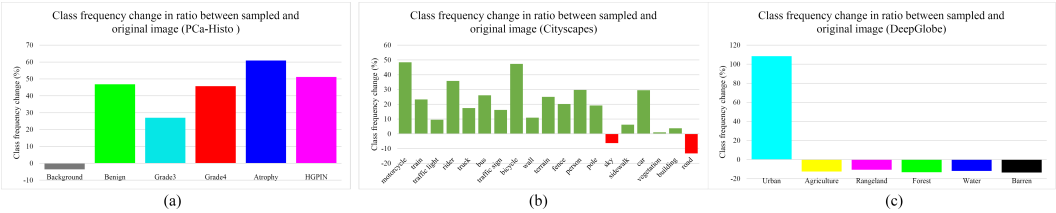


Figure 13: Class frequency change in the ratio between sampled and original image. The higher the value the denser the learnt sampling. Our method shows able to learn to adjust sampling either when foreground and background classes are clear (PCa-Histo and Cityscapes (Cordts et al., 2016)) and when less clear (DeepGlobe (Demir et al., 2018)). Results are from three datasets: (a) a local prostate cancer histology dataset (PCa-Histo) at dynamic downsampled size of 80×800 pixels (b) Cityscapes (Cordts et al., 2016) at downsampled size of 64×128 pixels and (c) DeepGlobe (Demir et al., 2018) at downsampled size of 300×300 pixels.

A.3 STAGE TRAINING: HOW MUCH CONTRIBUTION FROM THE DEFORMATION MODULE AND THE EDGE LOSS

To better understand the compounded **contribution from the *segmentation loss* and the *edge loss***, we break the training into two stages, and at each stage we optionally switch on/off either the loss terms to separately investigate their contribution, on the Cityscapes (Cordts et al., 2016) dataset. First, we separately train each module with a single loss, referred to as "stage - single loss" in Fig. 14. Specifically, we singularly train the downsampler, the *deformation module*, while freezing the *segmentation module* in the first stage with single *edge loss*, and in the second stage, we freeze the *deformation module* and only train the *segmentation module* with single *segmentation loss*. The **"stage - single loss" strategy** is a two-stage training strategy that is most similar to the original "edge-based" (Marin et al., 2019) and its results **represents the contribution of the "edge loss" term in separate training setting**. On top of "stage - single loss", we modify the second stage by jointly train both module with both the *segmentation loss* and the *edge loss*, referred to as "stage - joint loss". The **"stage - joint loss" result represents the contribution from the joint system**. The results in Fig. 14 shows that **contribution from "edge loss" in a separately trained setting is limited**, where "stage - single loss" under-performed than "uniform" by 6% in **mIoU**. While **contribution from our joint trained system is significant**, where "stage - joint loss" significantly improved "stage - single loss", and performs close to "uniform" and slightly better than "ours - single loss" and "edge-based".

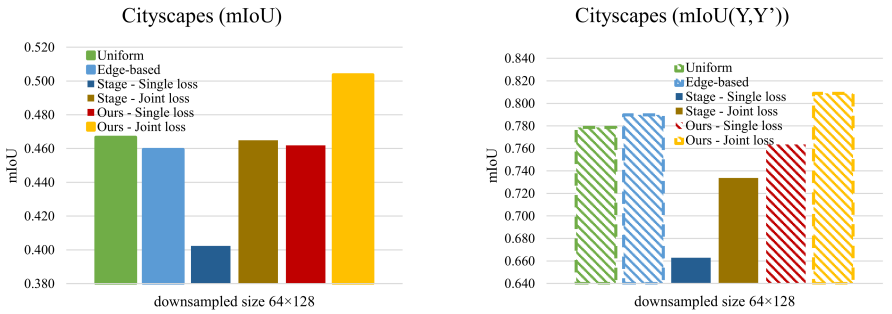


Figure 14: Comparison between two-stage training versus joint training with our approach. A key insight is the "edge loss" in a separate trained setting ("stage - single loss") does not work, while our joint trained system ("stage - joint loss") can significantly improve performance. Results for "uniform" and "edge-based" baselines are included as references. Results are from Cityscapes datasets at the downsampled size of 64×128.

A.4 SIMULATED "EDGE-BASED" RESULTS

We direct simulate different sampling density around edges by expanding sampling locations away from label edge at different distances in a Gaussian manner to mimic the effect of λ on sampling as shown by Marin et al. (2019). In specific, we calculate edge from label then apply gaussian blur to generate a simulated deformation map to guide sampling in our framework. As declared in the main text, our "edge-based" baseline is the best-performed segmentation network trained from a set of simulated "edge-based" downsamplers, each with a fixed sampling density around the edges. Here we present full segmentation accuracy with the set directly simulated "edge-based" downsamplers, for each of the three datasets in Fig. 15.

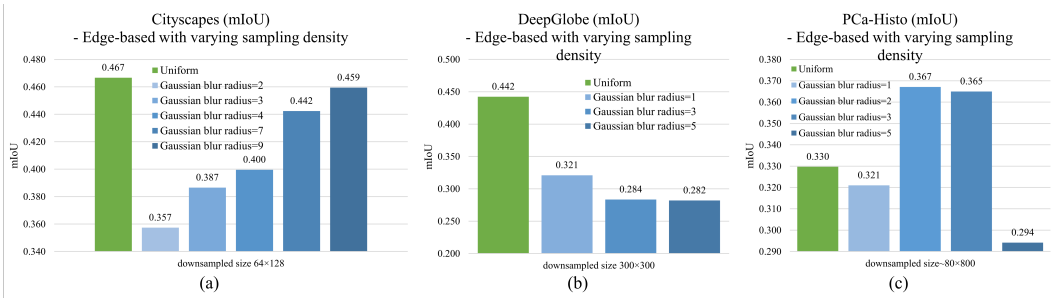


Figure 15: Simulated "edge-based" results, with a set of directly simulated "edge-based" downsamplers, each with a different fixed sampling density determined by Gaussian blur radius. The best-performed result is selected and referred to as the "edge-based" result presented in the main paper. Results are from three datasets: (a) Cityscapes (Cordts et al., 2016) (b) DeepGlobe (Demir et al., 2018) and (c) a local prostate cancer histology dataset (PCa-Histo).

