

# Instance Segmentation of Biological Images Using Harmonic Embeddings

Victor Kulikov

PicsArt Inc.\*

Moscow, Russian Federation

victor.kulikov@picsart.com

Victor Lempitsky

Samsung AI Center Moscow

Skolkovo Institute of Science and Technology

lempitsky@skoltech.ru

## Abstract

*We present a new instance segmentation approach tailored to biological images, where instances may correspond to individual cells, organisms or plant parts. Unlike instance segmentation for user photographs or road scenes, in biological data object instances may be particularly densely packed, the appearance variation may be particularly low, the processing power may be restricted, while, on the other hand, the variability of sizes of individual instances may be limited. The proposed approach successfully addresses these peculiarities.*

*Our approach describes each object instance using an expectation of a limited number of sine waves with frequencies and phases adjusted to particular object sizes and densities. At train time, a fully-convolutional network is learned to predict the object embeddings at each pixel using a simple pixelwise regression loss, while at test time the instances are recovered using clustering in the embedding space. In the experiments, we show that our approach outperforms previous embedding-based instance segmentation approaches on a number of biological datasets, achieving state-of-the-art on a popular CVPPP benchmark. This excellent performance is combined with computational efficiency that is needed for deployment to domain specialists.*

*The source code of the approach is available at <https://github.com/kulikovv/harmonic>.*

## 1. Introduction

Instance segmentation (object separation) in biological images is often a key step in their analysis. Many biological image modalities (e.g., microscopy images of cell cultures) are characterized by excessive numbers of instances. Other modalities (e.g., worm assays) are characterized by tight and complex overlaps and occlusions. On the other hand, in most situations the scale variations of objects of interest in biomedical data are less drastic than in natural

photographs due to the lack of strong perspective effects. In this work, we propose a new approach designed for biological image instance segmentation that can address the challenges (number of instances, overlaps) while exploiting the simplifying properties (limited scale variation).

Our approach continues the series of studies [7, 21] that perform instance segmentation by learning deep embeddings and using clustering in the embedding space to recover the instances at test time. While this approach is appealing in its simplicity, learning good embeddings for object instances with a fully-convolutional network is challenging, especially for biological data, where individual instances may have an almost indistinguishable appearance.

In order to utilize the specific nature of biomedical images, we depart from the end-to-end philosophy of the previous embedding-based instance segmentation works and split the learning process into two stages. At the first stage, we seek a small set of harmonic functions that can be used to separate objects in the training dataset. The search is implemented as an optimization process that tunes the frequencies and phases of the harmonics to a specific range of scales and object densities in the data. The selected set of harmonics then guides the second stage of the learning process as well as the inference process at test-time.

At the second learning stage, we assign each ground truth object instance its *harmonic embedding* based on the expectation of the learned set of functions. We then learn a deep fully-convolutional network to predict resulting embeddings at each pixel. We show that learning with a simple pixelwise regression loss is feasible as long as the information about harmonic functions is provided to convolutional layers in the network (which we achieve by augmenting the input to these convolutional layers). The learned networks generalize well to new images, and tend to predict pixel embeddings that can be easily clustered into object instances.

In the experiments, we compare our approach to direct embedding-based instance segmentation [7] as well as to other state-of-the-art methods. Four biomedical datasets corresponding to plant phenotyping, bacterial and human cell culture microscopy, as well as *C.elegans* assays, are

\*Work done while at Skolkovo Institute of Science and Technology.

