Supplementary Material: SideInfNet: A Deep Neural Network for Semi-Automatic Semantic Segmentation with Side Information

Jing Yu Koh[1⋆], Duc Thanh Nguyen[2], Quang-Trung Truong[1], Sai-Kit Yeung[3], and Alexander Binder[1]

[1] Singapore University of Technology and Design, Singapore
[2] Deakin University, Australia
[3] Hong Kong University of Science and Technology, Hong Kong

Abstract. In this supplementary material, we provide implementation details of our SideInfNet in Section 1. We present ablation experiments conducted to ascertain the effectiveness of our method in Section 2, in which we compare against existing fusion methods by implementing our model with the same baseline segmentation network. Computational analysis of our method is performed in Section 3. We present additional qualitative evaluations of our method and prior works in three case studies in Section 4.

1 Implementation Details

In this section, we detail the settings used to train our proposed SideInfNet in various case studies. All models of our SideInfNet were implemented in PyTorch v1.2 [6].

1.1 General Settings

The domain-dependent feature extractor of our proposed SideInfNet is built based on the Deeplab-ResNet [2] model. In our implementation, we optimized the Deeplab-ResNet using stochastic gradient descent (SGD) with a momentum of 0.9. The Deeplab-ResNet receives input an image of size $H \times W$ pixels and produces a *conv2_3* layer output of approximately size $\frac{H}{4} \times \frac{W}{4}$. Therefore, our maxpool layer uses a kernel size of 6 and a stride of 4 to achieve the desired size.

For processing side information, we used a single fully-connected layer that mapped input vectors to a 64-dimensional space. As shown in Section 2, this setting (i.e., 64 dimensions for side information) balanced both the accuracy and computational cost and worked well in all case studies. We also experimented with deeper multi-layer perceptrons (MLP) and non-linear activation functions, but found no improvement from these settings.

⋆ Currently an AI Resident at Google.

For all tasks, transfer learning was applied. We initialized our SideInfNet models with the weights of the Deeplab-ResNet trained for semantic segmentation on the Microsoft COCO (MS-COCO) dataset [5]. Due to concatenation of data in our models, the weights of the layers at the concatenation point (i.e., the *conv2_3* layer in the Deeplab-ResNet architecture) cannot be directly restored. Instead, we randomly initialized additional channels that are required for the concatenation. Specifically, we restored the first 256 channels of the *conv2_3* layer from the MS-COCO pretrained weights, and randomly initialized the additional 64 channels.

1.2 Zone Segmentation

Training For training, we used 80×80 pixels crops of each city from the zone segmentation dataset [3]. Patches containing more than 60% of masked data were discarded. Each patch was saved along with its coordinate information for retrieval of the geotagged photo data. We normalized images by performing mean subtraction from the RGB channels using the training set mean.

To augment the training data, we performed random horizontal and vertical flips of image patches. We also experimented with scaling the patches, but did not observe any improvement in performance.

Training was performed with a mini-batch size of 16 and over 20 epochs. We used a base learning rate of 0.00025 with a polynomial learning rate decay with power of 0.9. In addition, we set the learning rate for the MLP to 0.025, and for the data fusion *conv2_3* to 0.0005 respectively. The reason for this setting is that these layers are not restored through transfer learning, and benefit from higher learning rates. To make the training stable, we used a learning rate warmup of 20 data epochs, in which the learning rate linearly increased from epoch 1 to epoch 20.

In our models, each fractionally-strided convolution was multiplied by a learnable scalar. We initialized all the scalars to 1, which we found to be helpful in diffusing geotagged photo data. Intuitively, this initialization could result in maximum diffusion by default, which we found essential to aid in learning meaningful representations for sparse side information.

Evaluation We tested our model using 3-fold cross validation, in which two cities were used for training, and the other city was used for testing. In each validation, we scanned the test satellite image by a window of size 80×80 pixels and a spatial stride of 21×21 pixels for NYC and BOS, and 23×23 pixels for SFO (due to the different scales of the input data).

Inference was performed individually on the windows to retrieve the softmax class probabilities. The resulting softmax patches were then merged, and overlapping regions were averaged. The final inference result was achieved by taking an argmax over the averaged softmax result.

1.3 BreAst Cancer Histology Segmentation

Training Due to the large size of whole-slide images in the BreAst Cancer Histology (BACH) dataset [1], we downscaled the whole-slide images for computational efficiency. We first resized the whole-slide images to $\frac{1}{4}$ of their original size. We then cropped patches of 299×299 pixels with a stride length of 99×99 pixels. We discarded all patches that contained less than 5% of non-normal classes. Each patch was saved along with its coordinate information for retrieval of the brush stroke annotations. Lastly, we normalized images through mean subtraction where the mean was derived from training dataset. We also performed random horizontal and vertical flips for data augmentation.

Training was performed using a mini-batch size of 4 with gradients accumulated over 4 iterations. We used a base learning rate of 0.0001 with a polynomial learning rate decay with power of 0.9. In addition, we set the learning rate for the MLP to 0.01, and of the fusion layer *conv2_3* to 0.0002 respectively. The learning rate for the classification layer was set to 0.001.

The model was trained for 20 epochs, with early stopping imposed if accuracy on the validation set did not increase for 3 epochs. We used a learning rate warmup of 20 data epochs for stability in training; the learning rate linearly increased from epoch 1 to epoch 20.

The learnable scalar for the first fractionally-strided convolution was set to 1, and all others were set to 0, resulting in no diffusion by default. We found this essential to aid in learning good representations for dense side information, such as brush stroke annotations.

Evaluation We performed inference using patches processed as in the training procedure. We averaged the softmax probabilities of any overlapping regions. Similarly to the zone segmentation case study, the final results were achieved by taking an argmax over the averaged softmax result.

1.4 Urban Segmentation

For urban segmentation on the Zurich Summer Dataset [7], we cropped training images to patches of size 80×80 pixels with a stride length of 20×20 pixels. Images were saved with associated coordinate information for retrieval of brush stroke annotations. We normalized images by performing mean subtraction from the RGB channels using the training set mean. Data augmentation was also done using random horizontal and vertical flips.

Hyperparameter setting was similar to that of the zone segmentation case study. We initialized the learnable scalar for the first fractionally-strided convolution to 1, and all others to 0, similarly to the BACH dataset.

In our experiments, we used the images *zh5*, *zh7*, *zh8*, *zh11*, and *zh18* for testing. All other images were used for training. This split ensures that all classes are present in both training and testing.

Table 1. Performance of variants of SideInfNet in zone segmentation [3]. Best performances are highlighted.

Approach	Street Photo	Fractionally Strided Convolutions	Gate Rate (t)	Pixel Accuracy			
				BOS	NYC	SFO	Average
Deeplab-ResNet [2]	-	-	-	60.79%	59.58%	72.21%	64.19%
Geotagged	✓	-	-	60.19%	58.87%	74.18%	64.41%
Diffused	✓	✓	-	69.08%	71.95%	79.49%	73.51%
SideInfNet	✓	✓	0.8	70.10%	70.67%	79.38%	73.38%
SideInfNet	✓	✓	0.6	71.33%	71.08%	79.59%	74.00%
SideInfNet	✓	✓	0.4	70.45%	70.58%	79.51%	73.51%

2 Ablation Studies

2.1 Components of SideInfNet

In order to validate the benefits of our various technical novelties, we performed several ablation experiments on the main components of our proposed SideInfNet. In this supplementary material we present experimental results from the zone segmentation application, although similar trends are observed from the other case studies as well.

The results are summarized in Table 1. It is shown that the inclusion of side information in the form of street-level photos is essential in improving the segmentation accuracy. In particular, our best performing model (SideInfNet), fusing both domain-dependent features from satellite data and side information, achieved a relative gain of 15.28% over the baseline Deeplab-ResNet [2] that uses only satellite imagery. In addition, the results prove that side information diffusion using fractionally-strided convolutions (*Diffused* model) was important for performance improvements. This method of diffusion gained a relative improvement of 14.89% over the *Geotagged* model, which simply diffused the side information upon spatial distance (via nearest neighbor interpolation).

The SideInfNet model with adaptive inference gates also slightly improved over the *Diffused* model. An additional benefit of the adaptive inference gates is reduced computational complexity and model parameters, as not all the layers in the network architecture are executed for each run.

2.2 Varying Feature Dimension of Side Information

As presented in the implementation details of SideInfNet in Section 1, the side information is fed through a single fully-connected layer to produce a 64-dimensional feature vector. We experimented our SideInfNet with various out sizes of the fully-connected layer including 64, 128, 256, and 512, and report the results on the zoning [3], BACH [1], and Zurich Summer dataset [7] in Table 2.

Experimental results show that increasing the feature dimensionality of side information (i.e., the output size of the fully-connected layer) on the zoning dataset has a negligible effect on the performance, e.g., ¡2% of deviation in mIOU,

Table 2. Performance of SideInfNet when varying the dimension of side information. Note that "-" in the BACH dataset indicates that the model is unable to learn. Best performances are highlighted.

(a) Zoning [3]

Dimension	mIOU				Pixel Accuracy			
	BOS	**NYC**	**SFO**	**Average**	**BOS**	**NYC**	**SFO**	**Average**
64	41.96%	39.59%	60.31%	47.29%	71.33%	71.08%	79.59%	74.00%
128	40.63%	40.71%	44.98%	42.11%	70.79%	72.00%	72.32%	71.70%
256	39.52%	39.10%	57.67%	45.43%	70.29%	70.10%	78.31%	72.90%
512	38.78%	40.10%	57.00%	45.30%	69.18%	71.15%	77.07%	72.74%

(b) BACH [1]

Dimension	mIOU			Pixel Accuracy		
	A05	**A10**	**Average**	**A05**	**A10**	**Average**
64	59.03%	35.45%	47.24%	89.68%	54.29%	71.99%
128/256/512	-	-	-	-	-	-

(c) Zurich [7]

Dimension	mIOU	Pixel Accuracy
64	58.31%	78.97%
128	51.69%	74.71%
256	45.89%	73.09%
512	41.37%	69.94%

as the high dimensional side information vectors can be mapped meaningfully. In contrast, on the BACH and Zurich Summer datasets, worse performance is observed when increasing the feature dimensionality of side information. This is likely due to the simplicity of the side information in these datasets, e.g., brush strokes can be represented simply by scalars corresponding to different semantic classes. We also observe that, on the BACH dataset, when the size of the side information exceeds 64, SideInfNet is unable to learn any meaningful features, leading to either random or biased predictions. On the Zurich Summer dataset, side information of brush strokes has dimensionality of 8 and increasing the side information's dimensionality leads to overfitting. Therefore, to make a balance between the performance and computational complexity, we recommend 64-dimensional side information vectors for all the case studies and datasets.

2.3 Varying Levels of Side Information

In our main paper, we present an experiment on varying the availability of side information (see Section 4.4 in the main paper). In this experiment, we used all the side information available in training datasets to train the SideInfNet model

and tested the model by varying the level of side information in test sets. Recall that, to simulate various levels of side information while keeping the same spatial distribution, we sample the side information, e.g., brush strokes, using k-means algorithm applied on the centers of the brush strokes.

In this supplementary material, we provide more detailed results and in-depth analysis on the results. Specifically, we varied the availability of side information in both the training and test sets, e.g., $x\%$ of available side information is used in training vs $y\%$ of available side information is used in testing, where x and y vary in 20%, 40%, ..., 100%. We show the detailed performance of SideInfNet (in both mIOU and pixel accuracy) on the zoning [3], BACH [1], and Zurich Summer dataset [7] in Fig. 1, Fig. 2, and Fig. 3 respectively. From experimental results, we observe that, to achieve the best overall performance, SideInfNet should be trained with 100% side information available in the training data but can work well at inference time even with fewer side information. This confirms the practicality and applicability of our model in situations where a few annotations from users can significantly improve the segmentation quality.

We present several qualitative results of varying the availability of side information on various datasets in Fig. 4, Fig. 5, Fig. 6, and Fig. 7. We observe noticeable improvement of segmentation quality when side information is used. For instance, on the zoning dataset shown in Fig. 4, many regions cannot be identified from satellite imagery. Without using geotagged photos, the baseline Deeplab-ResNet misclassifies the majority of *commercial* regions as *industrial* in SFO. As the amount of side information available increases, the segmentation quality is steadily improved. Similar trends are also found in NYC and BOS.

On the BACH dataset (see Fig. 5), an increased number of brush strokes help to overcome under-segmentation in contiguous regions. Rarer classes such as *benign* in A05 slide and *in situ carcinoma* in A10 slide are more consistently identified with the inclusion of brush strokes.

On the Zurich Summer dataset, as illustrated in Fig. 6 and Fig. 7), the improvement is not as visually obvious as compared with the zoning dataset. This is likely due to the availability of high resolution imagery in the Zurich Summer dataset, which allows the model to make better baseline predictions without side information. However, the inclusion of side information via brush strokes also helps to correct errors made from the initial segmentation. For instance, in *zh5* (see Fig. 6), side information helps to correctly identify the tiny *Bare Soil* area. Similarly, in *zh8* (see Fig. 6), our method is able to segment the *Railway* class more accurately when provided with side information. We note that these classes are less presented in the dataset, which benefit the most when side information is included.

2.4 SideInfNet with another CNN Backbone

In our main paper, we experimented SideInfNet with VGG, the backbone used in the Unified model [8]. In this section, we show in detail the performance of SideInfNet with VGG backbone on all the datasets. In addition, to prove the robustness of our proposed method of multi-modal data fusion over the

Table 3. Comparison of SideInfNet and Unified model [8] on Deeplab-ResNet and VGG backbone.

(a) Zoning [3]

	mIOU				Pixel Accuracy			
	BOS	**NYC**	**SFO**	**Average**	**BOS**	**NYC**	**SFO**	**Average**
SideInfNet/DRN*	41.96%	39.59%	60.31%	47.29%	71.33%	71.08%	79.59%	74.00%
SideInfNet/VGG	41.94%	39.68%	56.73%	46.12%	68.28%	68.06%	75.95%	70.06%
Unified [8]/DRN*	37.61%	36.71%	57.31%	47.46%	66.87%	68.77%	77.96%	72.42%
Unified [8]/VGG	40.51%	39.27%	55.36%	45.05%	67.91%	70.92%	75.92%	71.58%

* DRN: Deeplab-ResNet

(b) BACH [1]

	mIOU			Pixel Accuracy		
	A05	**A10**	**Average**	**A05**	**A10**	**Average**
SideInfNet/Deeplab-ResNet	59.03%	35.45%	47.24%	89.68%	54.29%	71.99%
SideInfNet/VGG	66.34%	32.73%	49.53%	89.60%	46.50%	68.05%
Unified[8]/Deeplab-ResNet	47.94%	21.37%	34.66%	89.54%	40.42%	64.98%
Unified[8]/VGG	41.50%	17.23%	29.37%	91.38%	54.87%	73.12%

(c) Zurich [7]

	mIOU	**Pixel Accuracy**
SideInfNet/Deeplab-ResNet	58.31%	78.97%
SideInfNet/VGG	49.73%	77.74%
Unified[8]/Deeplab-ResNet	46.83%	74.26%
Unified[8]/VGG	42.09%	68.20%

Unified model [8], we provide results of SideInfNet and Unified model when the same baseline network is used. In particular, Workman et al. [8] proposed a modified VGG-16 network to extract features from the overhead satellite images, in which feature maps were integrated at the seventh convolutional layer. We re-implemented the same architecture by fusing our constructed feature map at the same layer. In addition, we also re-implemented the Unified model with Deeplab-ResNet, our recommended backbone. We show the comparison results in Table 3. As shown in experimental results, in general SideInfNet outperforms the Unified model [8] when the same baseline segmentation model is used, highlighting the advantages of our proposed method for multi-modal data fusion.

3 Computational Analysis

An additional advantage of our method is its computational efficiency, which comes into play with high density annotations. Specifically, the BACH dataset consists of very high resolution whole slide images, which is common in many

Table 4. Computational analysis performed on an NVIDIA Pascal Titan X GPU.

Approach	Time (s)			GPU Memory (MB)		
	Zoning	BACH	Zurich Summer	Zoning	BACH	Zurich Summer
Deeplab-ResNet [2]	0.047	0.101	0.048	779	821	781
Unified model* [8]	0.034	2.003	0.062	739	1843	725
SideInfNet	0.105	0.121	0.139	783	857	785

* Our implementation.

medical datasets. Coupled with dense brush stroke annotations, this results in significant bottlenecks for prior works, e.g., the Unified model [8].

In order to evaluate the computational complexity quantitatively, we benchmark the inference speeds of the Deeplab-ResNet [2], Unified model [8], and our SideInfNet. As the HO-MRF model requires an additional post-processing step in the form of global normalization, we do not compare against it in this experiment. Evaluation results are averaged across the inference speeds over single patches (i.e., batch size of 1). However, in practice the method can be sped up with batch based processing. For instance, with a batch size of 64, SideInfNet averages 0.057s per patch on the BACH dataset.

The results are summarized in Table 4. We observe that on datasets with smaller resolution images and sparser side information (e.g., the Zurich Summer dataset), the Unified model performs faster than SideInfNet. This is likely due to the multi-scale architecture of the Deeplab-ResNet, which increases the computational load as multiple images have to be processed. However, as we scale up to larger resolution images and denser side information, our method is much more efficient than the Unified model. In particular, on the BACH dataset which contains high resolution imagery and dense brush stroke annotations, we obtain approximately a 16 times speedup over the Unified model. This supports our hypothesis that on top of improved accuracy, SideInfNet is able to scale more efficiently to higher resolution images and denser side information.

4 Additional Qualitative Evaluations

4.1 Qualitative Results on BACH dataset

Several qualitative results of our method on the BACH dataset are as shown in Fig. 8. From the results presented, we observe that, compared with other methods, SideInfNet generally provides the highest quality results. The segmentation masks produced by SideInfNet are less noisy and sparse. In addition, compared with prior works, SideInfNet significantly produces less false positives.

A common challenge for SideInfNet and Unified model is the spaces demarcated by brush strokes, leading to segmentation results that only contain shape outlines, such as the circular object in the A05 slide. A possible solution to this

issue could be to perform global post-processing, e.g., by applying CRFs [4] or HO-MRFs [3]. However, these post-processing steps are computationally expensive and thus may not be feasible for high-resolution imagery data, e.g., the BACH images.

An alternative solution is applying manual post-processing. The refined results produced by SideInfNet allow these gaps to be easily filled in by users. These results suggest the viability of SideInfNet as a semi-automatic semantic segmentation tool.

4.2 Qualitative Results on Zurich Summer Dataset

Our qualitative results on the Zurich Summer dataset are presented in Fig. 9. As shown in the results, SideInfNet is able to draw a balance between fully automatic inference (e.g., Deeplab-ResNet), and completely manual segmentation (e.g., by a human expert). Our method produces much more accurate segmentation results as compared to the Unified model. For instance, as shown in the docks at the bottom right area in the *zh11* image, SideInfNet well distinguishes *Background* (white) from *Building* (gray). Docks are a relatively rare environmental feature, which make them difficult to be classified correctly. The Unified model, on the other hand, misclassifies this as *Buildings*.

SideInfNet also produces higher quality results compared to other models. The Unified model generates more dilated segmentation masks, while the baseline Deeplab-ResNet produces sparser masks.

SideInfNet is also able to accurately classifying smaller regions such as the *Bare Soil* region in the *zh5* image (see Fig. 9), which challenge other models. The segmentation results of SideInfNet on *Railway* class in the *zh8* image are also more coherent compared with other works.

References

1. Aresta, G., Araújo, T., Kwok, S., Chennamsetty, S.S., Safwan, M., Alex, V., Marami, B., Prastawa, M., Chan, M., Donovan, M., et al.: Bach: Grand challenge on breast cancer histology images. Medical image analysis (2019)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2018)
3. Feng, T., Truong, Q.T., Thanh Nguyen, D., Yu Koh, J., Yu, L.F., Binder, A., Yeung, S.K.: Urban zoning using higher-order markov random fields on multi-view imagery data. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 614–630 (2018)
4. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with gaussian edge potentials. In: Advances in neural information processing systems. pp. 109–117 (2011)
5. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)

6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
7. Volpi, M., Ferrari, V.: Semantic segmentation of urban scenes by learning local class interactions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–9 (2015)
8. Workman, S., Zhai, M., Crandall, D.J., Jacobs, N.: A unified model for near and remote sensing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2688–2697 (2017)

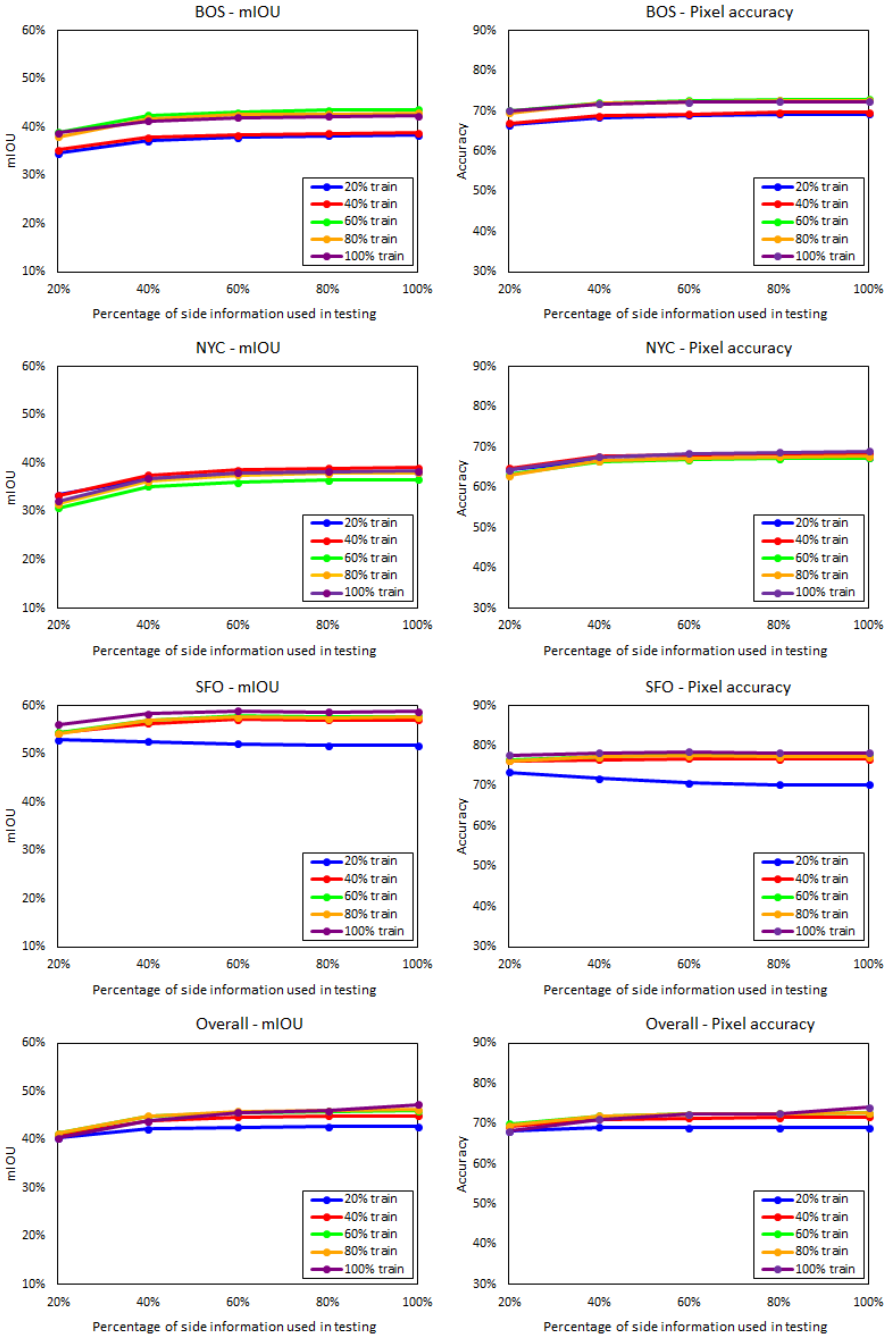

Fig. 1. Performance of SideInfNet when varying the availability of side information in both training and testing on zoning dataset [3].

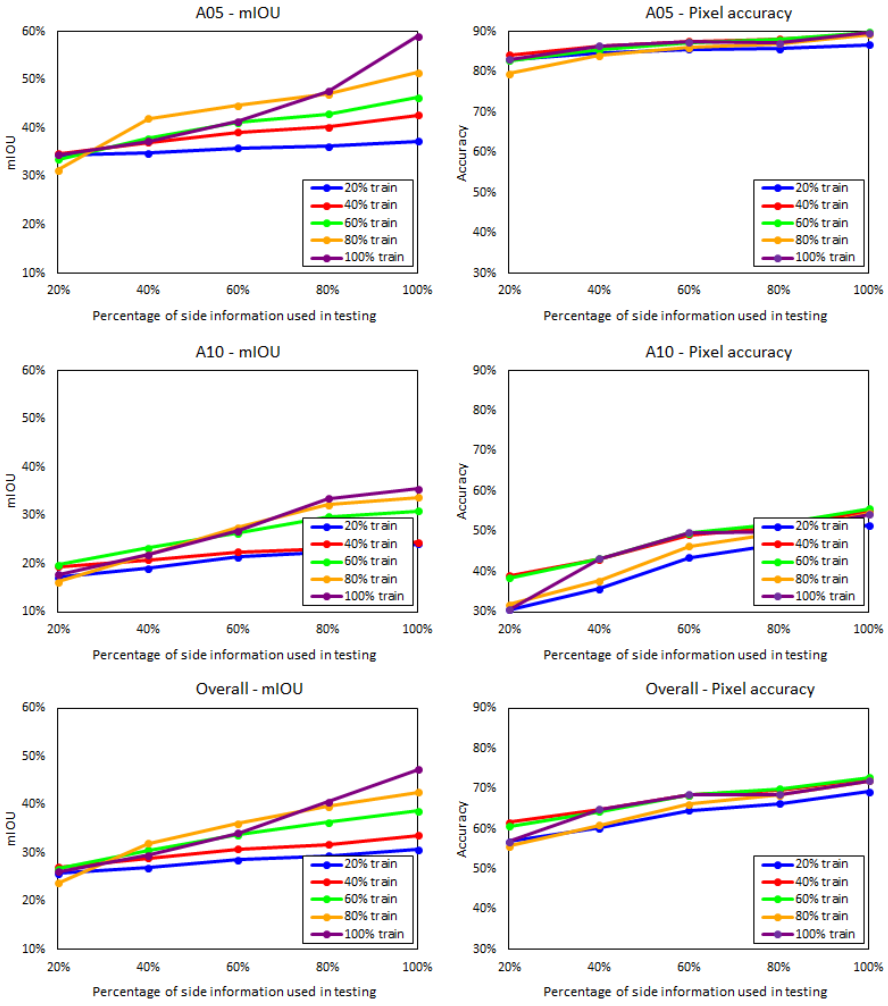

Fig. 2. Performance of SideInfNet when varying the availability of side information in both training and testing on BACH dataset [1].

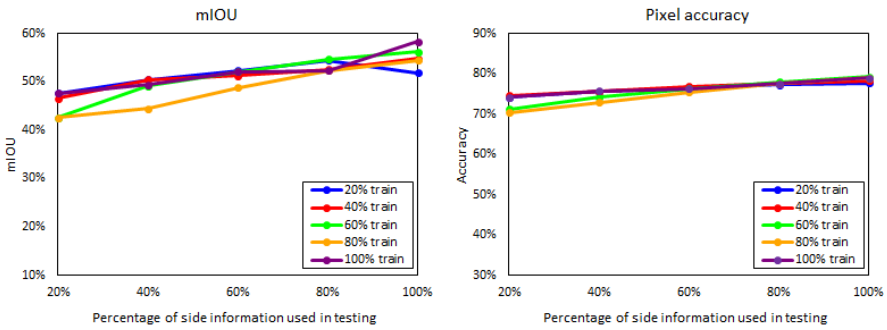

Fig. 3. Performance of SideInfNet when varying the availability of side information in both training and testing on Zurich Summer dataset [7].

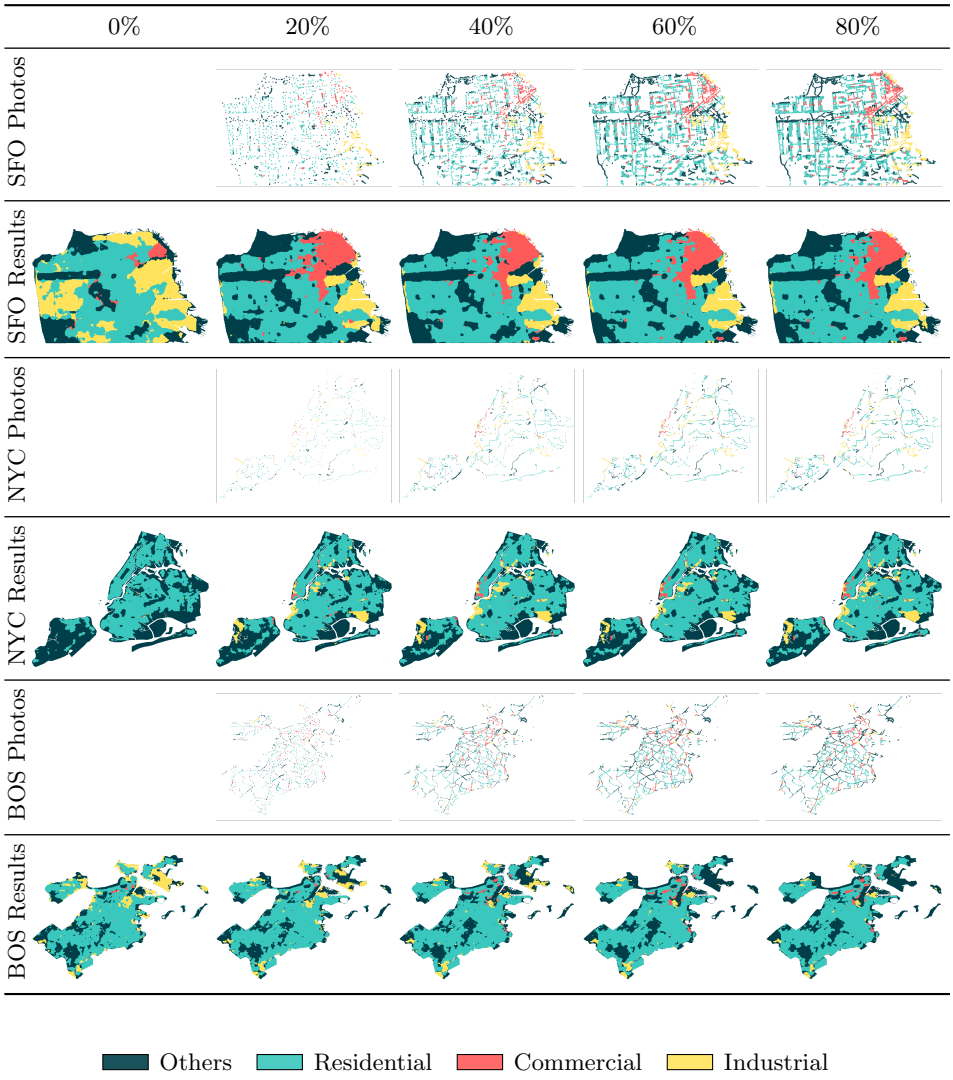

Fig. 4. Results on zone segmentation [3] with varying brush strokes. Best viewed in color.

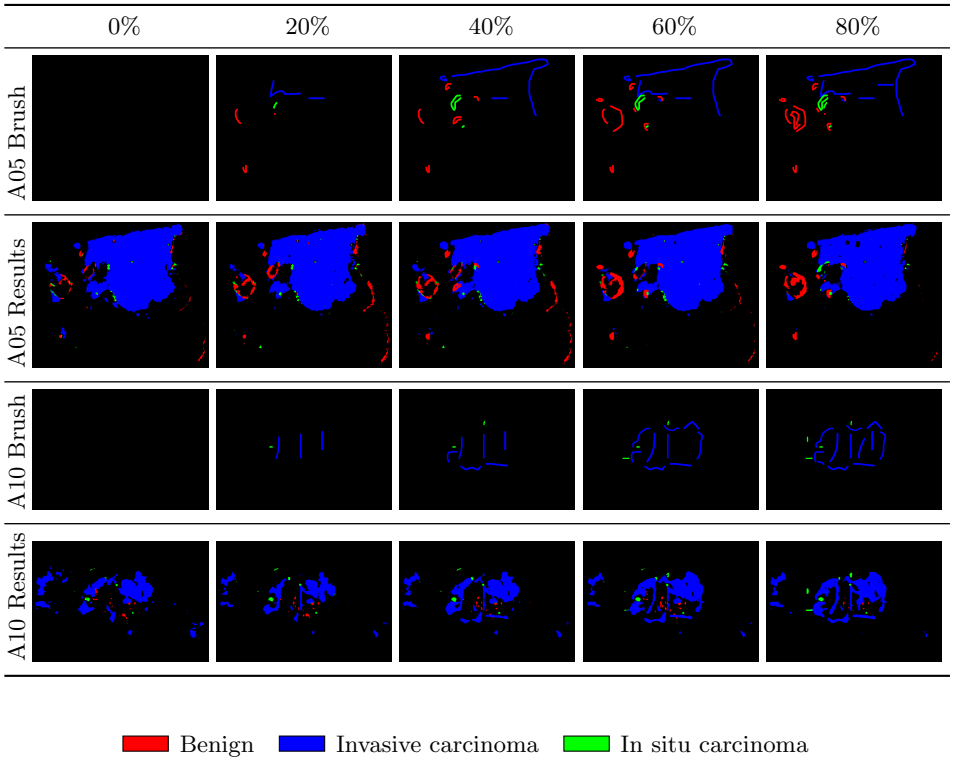

Fig. 5. Results on BACH [1] with varying brush strokes. Best viewed in color.

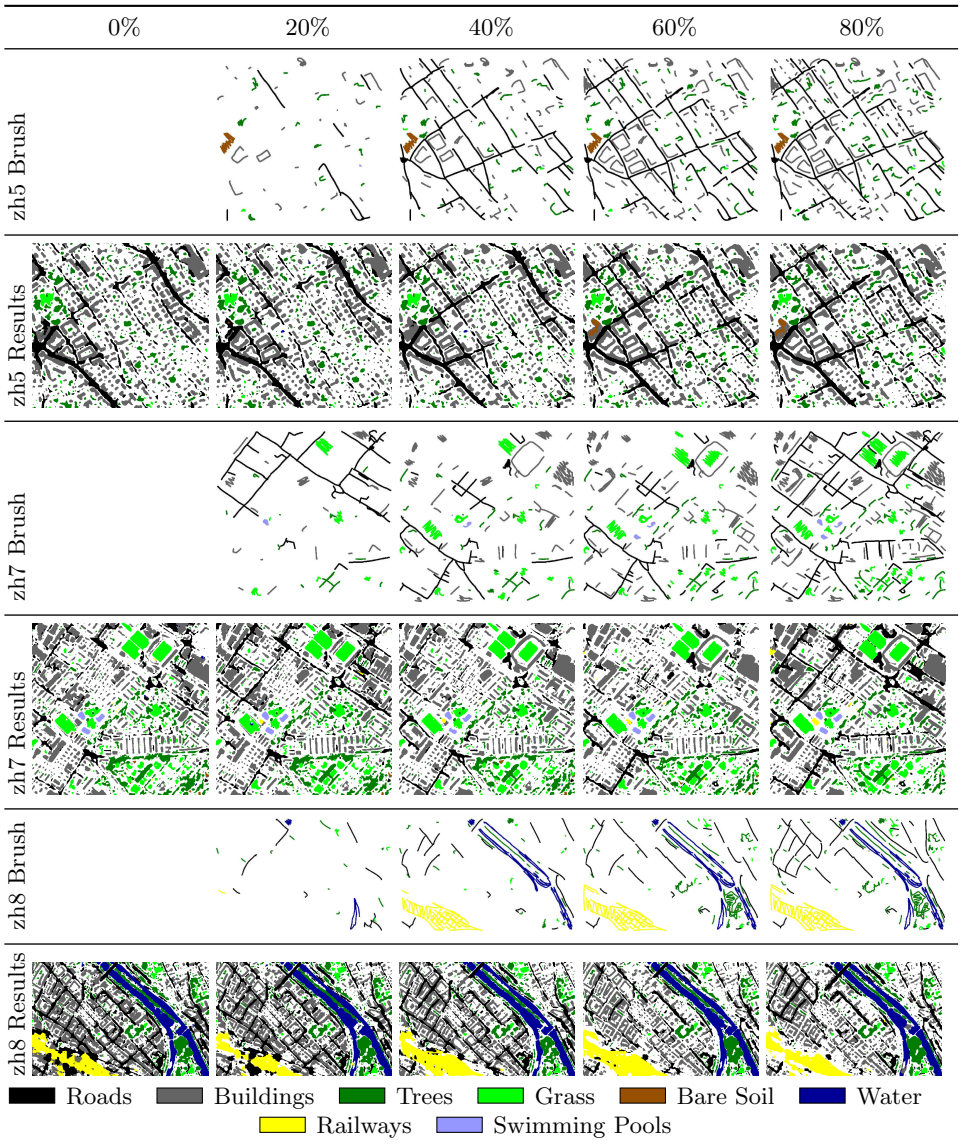

Fig. 6. Results on the Zurich Summer Dataset [7] with varying brush strokes. Best viewed in color.

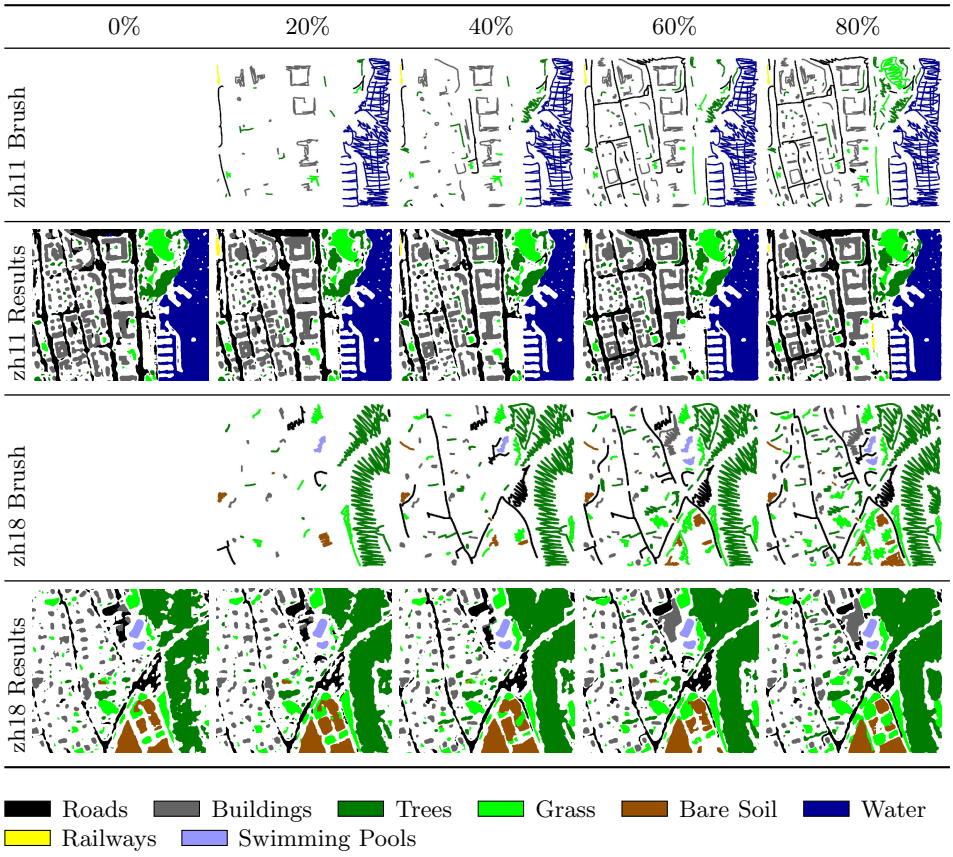

Fig. 7. Results on the Zurich Summer Dataset [7] with varying brush strokes. Best viewed in color.

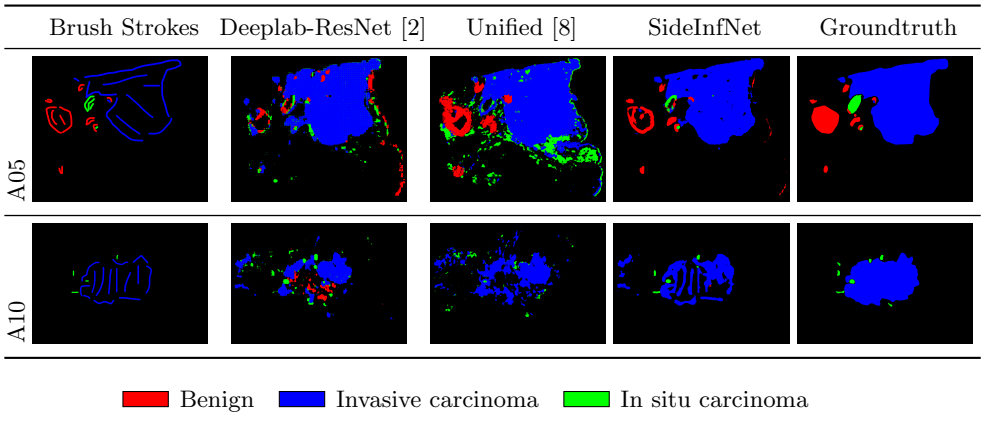

Fig. 8. Qualitative results on the BACH dataset [1]. Best viewed in color.

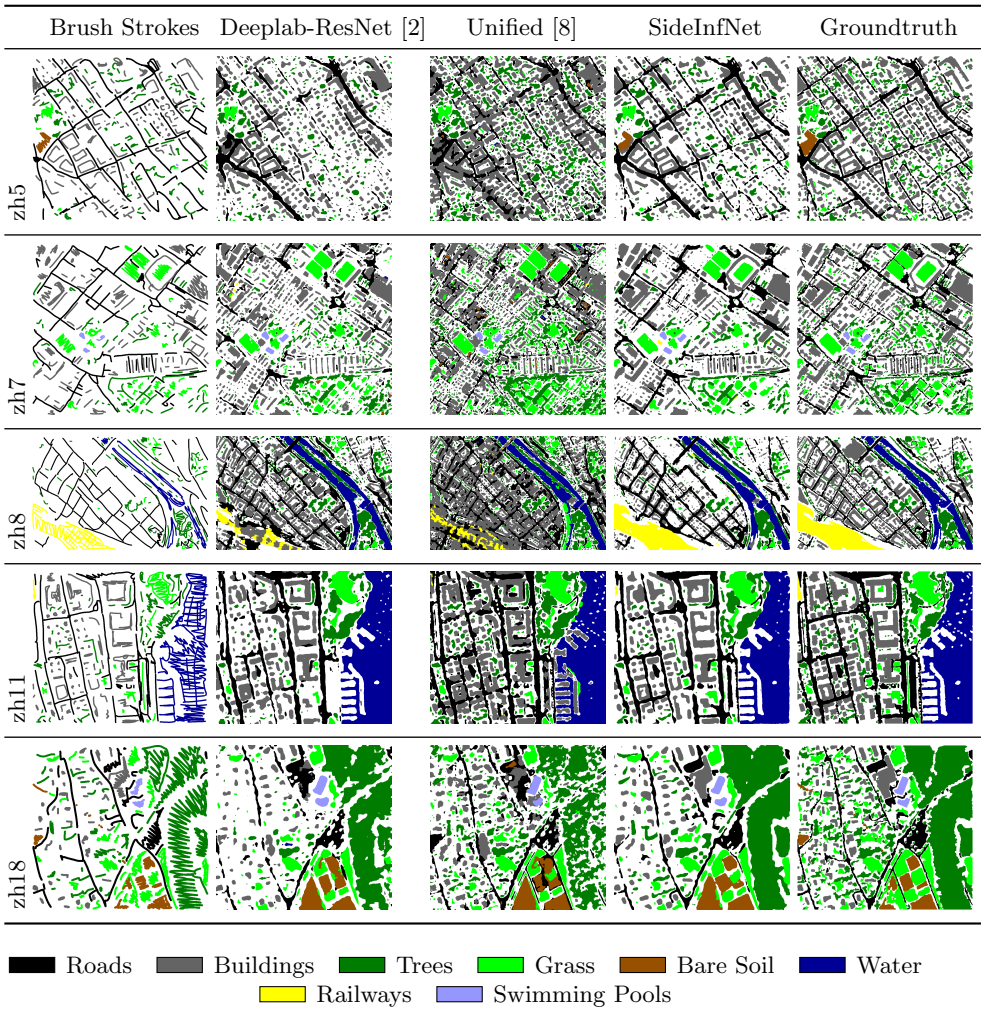

Fig. 9. Qualitative results on the Zurich Summer dataset [7]. Best viewed in color.