

Semi-supervised Instance Segmentation with a Learned Shape Prior ^{*}

Long Chen¹[0000-0002-5280-4727], Weiwen Zhang¹, Yuli Wu¹, Martin Strauch¹[0000-0001-6754-211X], and Dorit Merhof¹[0000-0002-1672-2185]

Institute of Imaging & Computer Vision, RWTH Aachen University, Germany
 {long.chen, martin.strauch, dorit.merhof}@lfb.rwth-aachen.de
<https://www.lfb.rwth-aachen.de/>

Abstract. To date, most instance segmentation approaches are based on supervised learning that requires a considerable amount of annotated object contours as training ground truth. Here, we propose a framework that searches for the target object based on a shape prior. The shape prior model is learned with a variational autoencoder that requires only a very limited amount of training data: In our experiments, a few dozens of object shape patches from the target dataset, as well as purely synthetic shapes, were sufficient to achieve results en par with supervised methods with full access to training data on two out of three cell segmentation datasets. Our method with a synthetic shape prior was superior to pre-trained supervised models with access to limited domain-specific training data on all three datasets. Since the learning of prior models requires shape patches, whether real or synthetic data, we call this framework semi-supervised learning. The code is available to the public¹.

Keywords: Semi-supervised · Instance segmentation · Shape prior · Variational autoencoder · Edge loss

1 Introduction

Instance segmentation, where many instances of an object have to be segmented in one image, is the basis of several practically relevant applications of computer vision, such as cell tracking [1]. Many approaches [2,3,4] have been proposed for instance segmentation, the majority of which are based on supervised learning. The practical applicability of these methods is often limited by the lack of a large training dataset with manually outlined objects. Here, we introduce an instance segmentation approach that only relies on a shape prior which can be learned from a considerably smaller number of training samples or even synthetic data.

The shape is one of the most informative cues in object segmentation and detection tasks. Anatomically constrained neural networks (ACNNs) [5] improve segmentation results by including a shape prior for model regularization. For

^{*} This work was supported by the Deutsche Forschungsgemeinschaft (Research Training Group 2416 MultiSenses-MultiScales).

¹ https://github.com/loooooongChen/shape_prior_seg

segmentation refinement, a shape prior has been used by [6] as a separate post-processing step. Segmentations generated by the shape prior model are reconstructed to the original MRI images through several convolutional layers in [7]. By minimizing the reconstruction error, the segmentation model can be trained in an unsupervised fashion. All these works report promising results, but are limited to cases where object position and extent are roughly the same in all images, such as for the cardiac images in [5], the lung X-ray images in [6] and the brain MRI scans in [7]. To our knowledge, this is the first work considering instance segmentation based on a shape prior, i.e. we detect and segment multiple, scattered object instances. Similar to [8], we use the spatial transformer [9] to localize objects. The main advantage of using the spatial transformer lies in its differentiability, making the whole framework end-to-end trainable.

The main contributions of this work are: We propose (1) an semi-supervised instance segmentation approach that searches for target objects based a shape prior, and (2) a novel loss computing the difference between two gradient maps. This framework provides a way to achieve instance segmentation with a small amount of manual annotations, or by utilizing unpaired annotations (where the correspondence between annotations and images is unknown). We compared our approach to the state-of-the-art supervised method, Mask R-CNN [2], in different training scenarios. On three experimental datasets, our approach is proved to be en par with a Mask R-CNN with full access to training data, while it outperforms a pre-trained Mask R-CNN with limited access to domain-specific training data.

2 Approach

As shown in Figure 1, our framework consists of three main parts: 1) the localization network, 2) the spatial transformer [9], and 3) the patch segmentation network. Based on the localization prediction, the spatial transformer crops local patches and feeds them to the patch segmentation network. The gradient maps of segmented patches are then stitched together. The entire model is trained by minimizing the reconstruction error of the gradient map.

During training, the model learns to predict the object position and to find the correspondence between the image patch and the segmentation. The shape prior model (gray part in Fig. 1; fixed during training) is guaranteed to output a plausible shape, but the correspondence has to be learned by the model itself.

2.1 Localization network

The localization network consists of 8 convolutional layers and 4 max pooling layers after every 2 convolutional layers. Given an image of size (H_{img}, W_{img}) , the localization network will spatially divide the image into an $(H_{img}/S_{cell}, W_{img}/S_{cell})$ grid of cells, where S_{cell} is the cell size and also the downsampling rate. Since 4 pooling layers with stride 2 are used, we have $S_{cell} = 16$.

Each cell is responsible to predict the presence of an object $L_{presence} \in [0, 1]$, its range described by the bounding box size (H_{obj}, W_{obj}) and the offset with respect to the cell center (O_x, O_y) (Figure 2(a)), with the implementation:

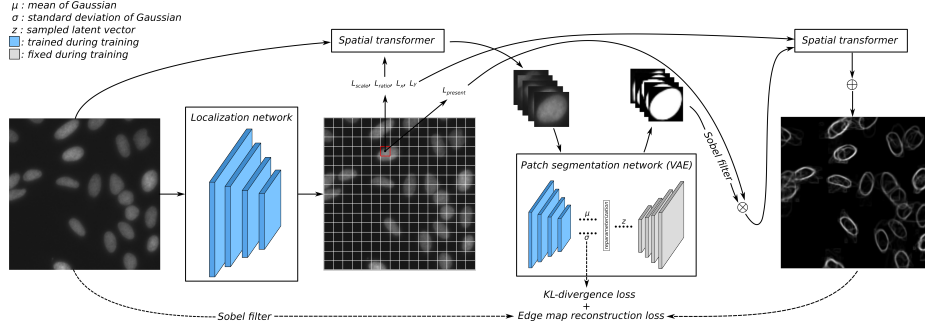


Fig. 1: Architecture of our framework: the localization network predicts the object position and a presence score, based on which object patches are cropped by a spatial transformer. A variational autoencoder with the decoder part fixed (shape prior) is responsible for the patch segmentation. At last, the gradient maps of segmented patches are stitched together. The model is trained by minimizing the reconstruction loss of the gradient map with the KL-divergence loss as regularization.

$$\begin{aligned}
 L_{\text{presence}} &= \text{sigmoid}(f_{\text{presence}}) \\
 L_{\text{scale}} &= \text{sigmoid}(f_{\text{scale}}) \cdot (S_{\text{max}} - S_{\text{min}}) + S_{\text{min}} \\
 L_{\text{ratio}} &= \exp(\tanh(f_{\text{ratio}}) \cdot \log(R_{\text{max}})) \\
 (L_x, L_y) &= (0.5 \cdot \tanh(f_x), 0.5 \cdot \tanh(f_y))
 \end{aligned}$$

where $f_{[\cdot]}$ is the corresponding input feature map. $\text{sigmoid}(\cdot)$ and $\tanh(\cdot)$ denote the sigmoid and tanh activation function. S_{min} , S_{max} and R_{max} are hyperparameters, which are the minimal scale, the maximal scale and the maximal aspect ratio, respectively. The position is parameterized according to:

$$\begin{aligned}
 (H_{\text{obj}}, W_{\text{obj}}) &= (L_{\text{scale}} \cdot S_{\text{cell}} / \sqrt{L_{\text{ratio}}}, L_{\text{scale}} \cdot S_{\text{cell}} \cdot \sqrt{L_{\text{ratio}}}) \\
 (O_x, O_y) &= (L_x \cdot S_{\text{cell}}, L_y \cdot S_{\text{cell}})
 \end{aligned}$$

It is worth mentioning that the maximal offset is $0.5 \cdot S_{\text{cell}}$, which means that an object will be detected by the cell in which its center lies.

2.2 Patch crop and stitch

Given the location parameters obtained from the localization network, we use a spatial transformer to crop local patches. The spatial transformer implements the crop by sampling transformed grid points, which is differentiable, enabling end-to-end training. The patch crop of the i -th cell can be described by transform:

$$T_{\text{crop}}^i = \begin{bmatrix} W_{\text{img}}/W_{\text{obj}}^i & 0 & W_{\text{img}} \cdot (X_{\text{cell}}^i + O_y^i)/W_{\text{obj}}^i \\ 0 & H_{\text{img}}/H_{\text{obj}}^i & H_{\text{img}} \cdot (Y_{\text{cell}}^i + O_x^i)/H_{\text{obj}}^i \\ 0 & 0 & 1 \end{bmatrix}$$

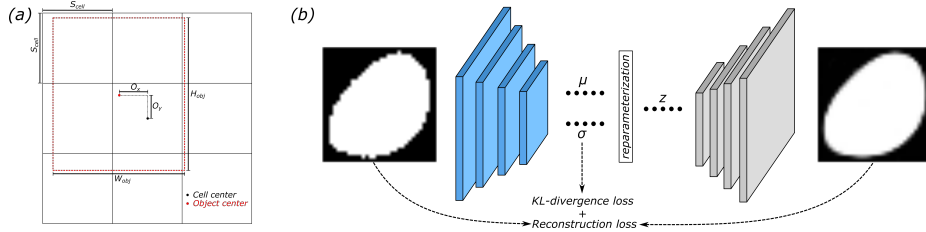


Fig. 2: (a) Demonstration of parameters of a bounding box. (b) Architecture of the patch segmentation network, which is firstly trained with shape patches. During the detector training, the decoder part is fixed and plays the role of shape prior.

where (X_{cell}^i, Y_{cell}^i) is the cell center. (O_x^i, O_y^i) and (H_{obj}^i, W_{obj}^i) are the predicted offset and size of the object. All cropped patches will be rescaled to size $S_{patch} \times S_{patch}$ ($S_{patch} = 32$ in this work) and segmented by the patch segmentation network, as described in Section 2.3. After that, the gradient map of segmented objects will be stitched together by adding up back transformed patches through:

$$T_{stitch}^i = \begin{bmatrix} W_{obj}^i/S_{patch} & 0 & X_{cell}^i + O_x^i \\ 0 & H_{obj}^i/S_{patch} & Y_{cell}^i + O_y^i \\ 0 & 0 & 1 \end{bmatrix}$$

The gradient map is computed by applying the x- and y-directional Sobel filter to the image and taking the square root of the summed square. The gradient map is normalized to range 0 to 1. In this work, we use an input size of 256×256 for all experiments. Considering $S_{cell} = 16$, 256 patches are cropped in total.

2.3 Shape prior and patch segmentation network

Similar to [5,6,7], we employ a variational autoencoder (VAE) as our shape model. As shown in Figure 2(b), the model is trained to reconstruct plausible patch segmentation masks with the KL-divergence loss as regularization. Compared to a standard autoencoder, a VAE learns a more continuous latent space, which is expected to generate plausible new shapes that do not appear in training data.

In this work, the VAE is trained with 32×32 patches. The encoder and decoder consist of 6 convolutional layers and 3 pooling/upsampling layers, respectively. Based on our experiments, model training requires only a small amount of data, especially when the shape variation is small. We train the shape prior with either annotations from a single image or synthetic data (Section 3).

After training, the decoder part will be used as the shape prior in the detector (Figure 1). Its parameters will be fixed during the detector training. The encoder will be reinitialized and trained together with the localization network.

2.4 Training

The model is trained end-to-end by minimizing the gradient map reconstruction error with the KL-divergence loss as regularization. In initial experiments,

we found the mean absolute/squared error (MAE/MSE) to be very unstable during training: The shape prior model tends to generate distorted shapes or degenerates into empty output. Thus, we propose the following novel loss:

$$L_{edge} = 1 - \frac{\frac{1}{N} \sum_i \min^2(G_{image}^i, G_{reconstruction}^i)}{\frac{1}{N} \sum_i G_{reconstruction}^i + \alpha} \quad (1)$$

where G_{image} and $G_{reconstruction}$ indicate the gradient map of the image and the reconstructed gradient map. N is the number of pixels. The $\min()$ operation are conducted pixelwise. The parameter α prevents the model from pushing $G_{reconstruction}$ to zero and is set to 0.01 empirically.

Instead of optimizing the value of each pixel, as MSE and MAE, this loss maximizes the proportion of the reconstructed gradient map under the image gradient map. In addition, the square operator in the numerator is proved to be crucial for stable training in our experiments. Our interpretation is that the square operator modulates the back-propagated gradient with the reconstructed gradient map, giving more emphasis to positions around the edge.

2.5 Pre- and post-processing

To reduce the influence of extreme values on the loss, we equalized the image and the gradient map by clipping and stretching. For all datasets, we truncated the gradient map at 0.8 times the maximum and normalized the value to the range 0 to 1. In addition, we also performed image equalization for the Fluo-N2DH-SIM+ dataset due to the bright spots inside the cell (Figure 3). The clip value was set to 1.2 times the image mean.

As post-processing, we first filtered out predictions with $L_{presence}$ smaller than 0.1. Non-max suppression is then performed to eliminate duplicate predictions: An instance mask is compared with another mask, when the overlapping area is larger than $p_{non_max} = 0.1$ with respect to its own area. A mask is only retained if its score is the highest in all comparisons.

3 Experiments and results

3.1 Datasets and experiments

We evaluate our approach on three datasets: the BBBC006 dataset² and two datasets Fluo-N2DH-SIM+ and PhC-C2DL-PSC from the cell tracking challenge [1]. In the following, we use BBBC, FLUO and PHC as abbreviations. The BBBC dataset contains 768 microscopic images of human U2OS cells, while the FLUO (HL60 cells with Hoechst staining) and PHC (pancreatic stem cells on a polystyrene substrate) datasets are smaller with 215 and 202 annotated images.

For comparison, we also report the performance of the supervised method Mask R-CNN. The following experiments are performed:

² <https://data.broadinstitute.org/bbbc>

Ours-annotation: We first evaluate our approach with the shape prior learned from manual annotations. We only took segmentation patches from one image. Specifically, 67, 8 and 138 object patch masks were used for the BBBC, FLUO and PHC shape model training. To model small shape changes and object rotation, we performed rotation (in steps of 30 degrees) and elastic deformation [11] to augment the training set. The scale range and maximal aspect ratio was set to 2-3/3, 1-2/1.5 and 1-2/3, respectively.

Ours-synthetic: Since the objects are approximately circular, especially for the BBBC and FLUO datasets, we could train the shape prior model with synthetic data consisting of elastically deformed ellipses [11] with random angle and major-minor axis ratio. The maximal major-minor axis ratio was 2, 1.5 and 3 for the BBBC, FLUO and PHC dataset, respectively.

MRCNN-scratch-one/full: We trained a Mask R-CNN from scratch using ResNet-50 backbone. The anchor box scale, aspect ratio and non-maximum suppression (NMS) threshold were set to values equivalent to those used in our approach. Since the Ours-annotation scenario can be considered as one image training, we also trained a Mask R-CNN with one image for comparison.

MRCNN-finetune-one/full: Since the dataset in our experiments is small, especially for FLUO and PHC, we pretrained the Mask R-CNN on the MS COCO dataset³. Afterwards, we finetuned the model, with only the head layers trainable, on the actual target dataset.

For the BBBC and PHC dataset, we cropped images to 256×256 and 128×128 for training and test. All images were resized to 256×256 for the network input. For the scenarios using one training image (Ours-annotation, MRCNN-scratch-one, MRCNN-finetune-one), the images a01_s1, 02/t000, 02/t150 were used for BBBC, FLUO and PHC, respectively. MRCNN-scratch-full and MRCNN-finetune-full used a01_s1-b24_s2, 02/t000-t149, 02/t150-t250 for training. Ours-synthetic requires no manual annotations. All remaining images were kept for testing.