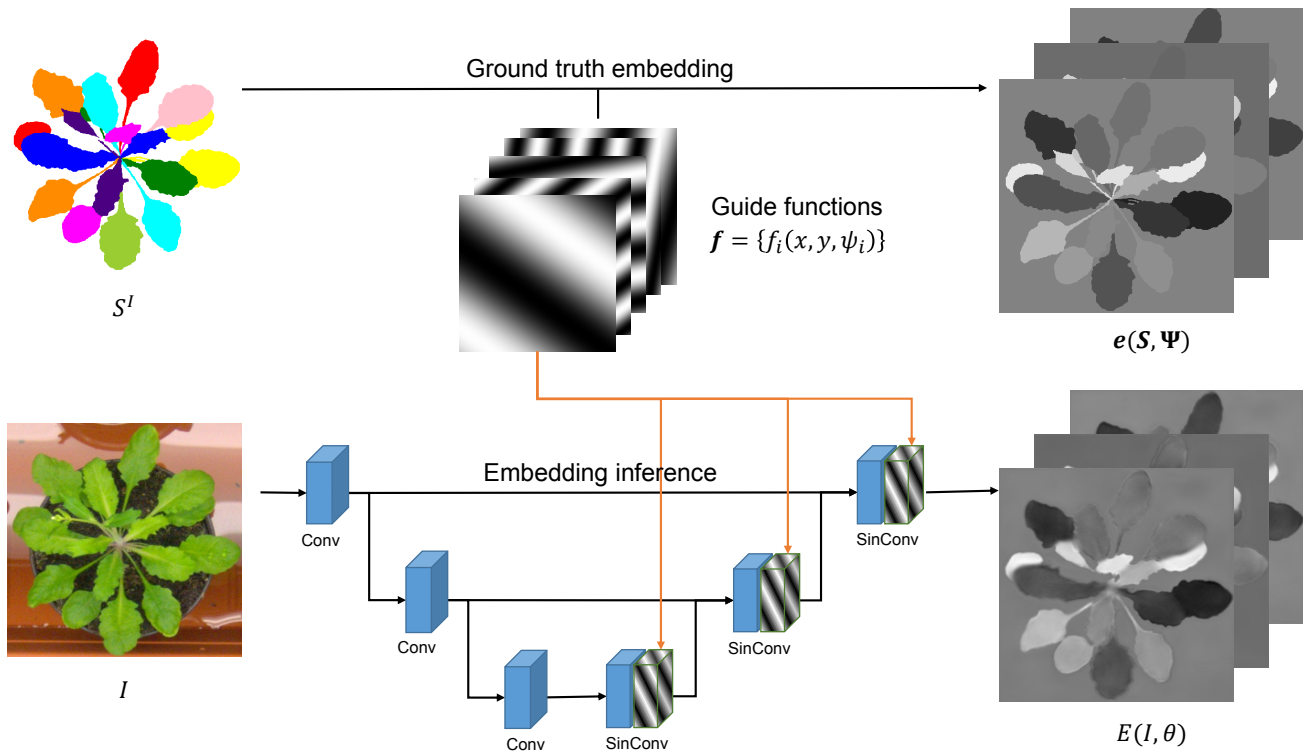


Figure 1. The **harmonic instance segmentation** framework. At train-time, we embed each pixel of the ground truth image  $S^I$  as the mean of predefined guide functions  $\mathbf{f}$  over instance pixels it belongs to, resulting in embeddings  $e(S, \Psi)$ . We then train the neural network  $E$  to reproduce the ground truth embedding given the inputted image  $I$ . To simplify learning, guide functions  $\mathbf{f}$  are inputted into intermediate representations of the network using *SinConv* layers. The learning process uses a simple pixelwise L1-Loss between ground truth embedding  $e(S, \Psi)$  and the neural network prediction  $E(I, \theta)$  as a learning objective. At test time instances are retrieved from the predicted embedding  $E(I, \theta)$  using mean shift clustering.

considered. We observe considerable improvement in performance brought by our approach.

## 2. Related Work

**Proposal-based instance segmentation** methods [5, 11, 3, 5, 6, 17, 12] combine object detection with object mask estimation and currently represent state-of-the-art in instance segmentation non-biological benchmarks. The necessity to perform object detection followed by non-maximum suppression makes learning and tuning of methods from this group complex, especially in the presence of tight object overlaps when non-maximum suppression becomes fragile.

Another group of approaches to instance segmentation is based on **recurrent neural networks** (RNNs) that generate instances sequentially. Thus, Romera et al. [24] trains a network for end-to-end instance segmentation and counting using LSTM networks [14]. Ren et al. [23] proposed a combination of a recurrent network with bounding box proposals. RNN-based frameworks show excellent performance on small datasets; they achieve state-of-the-art results on

the CVPPP plant phenotyping dataset. The major problem of recurrent methods is the vanishing gradient effect that becomes particularly acute when the number of instances is large.

Our method falls into the category of *proposal-free* approaches to instance segmentation based on **instance embedding**. In this case, neural networks are used to embed pixels of an image into a hidden multidimensional space, where embeddings for pixels belonging to the same instance should be close, while embeddings for pixels of different objects should be separated. A clustering algorithm may then be applied to separate instances. To achieve this, the previous approach [9] penalizes pairs of pixels using a logistic distance function in the embedding space. The embedding is learned using log-loss function and requires to weight pixel pairs in order to mitigate the size imbalance issue. This method also predicts a seedness score for each pixel, which correlates with the centeredness. They use this score to pick objects from the embedding. Kong et al. [15] use differentiable Gaussian Blurring Mean-Shift for the recurrent grouping of embeddings. Deep Coloring [16] proposes a reduction of instance segmentation to semantic seg-

mentation, whereas class labels are reused for non-adjacent objects. The instances are then retrieved using connected component analysis.

In their study most related to ours, De Brabandere et al. [7] use a non-pairwise discriminative loss function composed of two parts: one pushing the embedding mean of different objects further apart, while the other one pulling the embeddings of the same object pixels closer to its mean. Instances are retrieved using the mean-shift algorithm. The approach [21] uses metric learning together with an explicit assignment of the center of mass as the target embedding. Our approach follows the general paradigm of [7, 21] but suggests a special kind of embeddings detailed below. The use of new embeddings results in explicit assignment of embeddings to each pixel in the training image, thus simplifying the learning process.

### 3. Harmonic Instance Embedding

We now discuss our approach in detail. Existing instance embedding methods [7, 16, 9] do not prespecify target embeddings for pixels in the training set. Instead, they rely on the learning process itself to define these embeddings. In contrast, our goal is to define “good” embeddings to pixels a priori. “Goodness” here means amenability for clustering as well as learnability by a convolutional architecture.

Let  $\mathbf{f} = \{f_1(x, y; \psi_1), f_2(x, y; \psi_2), \dots, f_N(x, y; \psi_N)\}$  be a family of  $N$  real-valued functions  $f_i(x, y; \psi_i)$  in the image domain, where  $(x, y)$  corresponds to the coordinates of the argument, and  $\psi_i$  is a set of learnable parameters defining the shape of the function (e.g., the frequency vector and the phase of a sine function). As our approach is built in many ways around this family of functions, we call the function family  $\mathbf{f}$  the *guide functions*.

Let  $\mathbf{S}$  be an arbitrary set of pixels (e.g., an object instance in the ground truth annotation of a training image). We denote the expectation of  $f_i$  over  $S$  with  $e_i(\mathbf{S}; \psi_i)$ :

$$e_i(\mathbf{S}; \psi_i) = \frac{1}{|\mathbf{S}|} \sum_{(x,y) \in \mathbf{S}} f_i(x, y; \psi_i). \quad (1)$$

If  $\Psi = \{\psi_1, \dots, \psi_N\}$  denotes the joint vector of parameters of all  $N$  functions, then the *guided embedding* of an object  $\mathbf{S}$  determined by  $\Psi$  is defined as the following  $N$ -dimensional vector:

$$\mathbf{e}(\mathbf{S}; \Psi) = \{e_1(\mathbf{S}; \psi_1), e_2(\mathbf{S}; \psi_2), \dots, e_N(\mathbf{S}; \psi_N)\}. \quad (2)$$

To sum up, the guided embedding maps each object  $S$  to the vector of the guide functions’ expectations over this object.

#### 3.1. Picking good guide functions

Given a new dataset representing a new type of instance segmentation problem, our goal is to find a good set of guide

functions (1) so that different objects have well-separated guided embeddings.

To do that, we first restrict  $f_i$  to a certain functional family parameterized by the parameters  $\psi_i$ . As discussed above, in many biomedical datasets, there is a certain (imperfect) regularity in the location of objects. For example, monolayer cell cultures organize themselves in a texture composed of elements having approximately the same size and being adjacent to each other. Such loosely-regular, semi-periodic structure calls for the use of harmonic functions as guides:

$$f_i(x, y; \psi_i) = \sin\left(\frac{\psi_i[1]}{W}x + \frac{\psi_i[2]}{H}y + \psi_i[3]\right), \quad (3)$$

where  $W$  and  $H$  are image width and height, respectively;  $\psi_i[1]$  and  $\psi_i[2]$  are frequency parameters; and  $\psi_i[3]$  is the phase parameter.

Assume now that a set of training images  $\mathcal{I}$  is given. We can then estimate the quality of guided embeddings by looking at pairs  $\mathbf{S}$  and  $\mathbf{S}'$  of objects belonging to the same image (e.g., two different cells from the same image) and finding out how frequently they have very close embeddings. Ideally, we want to avoid such collisions in the embedding space (at the very least, we want to avoid them in the training set). The following loss is therefore considered:

$$\ell(\Psi) = \sum_{I \in \mathcal{I}} \frac{1}{|\mathcal{P}_I|} \sum_{(S, S') \in \mathcal{P}_I} \max(0, \epsilon - \|\mathbf{e}(S; \Psi) - \mathbf{e}(S'; \Psi)\|_1), \quad (4)$$

where  $\|\cdot\|_1$  is the  $L_1$  distance;  $\epsilon$  is the margin meta-parameter; and  $\mathcal{P}_I$  denotes the set of all pairs of objects from the training image  $I$ . Each individual term in (4) is a hinge loss term that is non-zero if the guided embeddings of a certain object pair are too close (closer than  $\epsilon$ ).

To find good guide functions, we minimize the loss (4) on the training set. We perform stochastic gradient descent over the training set by drawing minibatches of random pairs of objects from random images and updating  $\Psi$  to minimize (4) for the pairs from the minibatch. In our implementation, we initialize frequency parameters  $\psi_i[1]$  and  $\psi_i[2]$  to uniformly distributed random numbers from the interval  $(0; 50)$ , while the phase parameters  $\psi_i[3]$  are initialized uniformly from  $[0; 2\pi)$ .

The outcome of the learning is a set of guide functions, such that pairs of objects from training images have their guided embeddings separated in the embedding space. For typical settings  $N = 12$  and  $\epsilon = 0.5$ , most pairs in the training set end up being isolated by more than the margin.

#### 3.2. Learning good embedding network

Assume now that the parameters of the guide functions have been optimized on the training set so that the parameters  $\Psi$  are now fixed. To derive the loss for the second

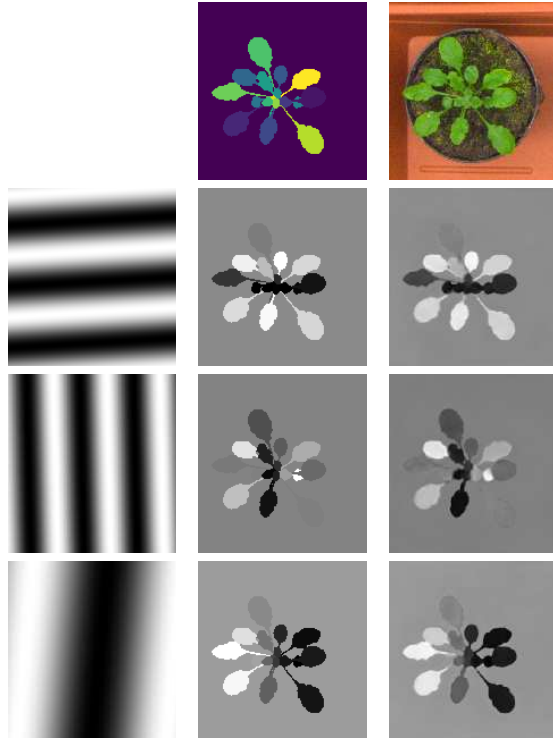


Figure 2. Left – guide functions (three of them shown here; 12 are used in our experiments). Middle – the GT masks (top) and the resulting guided embeddings. Note that each instance (leaf) is separated from other leaves in at least some of the guides. Right – the image (top) and the pixel embeddings as predicted from it by the convolutional network. Embeddings predicted by the network are very close to the guided embeddings, which greatly simplifies the clustering-based post-processing.

stage, we denote  $S^I(x, y)$  the set of all pixels of an object containing pixel  $(x, y)$ .

We then train a deep fully-convolutional embedding network  $E(I; \theta)$  with parameters  $\theta$  to map input images to sets of  $N$ -channel images, where each pixel is assigned an  $N$ -dimensional embedding. We minimize a simple pixelwise loss function that encourages the network to map each pixel to the guided embedding of the corresponding object:

$$\ell(\theta) = \sum_{I \in \mathcal{I}} \sum_{(x, y) \in \text{fg}(I)} \|E(I; \theta)[x, y] - \mathbf{e}(S^I(x, y); \Psi)\|_1. \quad (5)$$

Here,  $\text{fg}(I)$  denotes the set of foreground pixels of image  $I$  and  $E(I; \theta)[x, y]$  denotes the output of the network  $E$  at the spatial position  $(x, y)$  (if the foreground/background segmentation is not available, then the summation is taken over the full image).

We have found that standard fully-convolutional architectures (e.g., U-Net [25]) perform very well (Figure 2). Such architectures achieve low train and test set losses (5)

provided one important modification to convolutional layers is made. When modifying a convolutional layer  $L$ , we augment its input with an extra set of maps holding the guide functions values. Specifically, the extra maps contain the values  $\{f_1(x \cdot \Delta_L, y \cdot \Delta_L; \psi_1), \dots, f_N(x \cdot \Delta_L, y \cdot \Delta_L; \psi_N)\}$  at each spatial position  $(x, y)$ . Here,  $\Delta_L$  is the downsampling factor of the layer  $L$  (compared to the input/output resolution). The use of downsampling factor is needed to make sure that the augmenting maps in different layers are spatially aligned with the output.

Note that our augmentation idea generalizes the recently suggested CoordConv layer [19] that augmented the input of convolutional layers with  $\{f_1(x, y) = x, f_2(x, y) = y\}$ . By analogy and since the guiding functions in our implementation are harmonic, we call the new operation the *SinConv* layer (Figure 3).

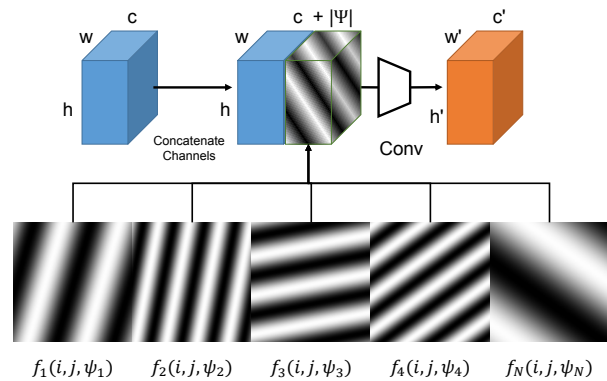


Figure 3. The SinConv layer maps a representation block with shape  $w \times h \times c$  to a new representation block of shape  $w' \times h' \times c'$  by concatenating guide functions and then performs convolution. The use of SinConv blocks greatly simplifies the task of learning to regress harmonic embeddings.

As mentioned above, our embedding architectures follow the design principles of the state-of-the-art semantic segmentation frameworks [25, 31, 13] which are composed of encoder and decoder parts with skip-connections between them. We use SinConv layers in the upsampling part (“decoder”) only. This allows to use a pretrained “backbone” networks in the encoder part, though we have not used this option in most of our experiments.

### 3.3. Instance segmentation of test images

At test time, the application of the learned embedding network is straightforward. The network  $E(I; \theta)$  is applied to an input image. Our post-processing is then similar to the one suggested in [7]. We use the mean-shift clustering algorithm [4] to obtain instance masks from the embedding space (Figure 2, right column). The mean-shift bandwidth is set to the margin  $\epsilon$  used in the guide function selection objective (4), since both parameters have the meaning of the



desirable separation between the embeddings of different instances.

## 4. Experiments

We provide results of our method on three challenging biomedical datasets of bright-field microscopy images of *C.elegans*, *E.coli*, HeLa and the plant phenotyping dataset (CVPPP 2017 sequence A1). In each case, learning was done on a single NVidia Tesla V100 GPU using ADAM optimizer with learning rate  $1e-5$ . Our implementation uses PyTorch [1].

The architecture and data augmentation were the same for all datasets. In our experiments, we have used the U-Net [25] neural network and replaced the first convolution of each upscaling block with the SinConv layer. The network was trained from scratch. Due to a small number of training images in those datasets, we have added some data augmentation procedures, namely, cropping patches of size  $448 \times 448$ , scaling, and left-right flips. The number of embedding dimensions was set to 12 and the margin  $\epsilon$  was set to 0.5. With these settings the objective 4 is minimized to zero during guide function search. Note that the availability of parameters that work well for diverse datasets is very important for practitioners. The number of training epochs was set differently for different datasets due to their varying complexity.

We used Symmetric best Dice coefficient (SBD) and average precision (AP) as metrics. The SBD metric averages the intersection over union (IOU) between pairs of predicted and the ground truth labels yielding maximum IOU. The AP metrics integrates precision for different recall values.

We have used the study by De Brabandere et al. [7] as the main baseline and have reimplemented their approach using the same network architecture as ours (the variants with and without SinConv layers were tried). On the CVPPP dataset where the authors’ implementation results are known, the result of our reimplementaion is considerably better suggesting that it forms a strong baseline.

### 4.1. CVPPP dataset

The Computer Vision Problems in Plant Phenotyping (CVPPP) dataset [26](Figure 4) is one of the most popular instance segmentation benchmarks. The dataset consists of five sequences of different plants. We have used the most common sequence A1 that has the most significant number of baselines. The A1 sequence has 128 top-down view images, with a size  $530 \times 500$  pixels each, as a training set and an additional hidden test set with 33 images from the same sequence. The task of instance segmentation is challenging because of the high variety of leaf shapes and complex occlusion between leaves. The performance of competing algorithms is SBD metrics and the absolute difference in counting  $|DiC|$  (c.f. [27]).

<i>Method</i>	<i>DiC</i>	<i>SBD</i> (%)
IPK [22]	2.6	74.4
Nottingham [27]	3.8	68.3
MSU [27]	2.3	66.7
Wageningen [29]	2.2	71.1
PRIAn [10]	1.3	-
Recurrent IS [24]	1.1	56.8
Recurrent IS+CRF [24]	1.1	66.6
Recurrent with attention [23]	0.8	84.9
Discriminative loss [7]	1.0	84.2
Deep coloring [16]	2.0	80.4
Discriminative loss [7] (our implementation)		
Without SinConv	4.	88.0
With SinConv	4.	89.0
Ours without SinConv	5.	78.3
<b>Ours</b>	3.	<b>89.9</b>

Table 1. Quantitative results on the CVPPP dataset (methods with published descriptions as well as our method and baselines included). Our method performs best in terms of using the Symmetric Best Dice coefficient (*SBD*) compared to previously published approaches.

We have trained the neural network for 500 epochs to fit this embedding space. Table 1 shows our method currently being the state-of-the-art in the main SBD metric compared to all published methods at the moment of submission.

### 4.2. E. coli dataset

The *E.coli* dataset (Figure 5) is interesting because the number of organisms is large and they are crowded. The dataset contains 37  $1024 \times 1024$  bright-field images. The ground truth is derived using the watershed algorithm [2] from weak annotations in which every organism is annotated by a line segment. The train-test splitting is done randomly selecting 30 as train images, and 7 as the test set. The validation is done on the train subset, so we have not used test set for the selection of meta-parameters.

At test time, images were processed by non-overlapping crops of size  $256 \times 256$ . The SBD score was calculated for each crop independently and then averaged. The performance of our method is better compared to other methods previously evaluated on this dataset (Table 2). Unfortunately, we were not able to obtain reasonable results from the method [7], probably due to a drastic change in the organism number between different crops.

### 4.3. HeLa dataset

The HeLa cancer cells dataset<sup>1</sup> (Figure 6) is quite different from the other three datasets. Cells take a large part of each image and, being cancerous, are more irregular and

<sup>1</sup>Courtesy Dr. Gert van Cappellen Erasmus Medical Center of Rotterdam.

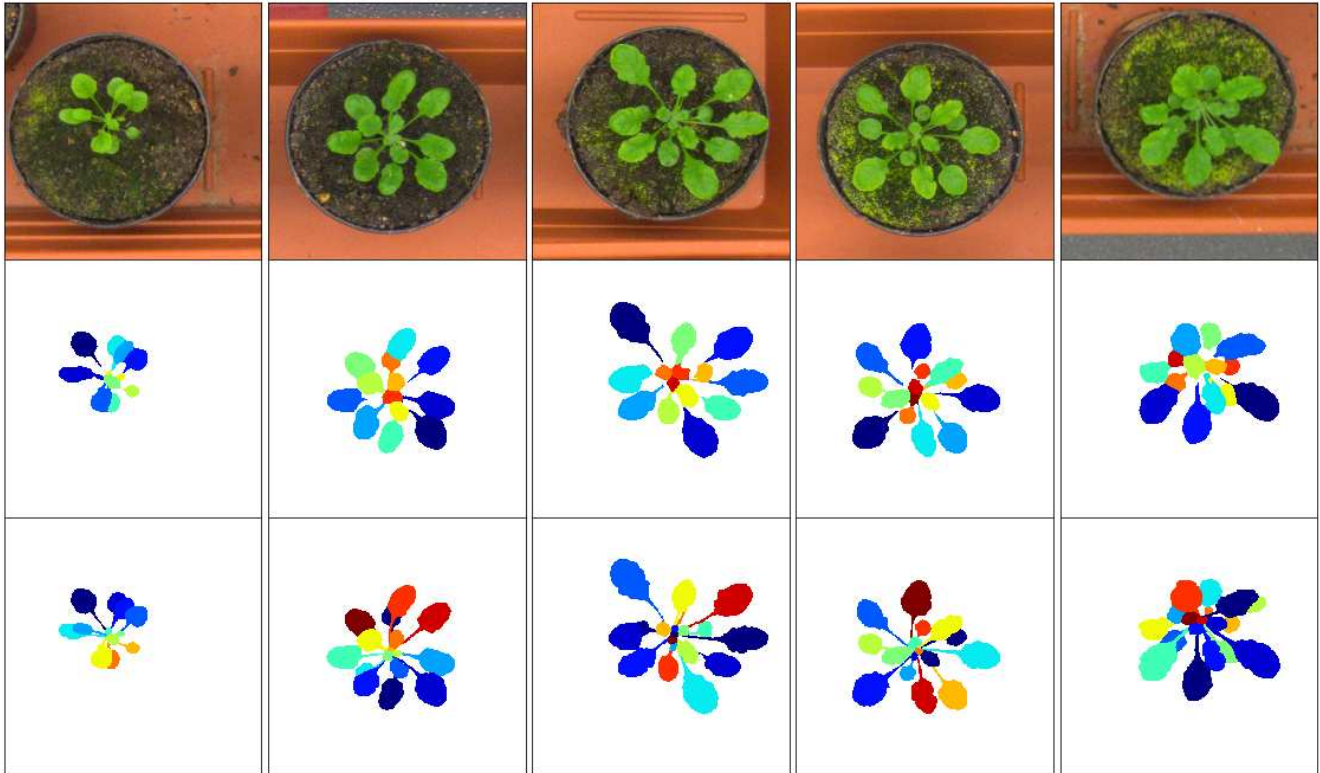


Figure 4. Sample results for the plant phenotyping dataset (CVPPP). The top row shows the source images; the second row shows our results; and the third row shows the ground truth (color display required).

Method	$DiC$	$SBD(\%)$
U-Net baseline [25]	-	59.3
Deep coloring [16]	2.2	61.9
<b>Ours</b>	<b>0.88</b>	<b>81.2</b>

Table 2. Results on the E.coli dataset. We follow the protocol from [26] to compare the results with those reported in [25] and [16].

form intricate patterns. In contrast to small and crowded pictures with E.coli, the number of cells is moderate, but they have a large area and more diverse sizes. The dataset contains 18 partially annotated single channel training images. Following best practices, we split the dataset into train and test parts (nine images each). The goal of this experiment is thus to show that our method can generalize well given very few training examples.

We trained the network for 8000 epochs. No information about the background was used during training. On this dataset, we achieve 78% SBD without foreground mask, and 86% SBD with foreground mask (imposed on top of the results), which slightly outperforms the semantic segmentation baseline of 77.5% IOU reported in [25]. Our implementation of the baseline method [7] did not show any reasonable results with current configuration.

#### 4.4. C.elegans dataset

Finally, we look at the C.Elegans dataset (Figure 7), which is available from the Broad Bioimage Benchmark Collection [20]. This sequence contains 97 two-channel images,  $696 \times 520$  pixels each, of roundworm C.elegans. Each image contains approximately 30 organisms, some of them in complex overlapping patterns. In order to compare our results with those reported in [21] we followed their protocol: the entire dataset was split into (two) equal parts: 50 training set and 47 test images. Here, we use the binary segmentation masks (following [21, 28, 30]). The network was trained for 1000 epochs.

We evaluate the instance segmentation using average precision (AP) metric computed with the standard COCO evaluation protocol [18]. Table 3 suggests that our method outperforms the previous works, including the well-known Mask-RCNN method [12] (as reported by [21]).

#### 4.5. Ablation study

We compare our method with the ablation that uses CoordConv in place of SinConv, as well as the ablations that use randomly initialized set of guide functions without optimization (three variants). Our unablated method is superior in all cases and across both datasets. In the case of Table 4,

Method	AP	AP <sub>0.5</sub>	AP <sub>0.75</sub>	AP <sub>S</sub>	AP <sub>M</sub>
Semi-convolutional operators [21]	0.569	0.885	0.661	0.511	0.671
Mask RCNN [12]	0.559	0.865	0.641	0.502	0.650
Discriminative loss [7] (our implementation without SinConv)	0.343	0.624	0.380	0.441	0.563
Discriminative loss [7] (our implementation with SinConv)	0.478	0.771	0.560	0.551	0.677
<b>Ours</b>	<b>0.724</b>	<b>0.900</b>	<b>0.723</b>	<b>0.775</b>	<b>0.875</b>

Table 3. Results on the C.elegans dataset. The results were obtained using the COCO standard metric [18]. Again, the proposed method outperforms previous approaches.

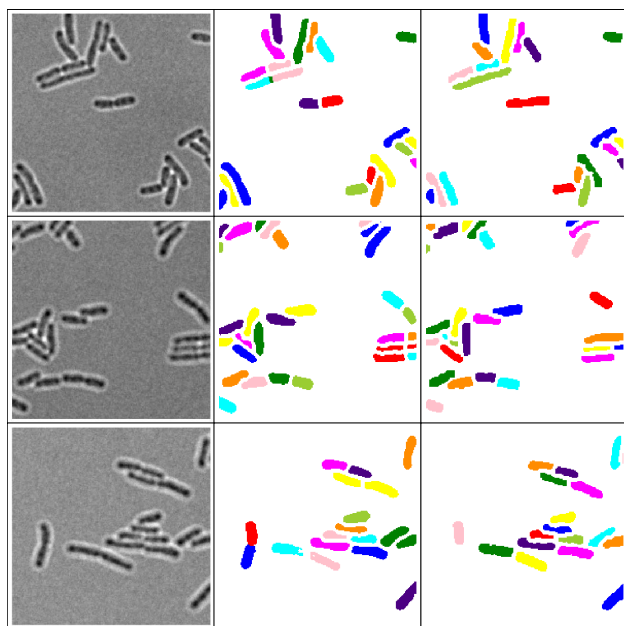


Figure 5. E.coli bacteria recorder under differential interference contrast microscopy. From left to right: raw image, result of our segmentation, ground truth labels.

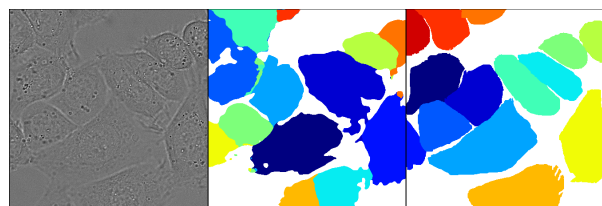


Figure 6. HeLa cells on glass recorded with differential interference contrast microscopy. From left to right: raw image, result of our segmentation, ground truth labels.

the advantage of our method over the baseline approach in [7] is not significant. However, in other cases (Table 5) it is considerable. It is not surprising that the ablation of our method without SinConv struggles, as the desired function is very hard to learn using fully-convolutional operations.

Method	DiC	SBD(%)
SinConv	3.	89.9
No guide	5.	78.3
CoordConv	3.	87.6
Random	11.	70.8
Low	5.	86.2
High	16.9	5.4

Table 4. The ablation study on the CVPPP dataset. We evaluate the variant without SinConv (standard convolutions – row 2) and with CoordConv replacing SinConv (row 3). The last three rows are the variants with guide functions sampled from the uniform distribution. Random - indicates the whole range of frequencies, Low and High - sampled from low and high frequencies, respectively. The unablated variant of our method (top row) performs best.

Method	AP	AP <sub>0.5</sub>	AP <sub>0.75</sub>	AP <sub>S</sub>	AP <sub>M</sub>
SinConv	0.724	0.900	0.723	0.775	0.875
No guide	0.028	0.063	0.023	0.105	0.081
CoordConv	0.436	0.631	0.414	0.485	0.685
Random	0.078	0.115	0.080	0.143	0.179
Low	0.273	0.498	0.240	0.350	0.503
High	0.001	0.002	0.002	0.010	0.01

Table 5. The ablation study of our framework on the C.elegans dataset. Same ablations as in the previous table are considered. The advantage of our full method over ablations is greater for this more challenging task.

#### 4.6. Method limitations

Despite state-of-the-art results on biological datasets, the proposed method has several limitations that need to be resolved before applying it to more complex datasets with severe variation in object scale such as COCO [18], PASCAL VOC [8], Cityscapes [5] (where our initial attempts to apply the method lead to mediocre, i.e., mid-table results). To the best of our understanding, the sub-par performance of the method is caused by the inability to handle very diverse scales gracefully.

To be specific, we have compared our method with Mask RCNN [12] implementation in PyTorch 1.2 [1] on MS COCO dataset [18] using our setup. We have used the validation part of the dataset; all images were transformed to the