A.5 DATASETS

In this work, we verified our method on three segmentation datasets: DeepGlobe aerial scenes segmentation dataset (Demir et al., 2018), Cityscapes urban scenes segmentation dataset (Cordts et al., 2016) and PCa-Histo medical histopathological segmentation dataset.

The **DeepGlobe (Demir et al., 2018)** dataset has 803 high-resolution (2448×2448 pixels) images of aerial scenes. There are 7 classes of dense annotations, 6 classes among them are used for training and evaluation according to Demir et al. (2018). We randomly split the dataset into train, validate and test with 455, 207, and 142 images respectively.

The **Cityscapes (Cordts et al., 2016)** dataset contains 5000 high-resolution (2048×1024 pixels) urban scenes images collected across 27 European Cities. The finely-annotated images contain 30 classes, and 19 classes among them are used for training and evaluation according to Cordts et al. (2016). The 5000 images from the Cityscapes are divided into 2975/500/1525 images for training, validation and testing.

The **PCa-Histo:** dataset contains 266 ultra-high resolution whole slide images, from 33 distinct biopsies. The size of the images ranged from 250800 pixels to 40880000 pixels. Each pixel is annotated into one out of six classes: Unlabelled, Benign, Gleason Grade 3, Gleason Grade 4, Atrophy, HGPIN (see Table 2 for explanations of each class). Among the six classes, informative classes are underrepresented comparing to Unlabelled class (see Table 2). We random split the dataset into 200 training, 27 validation and 39 test images. This dataset has a varying size, with height (the short axis) range between 229 to 2000 pixels (1967.5 ± 215.5 pixels), width (the long axis) range between 3386 to 20440 pixels (9391.9 ± 4793.8 pixels) and Aspect Ratio (AP) range between 1.69 to 33.9 (5.04 ± 3.54). The downsampling size of this dataset is calculated as follows: with a dynamic downsample size of for example 80×800, for each image, we calculate two candidates downsample sizes by either downsample the short axis to 80 or the long axis to 800 while in either case we keep AP unchanged. Then we select the one to downsample size that has fewer total pixels. For example, for an image with a size 368×7986 pixels, we have candidate downsample sizes of 200×4340 pixels or 36×800 pixels, and we will use downsample size of 36×800 in our experiment. Also we apply batch size per GPU node of 1 because of the varying downsampled size of each image.

The prostate biopsies of the PCa-Histo dataset are collected of 122 patients. The identity of the patients is completely removed, and the dataset is used for research. The biopsy cores were sliced, mounted

Table 2: Pixel frequency distribution of PCa-Histo dataset and explanation of each class

Label	Background	Benign	Gleason Grade 3	Gleason Grade 4	Atrophy	HGPIN
% pixels in dataset	93.99	0.97	1.23	3.35	0.26	0.20
Explanation	all unlabelled pixels	healthy	mild cancer	aggressive cancer	healthy noise	precursor legion

on glass slides, and stained with H&E. From these, one slice of each of 220 cores was digitised using a Leica SCN400 slide scanner. The biopsies were digitised at 5, 10 and 20 magnification. At 20 magnification (which is the magnification used by histopathologists to perform Gleason grading), the pixel size is 0.55 μm^2.

A.6 Network Architectures and Implementation Details

Architectures: The *deformation module* is defined as a small CNN architecture comprised 3 layers each with 3×3 kernels follower by BatchNorm and Relu. The number of kernels in each respective layer is $\{24,24,3,1\}$. A final 1×1 convolutional layer to reduce the dimensionality and a softmax layer for normalisation are added at the end. All convolution layers are initialised following He initialization (He et al., 2015). The *segmentation module* was defined as a deep CNN architecture, with HRNetV2-W48 (Sun et al., 2019) applied in all datasets, and PSP-net (Zhao et al., 2017) as a sanity check in the Cityscapes dataset. The segmentation network HRNetV2-W48 is pre-trained on the Imagenet dataset as provided by Sun et al. (2019). In all settings, the size of the deformation map $\mathbf{d}$ is either consistent or twice the size of the downsampled image.

Training: For all experiments, we employ the same training scheme unless otherwise stated. We optimize parameters using Adam (Kingma & Ba, 2014) with an initial learning rate of 1×10^{-3} and $\beta = 0.9$, and train for 200 epochs on DeepGlobe, 250 epoches on PCa-Histo dataset, and 125 epochs on Cityscapes dataset. We use a batch size of 4 for the Cityscapes dataset (Cordts et al., 2016) and the DeepGlobe dataset (Demir et al., 2018) and batch size of 2 for the PCa-Histo dataset. We employ a step decay learning rate policy. We scale the *edge loss* 100 times so it is in the same scale with the *segmentation loss*. During the training of the segmentation network we do not include upsampling stage but instead, downsample the label map. For all datasets, we take the whole image to downsample as input, without any crop. During training, we augment data by random left-right flipping. All networks are trained on 2 GPUs from an internal cluster (machine models: gtx1080ti, titanxp, titanx, rtx2080ti, p100, v100, rtx6000), with syncBN.