3.2 Results and discussion

We report the *average precision*⁴ (AP) over a range of IoU (intersection over union) thresholds from 0.3 to 0.9 as the evaluation score (Table 1). Our approach, including the evaluation scenarios where the shape prior is learned from one image annotation and synthetic data, outperforms the Mask R-CNN trained or finetuned with one image, which shows the advantage of our approach in cases where few or no annotations are available. Furthermore, our approach achieves comparable results with the Mask R-CNN trained/finetuned with the full training set on the BBBC and FLUO dataset, while the performance gap is apparent for the PHC dataset.

While Mask R-CNN achieved the best mean AP (mAP) on the BBBC dataset, our approach outperformed Mask R-CNN on the FLUO dataset by a relatively large margin. The main reason is that the FLUO dataset is indeed a very small

³ <https://cocodataset.org/>

⁴ <https://www.kaggle.com/c/data-science-bowl-2018>

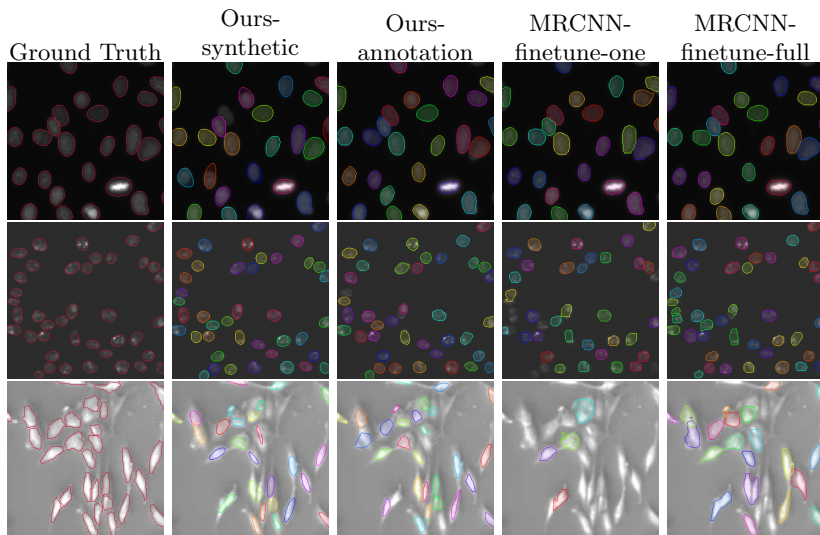


Fig. 3: Qualitative results: from top to bottom, the rows show the results on the BBBC006, Fluo-N2DH-SIM+ and PhC-C2DL-PSC datasets, respectively.

one for Mask R-CNN training, even with finetuning. This again illustrates the advantage of our method on small datasets.

On the PHC dataset, neither method performed particularly well. Both methods tended to detect nearby objects as one if there was no clearly visible edge between them. The average precision of our method in the low IoU range was close to or better than that of Mask R-CNN. Figure 3 shows that our method could detect most objects as well as the Mask R-CNN. However, our method has been designed to heavily rely on the edge clue, so that the segmentation will converge to strong edges. For the PHC dataset, the object boundaries do not generally correspond to the strongest edges. This explains why objects were undersegmented by our approach (Figure 3) and why the average precision decreased rapidly with increasing IoU (Table 1).

The performance improvement through training the shape prior with manually outlined shapes depends on the nature of the shape. On the FLUO dataset, annotated data and synthetic data shape priors performed almost equally well, while training with manual annotations was superior on the other two datasets, even though only a few dozen shapes were used.

4 Conclusion and outlook

We have proposed an instance segmentation framework which searches for target objects in images based on a shape prior model. In practice, this allows segmenting instances with a very limited amount of annotations, segmenting synthesizable shapes without any annotation, as well as reusing object annotations from other datasets.

Table 1: Average precision (AP) over different IoU for different datasets (the best two scores in bold). Experiments and abbreviations are introduced in Section 3.1.

Dataset	IoU	0.3	0.4	0.5	0.6	0.7	0.8	0.9	mAP
BBBC	Ours-annotation	.8345	.8260	.7977	.7632	.7083	.6100	.2660	.6865
	Ours-synthetic	.8171	.8012	.7641	.7170	.6525	.5247	.2042	.6401
	MRCNN-scratch-one	.6386	.5934	.5459	.4769	.3543	.1759	.0294	.4020
	MRCNN-scratch-full	.7901	.7851	.7708	.7473	.7128	.6296	.3374	.6817
	MRCNN-finetune-one	.7672	.7524	.7277	.7020	.6608	.5492	.1250	.6121
	MRCNN-finetune-full	.7997	.7949	.7851	.7720	.7521	.6923	.3485	.7064
FLUO	Ours-annotation	.9605	.9538	.9312	.8999	.8228	.6777	.1332	.7685
	Ours-synthetic	.9600	.9497	.9336	.8986	.8324	.6768	.1378	.7698
	MRCNN-scratch-one	.0458	.0324	.0156	.0018	.0000	.0000	.0000	.0014
	MRCNN-scratch-full	.9333	.9144	.8703	.7605	.5765	.2556	.01073	.6173
	MRCNN-finetune-one	.8224	.8133	.7905	.7389	.5909	.2404	.0049	.5716
	MRCNN-finetune-full	.9361	.9252	.8955	.8467	.7265	.4115	.0197	.6802
PHC	Ours-annotation	.6840	.6034	.4035	.1468	.0233	.0028	.0000	.2662
	Ours-synthetic	.6471	.5611	.3605	.1326	.0219	.0027	.0000	.2466
	MRCNN-scratch-one	.1124	.0991	.0847	.0668	.0353	.0049	.0000	.0576
	MRCNN-scratch-full	.6332	.6001	.5226	.4467	.2981	.1079	.0023	.3730
	MRCNN-finetune-one	.1647	.1602	.1460	.1146	.0633	.0108	.0000	.0942
	MRCNN-finetune-full	.6551	.6380	.5855	.5014	.3425	.1144	.0007	.4053

The main limitation of our approach lies in the dependency on the edge cues. Images should have a relatively clear background, which is, however, the case for many biomedical datasets⁴. Future work will focus on including area-based information, which will make our approach applicable to further datasets, e.g. in cases where edges and object boundaries do not always coincide.

References

1. Ulman, V., et al.: An Objective Comparison of Cell-tracking Algorithms. *Nature Methods*, **14**, 1141-1152 (2017)
2. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 ICCV, 2980-2988
3. Schmidt, U., Weigert, M., Broaddus, C., Myers, E.W.: Cell Detection with Star-Convex Polygons. In: 2018 MICCAI, 26-273
4. Chen, L., Strauch, M., Merhof, D.: Instance Segmentation of Biomedical Images with an Object-Aware Embedding Learned with Local Constraints. In: 2019 MICCAI, 451-459
5. Oktay, O., et al.: Anatomically Constrained Neural Networks (ACNNs): Application to Cardiac Image Enhancement and Segmentation. *IEEE Transactions on Medical Imaging*, **37**(2), 384-395 (2018)
6. Larrazabal, A. J., Martinez, C., Ferrante, E.: Anatomical Priors for Image Segmentation via Post-processing with Denoising Autoencoders. In: 2019 MICCAI, 585-593
7. Dalca, A. V., Gutttag, J., Sabuncu, M. R.: Anatomical Priors in Convolutional Networks for Unsupervised Biomedical Segmentation. In: 2018 CVPR, 9290-9299

8. Crawford, E., Pineau, J.: Spatially Invariant Unsupervised Object Detection with Convolutional Neural Networks. In: 2019 AAAI, 3412-3420
9. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial Transformer Networks. In: 2015 NIPS, 2017-2025
10. Kingma, D. P., Welling, M.: Auto-Encoding Variational Bayes. In: 2014 ICLR
11. Simard, P. Y., Steinkraus, D., Platt, J. C.: Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp. 958. IEEE (2003)