Inpainting Networks Learn to Separate Cells in Microscopy Images

Steffen Wolf[1]
swolf@mrc-lmb.cam.ac.uk
Fred A. Hamprecht[1]
fred.hamprecht@iwr.uni-heidelberg.de
Jan Funke[2]
funkej@janelia.hhmi.org

[1] HCI
Heidelberg University
Germany

[2] HHMI Janelia
Ashburn, VA

Abstract

Deep neural networks trained to inpaint partially occluded images show a deep understanding of image composition and have even been shown to remove objects from images convincingly. In this work, we investigate how this implicit knowledge of image composition can be be used to separate cells in densely populated microscopy images. We propose a measure for the independence of two image regions given a fully self-supervised inpainting network and separate objects by maximizing this independence. We evaluate our method on two cell segmentation datasets and show that cells can be separated without any supervision. Furthermore, combined with simple foreground detection, our method yields instance segmentation of similar quality to fully supervised methods.

1 Motivation

Recent inpainting neural networks demonstrate a remarkable ability to remove distortions in natural images (*e.g.*, text overlays, watermarks, or pixel-wise independent noise) and are even able to entirely remove foreground objects. Trained on large datsets, these networks learn the statistics that underlie images in a way that goes well beyond low level features. In this work, we aim to leverage those learnt statistics to distinguish individual objects in images from each other, without any form of supervision. Let us consider a high-capacity inpainting network trained on a very large corpus of microscopy images and imagine the following scenario: Given the image of a cell culture with a region in the center masked out to inpaint, such a network will be able to continue inpainting cells that are partially visible. If, however, the masked-out region is large enough to contain entire objects, the provided context will be uninformative about their location, shape, and texture and will therefore not be able to recover those objects. In other words, the success of predicting masked out objects depends on the information about those object contained in the surrounding context.

In this paper we ask the question if the predictability of image regions given partial information can be used to separate instances. In particular, we define an information gain measure between image segments that can be approximated efficiently given an inpainting

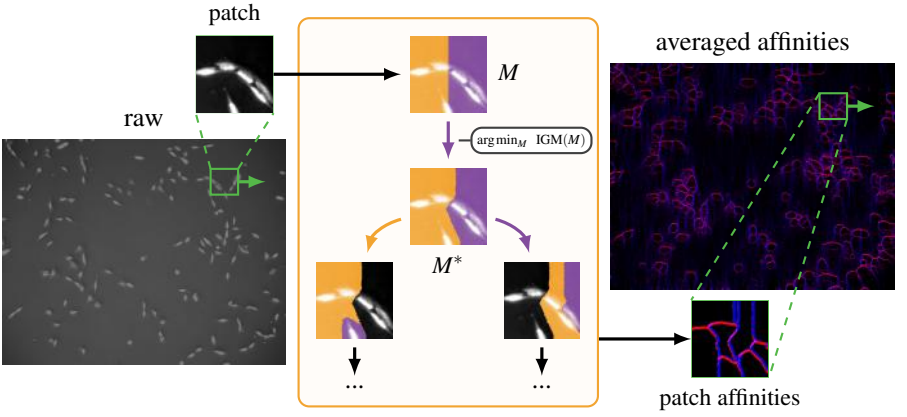


Figure 1: Extraction of instance separating affinities from an inpainting network. Given an image patch, we optimize a set of pixels M (shown in purple) to minimize the *information gain measure* (IGM), which is based on the predictions of a probabilistic inpainting network (see Section 3.2 and Fig. 2 for details). This optimization ensures that pixels in M^* provide minimal information about the intensity values of pixels in the complement $\overline{M}^*$ (shown in orange). We apply this procedure recursively to M^* and $\overline{M}^*$ to obtain a hierarchical segmentation of the image patch from which we extract affinities (shown in blue/red for x-/y-direction, respectively). These affinities are computed and averaged over a set of sliding image patches (green box) to obtain the final affinity estimates.

network. We show that minimizing this measure through a hierarchical optimization algorithm yields useful image decompositions. We represent those decompositions by *affinities*, *i.e.*, attractive or repulsive edges between pairs of pixels, which we average over a set of image patches in a sliding window fashion to obtain affinities for arbitrarily large images (see Fig. 1).

We evaluate our method on the problem of separating cells in microscopy images. We show that the inpainting network's understanding of image composition can be used to separate cells in images without any supervision. Furthermore, combined with simple foreground detection trained from few samples, our method yields instance segmentation of similar quality to fully supervised methods.

2 Related Work

While classical patch-based inpainting methods such as [2, 8, 26] synthesize high quality images, they fundamentally cannot make semantically aware decisions for intensity predictions. Deep inpainting networks, on the other hand, trained on large corpuses of data are known to develop an intrinsic understanding of images [17], which raises the question what aspects are captured by these networks. The usefulness of these inpainting models for image segmentation was shown by Pathak et al. [22], who demonstrate that features extracted from a trained inpainting network capture appearance and semantics of visual structures aiding in the pre-training of classification, detection, and segmentation tasks. Extending inpainting networks that directly minimize the reconstruction error [13, 31] with texture and structure

aware loss, such as multi-scale neural patch synthesis [32] or Structure-aware Appearance Flow [23] leads to high-fidelity images and prediction and modeling of higher order relations In parallel, specialized architectures and convolutions have been developed that make it possible to realistically inpaint arbitrary masks [19, 33].

In this work, we use the network architecture and loss proposed by Liu et al. [19], which is designed to inpaint arbitrary masks and is trained with an additional style component loss. Since we leverage the network's learned distribution by measuring information gain between image patches, we intentionally avoid networks trained with an additional GAN loss [5, 20, 34]. Although GANs produce extremely realistic looking images, they are prone to mode collapse that affects our estimate of information gain.

More generally, inpainting falls under the broader category of unsupervised prediction of left-out data, also known as *self-supervised* learning [6]. Self-supervised prediction objectives are formulated using only unlabeled data and, but do require higher-level semantic understanding in order to be solved [35].

Other than inpainting, self-supervised tasks include image colorization [16, 36], co-occurrence [9], predicting permutations [24], and denoising [14]. These methods are highly effective at extracting robust features for further transfer learning [37] and image embeddings [27] and can be considered a proxy task for developing a semantic understanding [17].

In some cases, the self-supervised task can be used as a free supervisory signal that directly translates to classically supervised tasks. For example, object tracking emerges from video colorization [28] or through obeying cycle-consistency in time [29]. When provided with background images and images with objects, Ostyakov et al. [21] learn to segment by predicting masks and paste patches from the object domain onto the background domain constrained by an adversarial and a cycle consistency loss.

Our work uses the statistical properties of instances to derive a method for separating instances, which closely relates to other self-supervised segmentation approaches that utilize different properties to identify objects. Burgess et al. [3] utilize compressibility, in a compositional generative model, where image regions are reconstructed through a low dimensional bottleneck. They show that their model is capable of discovering useful decompositions of scenes by identifying segments that can be represented in a common format. Another approach by Chen et al. [4] learns to find masks of objects by learning to replace the masked content that corresponds with altering the masked objects properties (e.g. altering the color of flowers).

3 Self-Supervised Instance Separation

In general, self-supervised instance separation or (more generally) segmentation is an under-constrained problem. What exactly constitutes a correct segmentation of an image depends not only on the application context (*e.g.*, segment all cells in a microscopy image), but also on a subjective level of detail (*e.g.*, segment nuclei and cell membrane individually). Without constraining assumptions or instructions, several different segmentations of the same image are plausible, leading to an intrinsic ambiguity. This ambiguity can be prominently observed as the inter-human variance for segmentation tasks where the concept of a segment is not precisely defined [1].

In the case of supervised image segmentation, this ambiguity is resolved by a set of training object instances in the form of, *e.g.*, affinities, labeled images, bounding boxes, or polygons. For self-supervised segmentation, on the other hand, assumptions about what

constitutes a segmentation have to fill in for the lack of training data.

Here, we propose to resolve this ambiguity by assuming that pixels of the same instance are more predictable from each other than across instances. We define the similarity between two pixels (and therefore the likelihood to be part of the same instance) as the information gained about the value of one pixel by observing the value of the other one. To obtain a segmentation, we therefore separate an image into segments that contain individual instances by repeatedly cutting the image (see 3.4). Our main insight is that a cut is more likely to separate instances if the information gain of pixel values across the cut is minimized. We quantify the information gain indirectly by measuring how accurately an inpainting network can predict the other side. Since we measure the accuracy pixel-wise, we can find an optimal cut by iteratively reassigning pixels that increase the inpainting inaccuracy.

Since inpainting networks are central to our method, we will revisit their probabilistic formulation in Sec. 3.1. We then show that those networks can be used to estimate the information gain (IG) of the given data to the inpainted pixels (Sec. 3.2). We use this estimate to introduce a new IG measure that estimates the information gain between two regions (groups of pixels). We show that this measure can be efficiently approximated, which requires only a minimal number of network inferences (Sec. 3.3). This efficiency makes it feasible to iteratively find optimal cuts and generate a segmentation as a set of optimal cuts (Sec. 3.4).

3.1 Self-supervised Inpainting

Let x_i be a random variable representing the intensity of pixel $i \in \Omega$, and x_M with $M \subseteq \Omega$ a set of random variables $\{x_i \mid i \in M\}$. Probabilistic inpainting is equivalent to learning a parameterized function $p_\theta(x_i|x_M)$, *i.e.*, the conditional distribution over intensities of pixel i, given known intensities of a partial observation M. The parameters θ of the distribution p_θ can be learned by minimizing the negative log-likelihood of a measurement $x = x^*$:

$$\mathcal{L}_{\text{inpaint}}(\theta; M) = \sum_{i \notin M} -\log p_\theta \left(x_i = x_i^* | x_M = x_M^*\right) \tag{1}$$

It is worth noting that this loss formulation resembles the objective of probabilistic NOISE2-VOID [15], highlighting the close connection between inpainting and denoising. In the next subsection, we will derive a similar connection between inpainting and instance separation.

3.2 Predictability is Affinity

Our central assumption is that the intensity value of a pixel in an instance is conditionally independent of all pixels outside the instance. In other words, pixel values should be well predictable given the values of other pixels in the same instance (high affinity). Conversely, pixel values from other instances should provide *no additional* information (low affinity). More formally, let $S = \{S_u \subseteq \Omega\}$ be a segmentation of Ω (*i.e.*, $\bigcup_u S_u = \Omega$ and $\forall u \neq v : S_u \cap S_v = \varnothing$), and let $S(i) \subseteq \Omega$ denote the segment containing pixel i. We assume that for the true instance segmentation S^*

$$p(x_i|x_{\Omega \setminus \{i\}}) = p(x_i|x_{S^*(i) \setminus \{i\}}), \tag{2}$$

i.e., that there is *no further* information gain provided by Ω compared to $S^*(i)$ for estimating the value of x_i. For any subset $M \subseteq \Omega$, let $\text{IG}(i|M)$ denote the additional information gained

for estimating the value of x_i when observing Ω compared to M alone, *i.e.*,

$$\mathrm{IG}(i|M) = D_{\mathrm{KL}}\Big(p(x_i|x_{\Omega\setminus\{i\}}) \;\Big|\Big|\; p(x_i|x_{M\setminus\{i\}})\Big), \tag{3}$$

where D_{KL} denotes the Kullback-Leibler divergence. Hence, $\mathrm{IG}(i|M)$ is a measure of how much x_i depends on values *not* contained in M.

Considering our assumption stated in (2), a sensible objective to recover a single segment of the true segmentation S^* would be to minimize (3) with respect to M. In practice, however, it would be unreasonable to assume that even for a correct segment M the information gain for pixels in this set from pixels outside this set is exactly zero. In other words, dilating M would trivially decrease $\mathrm{IG}(i|M)$ until $M = \Omega$. Therefore, instead of minimizing (3) directly, we propose to minimize a symmetric information gain measure. Let $\overline{M} = \Omega \setminus M$ be the complement of M. We introduce a *relative* information gain that indicates whether M or $\overline{M}$ provide more information about the value of x_i: $\mathrm{RIG}(i|M) = \mathrm{IG}(i|M) - \mathrm{IG}(i|\overline{M})$. The quality of a single segment M can now be assessed by the following symmetric information gain measure over all pixels i:

$$\mathrm{IGM}(M) = \sum_{i\in M} \mathrm{RIG}(i|M) + \sum_{i\in\overline{M}} \mathrm{RIG}(i|\overline{M}) = \sum_{i\in M} \mathrm{RIG}(i|M) - \sum_{i\in\overline{M}} \mathrm{RIG}(i|M). \tag{4}$$

3.3 Efficient Approximation of $\mathrm{IGM}(M)$

In its current form, $\mathrm{IGM}(M)$ requires evaluation of $\mathrm{IG}(i|M)$ for every pixel $i \in \Omega$. For each of these evaluations, $p_\theta(x_i|\cdot)$ has to be computed two times (conditioned on M and $\overline{M}$), which is too inefficient for a practical implementation.

To remedy this inefficiency, we make two approximations: First, we take advantage of convolutional neural network architectures that can inpaint an arbitrary set of pixels N for the same conditional [19]:

$$\prod_{i\in N} p_\theta(x_i|M\setminus\{i\}) \approx \prod_{i\in N} p_\theta(x_i|M\setminus N) \tag{5}$$

A similar approximation technique was first proposed by Krull et al. [14], who argue that this approximation is error-free for convolutional neuronal networks, if all pixels in N are spaced further apart than the field of view of the network. In our experiments, we find that even much denser subsets can be chosen without significant impact. We will refer to $\mathrm{RIG}(i|M)$ using this approximation as $\mathrm{RIG}_N(i|M)$ in the following.

Second, due to the limited field of view of the inpainting network, pixels far away from the conditional set have to be estimated via a constant prior and the relative information gain can therefore be computed without evaluating the neural network. Similarly, the complement conditional contains all pixels in the field of view and therefore yields zero information gain (assuming that all objects fit inside of the network's field of view). Thus, $\mathrm{RIG}(i|M) \approx \mathrm{const}$ for pixels far away from the boundary between M and $\overline{M}$.

In conclusion, limiting the computation of IGM to a specified region N close to the boundary combined with the approximate RIG_N leads to the following approximation:

$$\mathrm{IGM}_N(M) = \sum_{i\in M\cap N} \mathrm{RIG}_N(i|M) - \sum_{i\in\overline{M}\cap N} \mathrm{RIG}_N(i|M) \approx \mathrm{IGM}(M) + \mathrm{const} \tag{6}$$

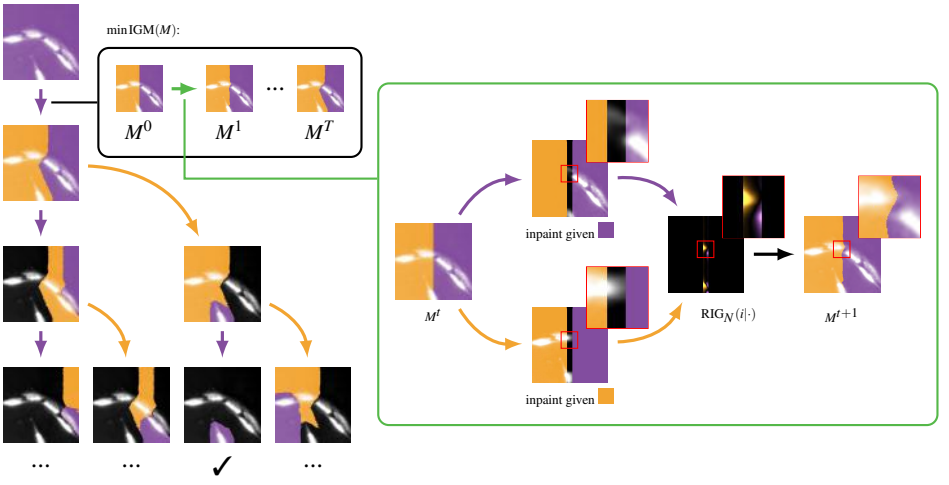


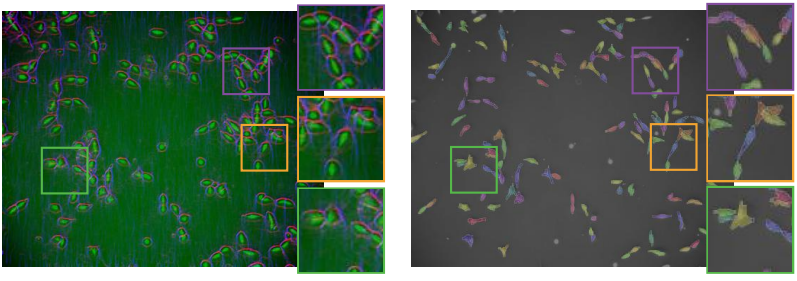
Figure 2: Details of the hierarchical segmentation of an image patch from an inpainting network. Given an image patch (top left), we recursively find optimal splits (shown in orange and purple) by evolving a randomly chosen horizontal or vertical split over T iterations (black box). For each step (illustrated in the green box), we evolve the boundary of the split by consulting a probabilistic inpainting network to predict the intensity of pixels in a region N around the boundary, once given only the information contained in M and once in its complement $\overline{M}$. We then measure the relative information gain RIG_N in the inpainting region to determine which component (orange or purple) provided more information about the pixels in N and reassign M accordingly.

3.4 Segmentation from Maximally Independent Regions

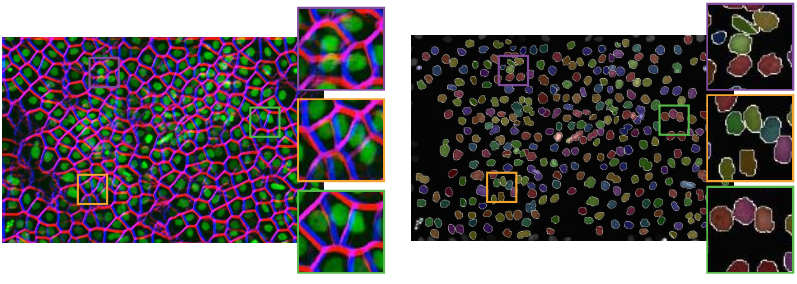
Although the approximation IGM_N introduced above reduces the computational burden of evaluating IGM, finding an optimal mask $M^* = \arg\min_M \ \mathrm{IGM}_N(M)$ still remains intractable in general due to the combinatorial number of possible masks. To understand which image regions are reconstructed independently by the inpainting network, we propose to solve this optimization problem by following a greedy optimization strategy that generates a sequence of masks M^t for $t \in \{0,\dots,T\}$ such that $\mathrm{IGM}_N(M^{t+1}) \leq \mathrm{IGM}_N(M^t)$, illustrated in Fig. 2.

To this end, we first separate Ω into two equally sized components M^0 and $\overline{M^0}$ by randomly splitting them horizontally or vertically. We then evolve the boundary of the split by evaluating $\mathrm{RIG}_N(i|M^t)$ for all pixels $i \in N$ in close proximity to the current boundary between M^t and $\overline{M^t}$. The sign of $\mathrm{RIG}_N(i|M^t)$ indicates whether M^t or $\overline{M^t}$ provide more information about the pixel i. We update M accordingly, *i.e.*, $M^{t+1} = (M^t \setminus N) \cup \{i \in N \mid \mathrm{RIG}_N(i|M^t) > 0\}$, which, by definition of (6), monotonically decreases IGM_N.

Finally, in order to obtain a decomposition of an image into arbitrarily many maximally independent regions, we apply the minimization recursively to already identified regions, *i.e.*, we repeat the optimization procedure described above on regions M^* and $\overline{M^*}$, until either M^* or $\overline{M^*}$ are empty. Further implementation details on our neighborhood selection can be found in the Supplement.



(a) INPAINTAFF (b) INPAINTAFF +TRUEFG



(c) INPAINTAFF (d) INPAINTAFF +TRUEFG

Figure 3: Instance separation results assuming an accurate foreground detection TRUEFG on the PANC dataset (top row) and the HELA dataset (bottom row). A foreground detection alone is not sufficient to segment touching cells (a, d). INPAINTAFF extracted from an inpainting network find non-trivial splits between instances (b, e).

4 Experiments on Cell Segmentation

To quantitatively answer our question if predictability of image regions can be used to segment or separate objects, we apply our method to microscopy images of cells. In those images, ggcells move freely in a substrate and can thus be considered as many independent instances of the same kind, which makes them suitable for the independence assumption we made in (2). Nevertheless, due to their high density, they pose a challenging segmentation problem to evaluate the intrinsic knowledge of inpainting networks by measuring the separation/segmentation accuracy. In the following, we will refer to the affinities extracted using our method as INPAINTAFF.

4.1 Cell Segmentation Benchmark Dataset

We evaluate INPAINTAFF on a subset of the ISBI Cell Segmentation Benchmark, which includes a diverse set of 2D microscopy videos covering a wide range of cell types and imaging quality. In particular, we selected two datasets that contain cells of irregular shape in close proximity for which instance separation is needed to obtain a correct segmentation:
(1) HELA contains cervical cancer cells expressing H2b-GFP and (2) PANC contains pancreatic stem cells on a polystyrene substrate (see the CTC website for further information about the datasets). Both datasets belongs to the most dense datasets of the ISBI Cell Segmentation Benchmark[1]. Therefore, a mere foreground segmentation is ineffective for the

[1]The PANC and HELA cell density is 2 s.d. higher than the average CTC cell density.

detection of individual cells. