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

Jing Yu Koh1∗ Duc Thanh Nguyen2 Quang-Trung Truong1 Sai-Kit Yeung3
Alexander Binder1

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

Abstract

Fully-automatic execution is the ultimate goal for many Computer Vision applications. However, this objective is not always realistic in tasks associated with high failure costs, such as medical applications. For these tasks, a compromise between fully-automatic execution and user interactions is often preferred due to desirable accuracy and performance. Semi-automatic methods require minimal effort from experts by allowing them to provide cues that guide computer algorithms. Inspired by the practicality and applicability of the semi-automatic approach, this paper proposes a novel deep neural network architecture, namely SideInfNet that effectively integrates features learnt from images with side information extracted from user annotations to produce high quality semantic segmentation results. To evaluate our method, we applied the proposed network to three semantic segmentation tasks and conducted extensive experiments on benchmark datasets. Experimental results and comparison with prior work have verified the superiority of our model, suggesting the generality and effectiveness of the model in semi-automatic semantic segmentation.

1. Introduction

Most studies in Computer Vision tackle fully-automatic inference tasks which, ideally, would be able to perform perfectly without human intervention in execution. To achieve such ability, machine learning models are often well trained on diverse and rich datasets. However, the well-trained models may still fail in reality when dealing with unseen and/or ambiguous samples. This challenge could be resolved by using assistive information provided by users, e.g., user-provided brush strokes and bounding boxes [19].

Human interactions are particularly critical for tasks with high costs of failure, in which fully-automatic methods may not be viable. Examples include medical applications where predictions generated by computer algorithms always have to be verified by experts before they can be used in treatment plans. In such cases, a semi-automatic approach that allows incorporation of easy-and-fast human annotations may prove more reliable and preferable.

Semantic segmentation is an important Computer Vision problem with a wide spectrum of applications. Given an input image, the aim of semantic segmentation is to associate each pixel on the input image with a semantic class label. Recent studies in semantic segmentation have been built upon deep neural networks and achieved state-of-the-art performance [13, 8, 4, 5]. However, these methods are not flexible to be extended with additional information from various sources, such as human annotations or multi-modal data. In addition, human interactions are not allowed seamlessly and conveniently in existing methods.

In this paper, we propose SideInfNet, a general model that is capable of integrating domain knowledge learnt from domain data (e.g., images) with side information provided from user annotations in an end-to-end fashion. SideInfNet is built upon state-of-the-art convolutional neural network (CNN) based semantic segmentation models and adaptive to various annotation types. To speed up the inference process and reduce the computational cost while maintaining the quality of segmentation, adaptive inference gates are proposed to make the network’s topology flexible and optimal.

A key challenge in designing such a model is how to make it general to different sparsity and modalities of side information. Existing work focuses on sparse pixel-wise annotations, such as geotagged photos [7, 25] or user-defined keypoints [22, 15]. However, these methods may not perform optimally when the annotations are non-uniformly
and/or poorly localized, e.g., brush strokes which can be drawn dense and intertwined. SideInfNet provides competitive results under these situations and outperforms existing models in various tasks and on different datasets. Importantly, our model provides a principled compromise between fully-automatic and manual segmentation. The benefit gained by the model is well shown in tasks where there exists a mismatch between training and test distribution. A few brush strokes can drastically improve the performance on these tasks. We show the versatility of our proposed model on three tasks:

- **Zone segmentation** of low-resolution satellite imagery [7]. Geotagged street-level photographs from social media are used as side information.

- **BreAst Cancer Histology (BACH) segmentation** [2]. Whole-slide images are augmented with expert-created brush strokes to segment the slides into possible normal, benign, in situ carcinoma and invasive carcinoma regions.

- **Urban segmentation** of very high-resolution (VHR) overhead crops taken of the city of Zurich [24]. Brush annotations indicate particular geographic features and are augmented with imagery features to identify eight different urban and peri-urban classes for VHR images from the Zurich Summer dataset [24].

2. Related Work

The idea of incorporating user annotations and multiple modalities for semantic segmentation has been explored under various contexts.

2.1. Interactive Segmentation

A classical seminal work of interactive segmentation is the GrabCut [19] that tackles the problem of foreground-background segmentation and works in an unsupervised manner without the need for training. The method allows users to provide interactions in the form of brush strokes and bounding boxes demarcating foreground objects. However, a significant drawback of the original GrabCut is that it is not directly extensible to semantic segmentation on multiple classes.

Several methods have been proposed for extending the GrabCut framework for semantic segmentation and instance segmentation, e.g., [9, 26]. However, these methods only support bounding box annotations and thus cannot be used in datasets containing irregular object shapes, such as non-rectangular zones in the Zurich Summer dataset [24].

2.2. Semantic Segmentation for Remote Sensing

Literature has demonstrated the advantages of fusing overhead and ground-level imagery data for semantic segmentation in remote sensing. For instance, CNNs were utilized for learning features from satellite images and street-level photos in [7]. These learnt features were then fused in a Higher-Order Markov Random Field (HO-MRF) model for segmenting zones in the satellite images. However, this method is not trainable in an end-to-end fashion.

Workman et al. [25] proposed a model for fusing multi-view imagery data for estimating geospatial functions land cover and land use. While this method is general, it requires heavy computer memory for its operation, e.g., for calculating and storing $k$ nearest annotations, and thus may not be tractable for tasks with high density annotations. Similarly, in [14], multi-view imagery data, including aerial and ground images, were used to enhance the quality of fine-grained road segmentation.

2.3. Keypoint Annotations

Annotations can be provided via keypoints. The effectiveness of oracle keypoints in human segmentation is illustrated in [22]. Similarly, in [15], a method for automatically learning keypoints was proposed. The keypoints are grouped into pose instances and used for instance segmentation of human subjects. However, while the spatial layout of keypoints is important to represent a meaningful human body structure, such constraint is not always held for other object types, such as cell masses in histopathology.

2.4. Semi-Supervised Segmentation

Semi-supervised methods are also often considered for segmenting user-defined objects in video data [17, 3, 20, 11]. These methods utilize user annotations to determine the target objects. The annotations provide prior and reliable cues to guide the segmentation process over time. For instance, Perazzi et al. [17] introduced a CNN-based guidance method for video segmentation by allowing users to provide object bounding boxes or regions. They also showed that increasing the number of user annotations led to improved segmentation quality. In a similar manner, Nagaraja et al. [20] tackled the task of object segmentation from video by combining motion cues and user annotations. In their work, users provided scribbles to delineate the objects of interests. Experimental results verified the cooperation of sparse user annotations and motion cues, filling the gap between fully automatic and manual object segmentation.

3. SideInfNet

SideInfNet is a novel deep neural network that tackles the fusion of domain knowledge and user-provided side information into an end-to-end trainable architecture. SideInfNet allows the incorporation of multi-modal data. It is flexible with different types of annotations and adaptive to various segmentation network architectures. SideInfNet is built upon state-of-the-art semantic segmentation [13, 5, 18] and
recent advances in adaptive neural networks [23, 21]. This combination allows our model to define its topology dynamically on the fly, reduce computational complexity, and produce high quality segmentation results.

For the sake of ease and simplicity in presentation, we describe our method in regard to a particular case study: zone segmentation. However, as shown in our experiments, our method can be applied in other scenarios.

Zone segmentation aims to provide a zoning map for an aerial image, i.e., to segment the aerial image into different regions and identify the zone types for those regions. Side information in this case includes street-level photos. These photos are provided by users and associated with geocodes that refer to their spatial locations on the aerial image. Domain-dependent features on the input aerial image are first extracted at multiple scales using some CNN-based semantic segmentation model (see Section 3.1). Side information features are then constructed from user-provided street-level photos (see Section 3.2). Side information feature maps are also computed at multiple scales accordingly with their counterpart domain-dependent feature maps. Associated geocodes in the street-level photos help to identify their locations in the receptive fields in the SideInfNet architecture where both domain-dependent and side information features are fused. To reduce the computational cost of the model while not sacrificing the quality of segmentation, adaptive inference gates are proposed to skip layers conditioned on input (see Section 3.3). Figure 1 illustrates the workflow of SideInfNet whose components are described in detail in the following subsections.

3.1. CNN-based Semantic Segmentation

To extract domain-dependent features (on input images), we adopt the Deeplab-ResNet proposed in [5], a state-of-the-art CNN-based semantic segmentation. Deeplab-ResNet makes use of a series of atrous (dilated) convolutional layers, with increasing rates to aggregate multi-scale features. To adapt Deeplab-ResNet into our framework, we retain the same network architecture but extend the $conv2_3$ layer with side information (presented in Section 3.2).

Specifically, the side information feature map is concatenated to the output of the $conv2_3$ layer of Deeplab-ResNet in the channels dimension (see Figure 1). As the original $conv2_3$ layer outputs a feature map with 256 channels, concatenating the side information feature map results in a $H \times W \times (256 + d)$ dimensional feature map where $H$ and $W$ is the height and width of the input image, and $d$ is the number of channels of the side information feature map. This extended feature map is the input to the next convolutional layer, $conv3_1$. Whenever the side information feature map can be constructed automatically, the entire SideInfNet can be trained end-to-end using standard backpropagation on a cross-entropy loss.

To evaluate the effectiveness of including side information, we compare our extended architecture against the original Deeplab-ResNet in Section 4.
3.2. Side Information Feature Map Construction

Due to the variation in the modality of side information, several preprocessing steps are required to effectively fuse the side information with domain-dependent features. Depending on applications, specific processing can be applied on the side information.

For instance, for the application to zoning, we use the Places365-CNN proposed in [27] to create vector representations of street-level photos (see details in Section 4.1.1). These vectors are then passed through a fully-connected layer that maps them into a d-dimensional vector space. Suppose that the input aerial image is of size $H \times W$. A side information feature map $x^l$ of size $H \times W \times d$ can be created by initializing the $d$-dimensional vector at every location in $H \times W$ with the feature vector of the corresponding street-level photo, if one exists there. The feature vectors at locations that are not associated to any street-level photos are padded with zeros. Mapping image locations to street-level photos can be done using the associated geocodes of the street-level photos. Nearest neighbor interpolation is also applied on the side information feature map to create multi-scale features. Nearest neighbor interpolation is used to minimize distortion of features. Features that fall in the same image locations (on the aerial image) due to downsampling are averaged. To make feature vectors consistent across scales and data samples, all feature vectors are normalized to the unit length.

There may exist misalignment in associating the side information features with their corresponding locations on the side information feature map. For instance, a brush stroke provided by a user may not well align with a true region. In zoning application, a street-level photo may not record the scene at the location where the photo is captured, but a nearby scene. Therefore, a direct reference of a street-level photo to a location on the feature map via the photo’s geocode may not be a perfect association. However, one could expect that the side information could be propagated from nearby locations. To address this issue, we apply a series of fractionally-strided convolutions to the normalized feature map $x^l$ to distribute the side information spatially. In our implementation, we use $3 \times 3$ kernels of ones, with stride length 1 and padding of 1. After a single fractionally-strided convolution, side information features are distributed onto neighbouring $3 \times 3$ regions. We repeat this operation (denoted as $f^l_c$) $n$ times in series and sum up all the feature maps in the entire series to create the features for the next layer as follows,

$$x^{l+1} = F(x^l) = \sum_{i=1}^{n} w_i f^l_c(x^l)$$

where $w_i$ are learnable parameters and $f^l_c$ is the $i$-th functional power of $f_c$, i.e.,

$$f^l_c(x^l) = \begin{cases} f_c(x^l), & i = 1 \\ f_c(f^l_{i-1}(x^l)), & \text{otherwise} \end{cases}$$  

(2)

The parameters $w_i$ in (1) allow our model to automatically learn the importance of spatial extent. We observe that, after training, the values of the learnt parameters $w_i$ generally follow a decreasing pattern (i.e., $w_1 > w_2 > \cdots$). This matches our intuition that information is likely to become less relevant with increased distances. The resulting feature map $x^{l+1}$ represents a weighted sum of nearby feature vectors. We also normalize the feature vector at each location in the feature map by the number of the fractionally-strided convolutions used at that location. This has the effect of averaging overlapping features.

Lastly, we perform maxpooling to further downsample the side information feature map to fit with the size of the counterpart domain-dependent feature map for feature fusion. In our implementation, we choose to perform feature fusion before the second convolutional block of DeeplabV3+ with the output of the conv2d_3 layer. We empirically found that this provided a good balance between computational complexity and segmentation quality. The output of the maxpooling layer is concatenated in the channels dimension to the output of the original layer, as shown in Figure 1. It is important to note that our proposed side information feature map construction method is general and can be applied alongside any CNN-based semantic segmentation architectures.

3.3. Adaptive Architecture

Inspired by recent advances in adaptive neural networks [23, 21], we propose the use of adaptive inference graphs in designing the architecture of SideInfNet. Adaptive inference graphs make use of adaptive gates $z^l$ to decide skip-connections in the inference process in network architecture. Specifically, we define,

$$x^{l+1} = x^l + z^l(h(x^l)) \cdot F(x^l)$$

(3)

where $z^l(x^l) \in \{0, 1\}$ and $h$ is some function that maps $x^l \in H \times W \times d$ into a lower-dimensional space of $1 \times 1 \times d$. The gate $z^l$ is conditioned on $x^l$ and takes a binary decision (“on” for 1 and “off” for 0).

Like [23], we set the early layers and the final classification layer of our model to always be executed, as these layers are critical for maintaining the accuracy. The gates are included in every other layer. The mapping function $h$ in (3) aims to compress the input feature map $x^l$ into a compact representation. In particular, we define,

$$h(x^l) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x^l_{i,j}$$

(4)
This compact feature map is passed into a multi-layer perceptron, which computes a relevance score that determines whether layer \( l \) is executed. In addition, we have a hyperparameter \( t \), the gate target rate, that determines what fraction of layers should be activated. For example, a target rate \( t = 0.8 \) imposes a penalty on the loss function when the proportion of layers executed is greater or less than 80%. This is implemented as a mean squared error loss and jointly optimized with the cross entropy loss. Our experimental results on this adaptive model are presented in Section 4, where we find that allowing a proportion of layers to be skipped helps improve segmentation quality.

### 4. Experiments and Results

In this section, we extensively evaluate our proposed SideInfNet in three different case studies.

#### 4.1. Zone Segmentation

##### 4.1.1 Experimental Setup

Like [7], we conducted experiments on three US cities: Boston (BOS), New York City (NYC), and San Francisco (SFO). Freely available satellite images hosted on Microsoft Bing Maps [6] were used. The ground-truth maps were retrieved at a service level of 12, which corresponds to a pixel resolution of 38.2185 meters per pixel. An example of the satellite imagery is shown in Figure 2. We retrieved street-level photos from Mapillary [1], a service for sharing crowdsourced geotagged photos. There were four zone types: Residential, Commercial, Industrial and Others. Table 1 summarizes the dataset used in this case study.

![Figure 2. Satellite image of San Francisco from Bing Maps.](image)

Table 1. Proportion of street-level photos (#photos).

| Zone Type | BOS    | NYC    | SFO    |
|-----------|--------|--------|--------|
| Residential | 25,607 | 16,395 | 50,116 |
| Commercial | 13,412 | 5,556  | 19,641 |
| Industrial | 2,876  | 9,327  | 15,219 |
| Others    | 25,402 | 15,281 | 50,214 |

We applied a series of \( n = 5 \) fractionally-strided convolutions on feature maps generated from Places365-CNN. This acts as to distribute the side information from each geotagged photo 5 pixels in each cardinal direction.

##### 4.1.2 Results

We evaluate our method and compare it with two recent works: Higher-Order Markov Random Field (HO-MRF) [7] and Unified model [25] using 3-fold cross validation, i.e., two cities are used for training and the other one is used for testing. To have a fair comparison, the same Places365-CNN model is used to extract side information in all methods. We also compare our method against the baseline Deeplab-ResNet, which directly performs segmentation of satellite imagery without using geotagged photos.

Our results on both pixel accuracy and mean intersection over union (mIOU) are as summarized in Table 2. As shown in the table, our method significantly improves its baseline, Deeplab-ResNet, proving the importance of side information. SideInfNet also outperforms the prior works, with a relative improvement in pixel accuracy from the Unified model by 3.38% and from the HO-MRF by 7.92%. Improvement on mIOU scores is also significant, e.g., by 4.97% relatively compared with the Unified model, and 25.07% relatively compared with the HO-MRF.

In addition to improved accuracy, our method offers several advantages over the previous works. First, compared with the HO-MRF [7], our method is trained end-to-end, allowing it to jointly learn optimal parameters for both semantic segmentation and side information feature extraction. Second, our method is efficient in computation. It simply performs a single forward pass through the network to produce segmentation results, opposed to iterative inference in the HO-MRF [7]. Third, by using fractionally-strided convolutions, the complexity of our method is invariant to the side information density. This allows optimal performance on regions with high density of side information. In contrast, the Unified model [25] requires exhaustive searches to determine nearest street-level photos for very pixel on satellite image and thus depend on the density of the street-level photos and the size of the satellite image.

We qualitatively show the segmentation results of our method and other works in Figure 3. A clear drawback of
the HO-MRF is that the results tend to be grainy, likely due to the sparsity of street-level imagery. In contrast, our method generally provides smoother results that form contiguous regions. In addition, we also observe that our method is able to better capture fine grained details from street-level imagery.

4.2. Breast Cancer Histology Segmentation

4.2.1 Experimental Setup

BACH (Breast Cancer Histology) [2] is a dataset for breast cancer histology microscopy segmentation. This dataset consists of high resolution whole-slide images that contain an entire sampled tissue. The whole-slide images were an-

---

Table 2. Experimental results of zone segmentation. Best performances are highlighted.

| Approach          | Accuracy  | mIOU  |
|-------------------|-----------|-------|
|                   | BOS  | NYC  | SFO  | Average | BOS  | NYC  | SFO  | Average |
| Deeplab-ResNet [5]| 60.79% | 59.58% | 72.21% | 64.19% | 28.85% | 23.77% | 38.40% | 30.34% |
| HO-MRF* [7]       | 59.52% | 72.25% | 73.93% | 68.57% | 31.92% | 34.99% | 46.53% | 37.81% |
| Unified* [25]     | 67.91% | 70.92% | 75.92% | 71.58% | 40.51% | 39.27% | 55.36% | 45.05% |
| SideInfNet        | 71.33% | 71.08% | 79.59% | 74.00% | 41.96% | 39.59% | 60.31% | 47.29% |

* Our implementation.
notated by two medical experts, and images with disagreements were discarded. There are four classes: normal, benign, in situ carcinoma and invasive carcinoma. An example of a whole-slide image and its labels is shown in Figure 4. As the normal class is considered background, it is not evaluated. Side information for BACH consists of expert brush stroke annotations, indicating the potential presence of each class. In this case study, we use four different brush stroke colors to annotate the four classes.

BACH dataset does not include actual expert-annotated brush strokes. Therefore, to evaluate our method, we simulated expert annotations by using ground-truth labels in the dataset. Since the ground-truth was created by two experts, our brush strokes can be viewed as simulated rough expert input. To simulate situations where users have limited annotation time, we skipped annotating small regions that are likely to be omitted under time constraints. Figure 4 shows an example of our simulated brush strokes.

In our experiments, we used slides A05 and A10 for testing, slide A02 for validation, and all other slides for training.

4.2.2 Results

We evaluate three different methods: our proposed SideInfNet, Unified model [25], and GrabCut [19]. We were unable to run the HO-MRF model [7] on the BACH dataset due to the large size of the whole-slide images (note that the HO-MRF makes use of fully-connected MRF and thus is not computationally feasible under this context). In addition, since GrabCut is a binary segmentation method, to adapt this work to our case study, we ran the GrabCut model independently for each class. We report the performance of all the methods in Table 3. We also provide some qualitative results in Figure 5.

| Approach          | mIOU A05 | mIOU A10 | mIOU Average |
|-------------------|----------|----------|--------------|
| Deeplab-ResNet [5]| 34.08%   | 21.64%   | 27.86%       |
| GrabCut† [19]     | 30.20%   | 25.21%   | 27.70%       |
| Unified∗ [25]     | 41.50%   | 17.23%   | 29.37%       |
| SideInfNet        | 59.03%   | 35.45%   | 47.24%       |

† Averaged over binary segmentation scores.  
* Our implementation.

Experimental results showed that our method greatly outperforms previous works on BACH dataset. We also found that the Unified model even performed worse than the baseline Deeplab-ResNet that used only whole-slide imagery. This suggests the limitation of the Unified model [25] in learning from dense annotations. Table 3 also confirms the role played by the side information (i.e., the Deeplab-ResNet vs SideInfNet). This aligns with our intuition, as we would expect that brush strokes provide stronger cues to guide the segmentation.
4.3. Urban segmentation

4.3.1 Experimental Setup

The Zurich Summer v1.0 dataset [24] is a collection of 20 very high resolution (VHR) overview crops taken of the city of Zurich, pansharpened to a PAN resolution of about 0.62 centimeters ground sampling distance (GSD). This is a much higher resolution compared to the low-resolution satellite imagery used in the zoning dataset (Section 4.1) which has a GSD of approximately 38 meters. The Zurich Summer dataset contains eight different urban and periurban classes: Roads, Buildings, Trees, Grass, Bare Soil, Water, Railways and Swimming pools. Examples of satellite imagery, ground-truth labels, and brush annotations are shown in Figure 6.

Preprocessing steps and feature map construction are performed similarly to that of BACH. In this case study, we also use rough brush strokes demarcating potential urban classes as side information.

4.3.2 Results

Our experimental results on the Zurich Summer dataset are summarized in Table 4. In general, similar trends with the BACH dataset are found, and our proposed method outperforms all prior works. We qualitatively compare our method with other works in Figure 7.

On the Zurich Summer dataset, SideInfNet greatly outperforms all prior works. By using brush stroke data, we are able to gain a relative improvement of 7.88% on accuracy and 35.76% on mIOU over the baseline Deeplab-ResNet. The Zurich dataset contains high-resolution satellite imagery, which suggests the usefulness of including brush annotations even with high fidelity image data. SideInfNet also outperforms the Unified model, with a relative improvement of 15.79% on accuracy and 38.53% on mIOU. This result proves the robustness of our method in dealing with dense annotations, which challenge the Unified model.

Table 4. Experimental results on the Zurich Summer dataset. Best performances are highlighted.

| Approach           | Accuracy | mIOU  |
|--------------------|----------|-------|
| Deeplab-ResNet [5] | 73.20%   | 42.95%|
| GrabCut† [19]      | 90.53%   | 26.89%|
| Unified* [25]      | 68.20%   | 42.09%|
| SideInfNet         | 78.97%   | 58.31%|

† Averaged over binary segmentation scores.
* Our implementation.

GrabCut also under-performs due to its limitations as an unsupervised binary segmentation method.

5. Conclusion

This paper proposes SideInfNet, a novel adaptive end-to-end convolutional neural network architecture for semi-automatic semantic segmentation with additional side information. Through extensive experiments on various datasets and modalities, we have shown the advantages of our method over prior works. Our model is applicable to a wide range of applications, including but not limited to remote sensing and medical image segmentation. In addition to being general, our method boasts improved accuracy and computational advantages over prior models. Lastly, our architecture is easily adapted to various semantic segmentation models and side information feature extractors.

The method proposed in this paper acts as a compromise between fully-automatic and manual segmentation. This is essential for many fields with high cost of failure, in which fully-automatic methods may not be widely accepted as of yet. Our model works well with dense brush stroke information, providing a quick and intuitive way for human experts to refine model outputs. In addition, our model also outperforms prior work on sparse pixel-wise annotations. By including side information to shape predictions, we are
able to achieve an effective ensemble of human expertise and machine efficiency, producing both fast and accurate segmentation results.

References

[1] Mapillary AB. Mapillary, 2019. https://www.mapillary.com, Last accessed on 2019-11-01. 5

[2] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, et al. Bach: Grand challenge on breast cancer histology images. Medical image analysis, 2019. 2, 6, 7, 11, 12, 13, 16, 19

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 221–230, 2017. 2

[4] 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. 1

[5] 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, 40(4):834–848, 2018. 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 19, 20

[6] Microsoft Corporation. Bing maps tile system, 2019. https://msdn.microsoft.com/en-us/library/bb259689.aspx, Last accessed on 2019-11-01. 5

[7] Tian Feng, Quang-Trung Truong, Duc Thanh Nguyen, Jing Yu Koh, Lap-Fai Yu, Alexander Binder, and Sai-Kit Yeung. Urban zoning using higher-order markov random fields on multi-view imagery data. In Proceedings of the European Conference on Computer Vision (ECCV), pages 614–630, 2018. 1, 2, 5, 6, 7, 10, 11, 12, 13, 14, 15

[8] 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, pages 580–587, 2014. 1

[9] Christoph Göring, Björn Fröhlich, and Joachim Denzler. Semantic segmentation using grabcut. In VISAPP, pages 597–602, 2012. 2

[10] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In Advances in neural information processing systems, pages 109–117, 2011. 14

[11] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qiu Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6526–6535, 2018. 2

[12] 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, pages 740–755. Springer, 2014. 10

[13] 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, pages 3431–3440, 2015. 1, 2

[14] Gellért Mártyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Hd maps: Fine-grained road segmentation by parsing ground and aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3611–3619, 2016. 2

[15] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In Proceedings of the European Conference on Computer Vision (ECCV), pages 269–286, 2018. 1, 2

[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 10

[17] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2663–2672, 2017. 2

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015. 2
Supplementary Material

In this supplementary material, we provide implementation details of our method in Section A. We present ablation experiments conducted to ascertain the effectiveness of our method in Section B, 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 C. We present additional qualitative evaluations of our method and prior works in three case studies in Section D.

A. 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 [16].

A.1. General Settings

The domain-dependent feature extractor of our proposed SideInfNet is built based on the Deeplab-ResNet [5] 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 $\text{conv}2\_3$ layer output of approximately size $H/4 \times W/4$. 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. We also experimented with deeper multi-layer perceptrons (MLP) and non-linear activation functions, but found no improvement from these settings.

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 [12]. Due to concatenation of data in our models, the weights of the layers at the concatenation point (i.e., the $\text{conv}2\_3$ 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 $\text{conv}2\_3$ layer from the MS-COCO pretrained weights, and randomly initialized the additional 64 channels.

A.2. Zone Segmentation

A.2.1 Training

For training, we used $80 \times 80$ pixels crops of each city from the zone segmentation dataset [7]. 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 per-
Table 5. Performance of variants of SideInfNet in zone segmentation [7].

| Approach         | Street Imagery | Fractionally Strided Convolutions | Gate Target Rate \( (t) \) | Pixel Accuracy |
|------------------|----------------|-----------------------------------|-----------------------------|---------------|
|                  |                |                                   |                             | BOS           |
| Deeplab-ResNet [5] | -              | -                                 | -                          | 60.79%        |
| Geotagged        | ✓              | -                                 | -                          | 60.19%        |
| Diffused         | ✓              | ✓                                 | 0.8                         | 71.33%        |
| SideInfNet       | ✓              | ✓                                 | 0.6                         | 70.45%        |
| SideInfNet       | ✓              | ✓                                 | 0.4                         | 70.58%        |

forming 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.

A.2.2 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.

A.3. Breast Cancer Histology Segmentation

A.3.1 Training

Due to the large size of whole-slide images in the Breast Cancer Histology (BACH) dataset [2], we downscaled the whole-slide images for computational efficiency and for ease of training. 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 with mean value derived from the 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.

A.3.2 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.
Table 6. Performance of SideInfNet with VGG as the segmentation model.

| Model          | mIOU | Zoning [7] | BACH [2] | Zurich [24] |
|----------------|------|------------|----------|-------------|
| SideInfNet-VGG | 46.12% | 49.53% | 49.73% |
| Unified [25]  | 45.05% | 29.37% | 42.09% |

A.4. Urban Segmentation

For urban segmentation on the Zurich Summer Dataset [24], 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 zh11, 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.

B. Ablation Studies

B.1. SideInfNet with VGG-16 for Segmentation

To provide a fair comparison between our proposed method of fusion and the method outlined in [25], we provide results of SideInfNet when the same baseline segmentation network is used. [25] proposes a modified VGG-16 network to extract features from the overhead satellite images, in which feature maps are integrated at the seventh convolutional layer. We implement the same architecture by fusing our constructed feature map at the same layer. The results are as summarized in Table 6. Our model that utilizes the proposed VGG-16 segmentation network is denoted as SideInfNet-VGG. We utilized the original hyperparameters proposed in [25] without further hyperparameter search for our approach.

We observe that SideInfNet is able to outperform the Unified model [25] when the same baseline segmentation model is used, highlighting the advantages of our proposed method for feature map construction and fusion.

In addition to the increase in performance, we also note that the proposed VGG-19 backbone is computationally more efficient than our original Deeplab-ResNet backbone due to the lack of multiscale processing. Hence, SideInfNet-VGG is significantly faster during inference compared to both the Unified and the original SideInfNet models, at the cost of a drop in accuracy compared to the original SideInfNet model.

B.2. 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 5. 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 [5] 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.

B.3. Varying Brush Strokes

A possible concern with interactive tools is how they perform with varying levels of user interactions. In this ablation study, we investigate the performance of our method when varying the availability of brush strokes.

To simulate various densities of brush strokes for an input image, we sample the original brush strokes (e.g., from 0% to 100% of the total number) and evaluate the segmentation performance of our method accordingly. The brush strokes could be randomly sampled. However, this approach may bias the spatial distribution of the brush strokes. To maintain the spatial distribution of the brush strokes as same as possible for every sampling case, instead of randomly sampling the brush strokes, we perform k-means clustering on the original set of the brush strokes. For example, if we wish to utilize a percentage p of the total brush strokes, and n brush strokes are present in total, we apply k-means algorithm with k = ceil(np) on the centers of the brush strokes to spatially cluster the brush strokes into k groups. For each group, we select the brush stroke whose
Table 7. Performance of SideInfNet with varying brush strokes.

| Side Information Used | mIOU          | Mean Accuracy          |
|-----------------------|---------------|------------------------|
|                       | Zoning [7] | BACH [2] | Zurich [24] | Zoning | BACH | Zurich |
| 100%                  | 47.29%     | 47.24% | 58.31%     | 74.00% | 71.99% | 78.97% |
| 80%                   | 40.27%     | 40.53% | 52.32%     | 72.46% | 68.60% | 77.58% |
| 60%                   | 39.56%     | 34.16% | 52.14%     | 72.39% | 68.56% | 76.33% |
| 40%                   | 37.70%     | 29.66% | 49.49%     | 71.01% | 64.87% | 75.83% |
| 20%                   | 34.04%     | 26.15% | 47.72%     | 68.11% | 56.86% | 74.29% |
| 0%                    | 28.11%     | 23.86% | 45.98%     | 58.63% | 60.48% | 73.36% |

Table 8. Computational analysis performed on an NVIDIA Pascal Titan X GPU.

| Approach          | Time (s) | GPU Memory (MB) |
|-------------------|----------|-----------------|
|                   | Zone [7] | BACH [2] | Zurich Summer [24] | Zone | BACH | Zurich Summer |
| Deeplab-ResNet [5] | 0.047     | 0.101 | 0.048     | 779 | 821 | 781         |
| Unified* [25]     | 0.034     | 2.003 | 0.062     | 739 | 1843 | 725         |
| SideInfNet        | 0.105     | 0.121 | 0.139     | 783 | 857 | 785         |

* Our implementation.

center is closest to the group’s centroid. This step results in \( k \) brush strokes. We note that a similar procedure can be applied to sample street-level photos for zone segmentation.

We report the quantitative results of our method w.r.t varying brush strokes in Table 7. Several qualitative results on the various datasets are presented in Figures 8, 9, 10, and 11. In general, we observe a decreasing trend over accuracy and mIOU as the proportion of side information available decreases. This supports our hypothesis that side information is a key signal for improving semantic segmentation accuracy. We also observe a trade off between human effort and segmentation accuracy, with a greater number of annotations leading to increased accuracy. For example, on the zone segmentation dataset [7], improvement over the baseline Deeplab-ResNet is achieved with as little as 20% of the original number of geotagged photos. This suggests that our proposed method can provide significant performance gains even with minimal human effort.

We are also able to observe noticeable improvement of segmentation quality as the amount of side information available increases. For example, on the zone segmentation dataset shown in Figure 8, 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, segmentation quality is steadily improved. Similar trends are also found in NYC and BOS.

On the BACH dataset (see Figure 9), 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 the A10 slide are more consistently identified with the inclusion of brush strokes.

On the Zurich Summer Dataset, illustrated in Figures 10 and 11, the improvement is not as visually obvious as compared with the zone segmentation 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 example, in zh5 (see Figure 10), side information helps to correctly identify the tiny Bare Soil area. Similarly, in zh8 (see Figure 10), 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.

C. Computational Analysis

An additional advantage of our method is its computational efficiency, which comes into play with high density annotations. In particular, the BACH dataset consists of very high resolution whole slide images, which is common in many medical datasets. Coupled with dense brush stroke annotations, this results in significant bottlenecks for prior works, e.g., the Unified model [25].

In order to evaluate the computational complexity quantitatively, we benchmark the inference speeds of the Deeplab-ResNet model [5], Unified model, and SideInfNet. As the HO-MRF model requires an additional post-processing step in the form of global normalization, we do
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 example, with a batch size of 64, SideInfNet averages 0.057s per patch on the BACH dataset.

The results are summarized in Table 8. We observe that on datasets with smaller resolution images and sparser side information (e.g., the Zurich Summer dataset), the Unified model is able to outperform SideInfNet in terms of inference speed. 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 greatly outperforms the Unified model. Specifically, 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.

D. Additional Qualitative Evaluations

D.1. Qualitative Results on BACH dataset

Several qualitative results of our method on the BACH dataset are as shown in Figure 12. From the results presented, we observe that, compared to other models, SideInfNet generally provides the highest quality results. The segmentation masks produced by SideInfNet are less noisy and sparse. In addition, compared to 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 [10] or HO-MRFs [7]. 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 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.

D.2. Qualitative Results on Zurich Summer Dataset

Our qualitative results on the Zurich Summer dataset are presented in Figure 13. 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 example, as shown in the docks at the bottom right area in the zh11 image, SideInfNet can well distinguish between Background (white) and Building (gray). Docks are a relatively rare environmental feature, which make them difficult to correctly classify. The Unified model 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 Figure 13), which challenge other models. The segmentation results of SideInfNet on Railway class in the zh8 image are also more coherent compared to prior works.
|          | 0%  | 20% | 40% | 60% | 80% |
|----------|-----|-----|-----|-----|-----|
| SFO Photos | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) |
| SFO Results | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) |
| NYC Photos | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) |
| NYC Results | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) |
| BOS Photos | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) |
| BOS Results | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) |

Figure 8. Results on zone segmentation [7] with varying brush strokes. Best viewed in color.
| Brush Strokes | A05 | A10 |
|--------------|-----|-----|
| 0%           | ![Image](image1) | ![Image](image2) |
| 20%          | ![Image](image3) | ![Image](image4) |
| 40%          | ![Image](image5) | ![Image](image6) |
| 60%          | ![Image](image7) | ![Image](image8) |
| 80%          | ![Image](image9) | ![Image](image10) |

- **Benign**: Red
- **Invasive carcinoma**: Blue
- **In situ carcinoma**: Green

Figure 9. Results on BACH [2] with varying brush strokes. Best viewed in color.
|     | 0%     | 20%     | 40%     | 60%     | 80%     |
|-----|--------|---------|---------|---------|---------|
| zh5 | ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) | ![Image](image4.png) | ![Image](image5.png) |
| zh7 | ![Image](image6.png) | ![Image](image7.png) | ![Image](image8.png) | ![Image](image9.png) | ![Image](image10.png) |
| zh8 | ![Image](image11.png) | ![Image](image12.png) | ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) |

Figure 10. Results on the Zurich Summer Dataset [24] with varying brush strokes. Best viewed in color.
Figure 11. Results on the Zurich Summer Dataset [24] with varying brush strokes. Best viewed in color.
Figure 12. Qualitative results on the BACH dataset [2]. Best viewed in color.
Figure 13. Qualitative results on the Zurich Summer dataset [24]. Best viewed in color.