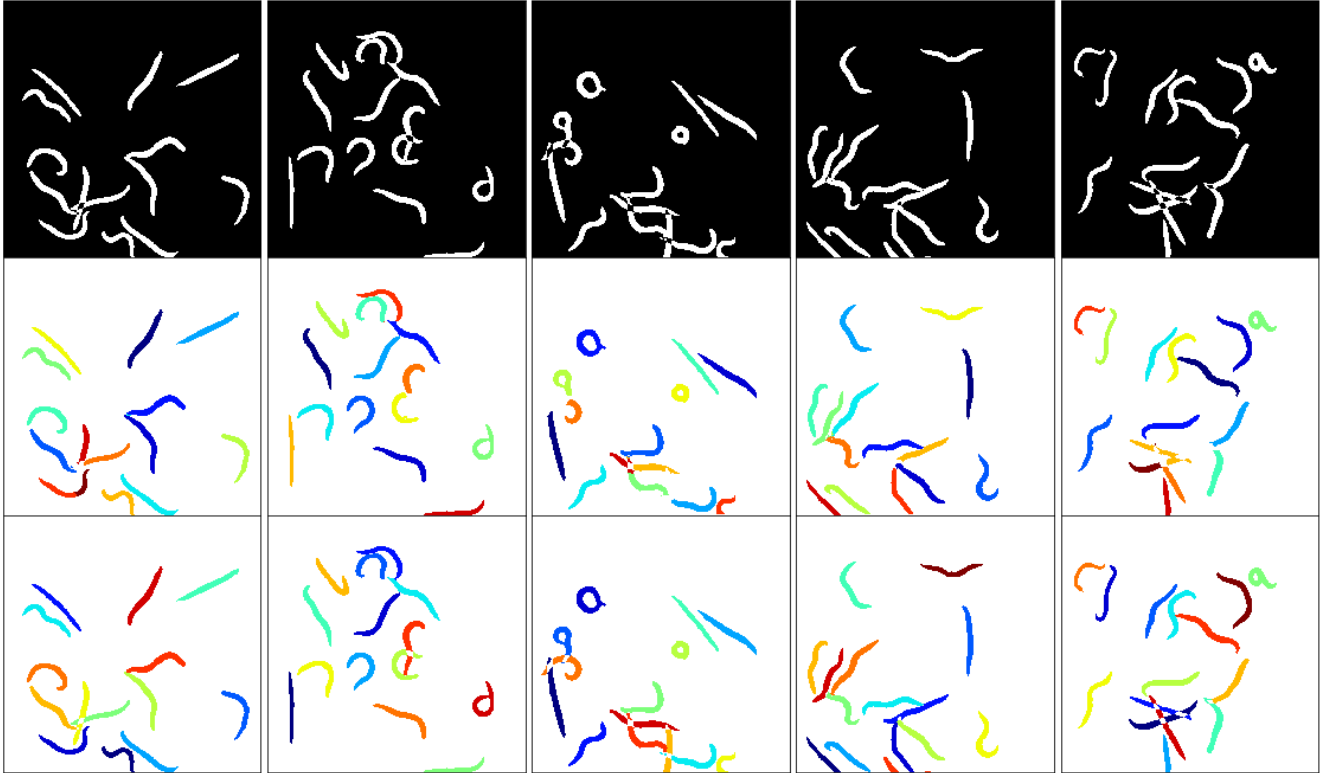


Figure 7. Binary images of *C.elegans* roundworm. Simple images crops (top row), second row - our results, third row - ground truth lables (color display needed).

<i>Method</i>	<i>AP</i>	<i>AP</i> <sub>0.5</sub>	<i>AP</i> <sub>0.75</sub>	<i>AP</i> <sub>S</sub>	<i>AP</i> <sub>M</sub>	<i>AP</i> <sub>L</sub>
Mask RCNN [12]	0.319	0.619	0.309	0.120	0.365	0.520
<b>Ours</b>	0.223	0.476	0.194	0.029	0.238	0.484

Table 6. Results on the COCO validation set (person class only). For this dataset with large scale variability, the results of our method are worse, especially for small objects.

448 × 448 resolution by down-sampling and zero-padding along the smaller dimension. The evaluation was performed on a single class: person (class=1). In this setting, the results of our methods are worse that those obtained using Mask RCNN (Table 6).

## 5. Conclusions

We have presented a new instance segmentation approach that exploits the peculiarities of biological images. The approach relies on a new type of embedding based on sine waves with parameters pre-adjusted to achieve separation of ground truth instances in the train dataset. We have shown that such precomputation of good embeddings greatly simplifies the learning stage whenever the same guide patterns are inputted in some of the convolutional layers of the embedding network. The ease of training is evidenced by the superior performance of our method com-

pared to the original embedding-based approach proposed in [7].

We have shown that our method can handle rather diverse biological image data, while using the same relatively small architecture and the same set of meta-parameters. Such versatility is valuable for practical deployment of the method to domain specialists.

**Acknowledgements.** This work was supported by the Skoltech NGP Program (MIT-Skoltech 1911/R). Computations were performed on Skoltech HPC cluster Zhores.

## References

- [1] PyTorch tensors and dynamic neural networks in python with strong gpu acceleration. <http://pytorch.org>. Accessed: 2018-11-15. 5, 7



- [2] A. Bieniek and A. Moga. An efficient watershed algorithm based on connected components. *Pattern recognition*, 33(6):907–916, 2000. 5
- [3] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3470–3478, 2015. 2
- [4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 4
- [5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 2, 7
- [6] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549. Springer, 2016. 2
- [7] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 1, 3, 4, 5, 6, 7, 8
- [8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>. 7
- [9] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017. 2, 3
- [10] M. V. Giuffrida, M. Minervini, and S. A. Tsafaris. Learning to count leaves in rosette plants. 2016. 5
- [11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 2
- [12] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 6, 7, 8
- [13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4
- [14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2
- [15] S. Kong and C. Fowlkes. Recurrent pixel embedding for instance grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9018–9028, 2018. 2
- [16] V. Kulikov, V. Yurchenko, and V. Lempitsky. Instance segmentation by deep coloring. *arXiv preprint arXiv:1807.10007*, 2018. 2, 3, 5, 6
- [17] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2
- [18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6, 7
- [19] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *arXiv preprint arXiv:1807.03247*, 2018. 4
- [20] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter. Annotated high-throughput microscopy image sets for validation. *Nat Methods*, 9(7):637, 2012. 6
- [21] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi. Semi-convolutional operators for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 86–102, 2018. 1, 3, 6, 7
- [22] J.-M. Pape and C. Klukas. 3-d histogram-based segmentation and leaf detection for rosette plants. In *ECCV Workshops (4)*, pages 61–74, 2014. 5
- [23] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 5
- [24] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *European Conference on Computer Vision*, pages 312–329. Springer, 2016. 2, 5
- [25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 4, 5, 6
- [26] H. Scharr, M. Minervini, A. Fischbach, and S. A. Tsafaris. Annotated image datasets of rosette plants. In *European Conference on Computer Vision. Zürich, Suisse*, pages 6–12, 2014. 5, 6
- [27] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, et al. Leaf segmentation in plant phenotyping: a collation study. *Machine vision and applications*, 27(4):585–606, 2016. 5
- [28] C. Wählby, T. Riklin-Raviv, V. Ljosa, A. L. Conery, P. Golland, F. M. Ausubel, and A. E. Carpenter. Resolving clustered worms via probabilistic shape models. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 552–555. IEEE, 2010. 6
- [29] X. Yin, X. Liu, J. Chen, and D. M. Kramer. Multi-leaf tracking from fluorescence plant videos. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 408–412. IEEE, 2014. 5
- [30] V. Yurchenko and V. Lempitsky. Parsing images of overlapping organisms with deep singling-out networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6280–6288, 2017. 6
- [31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4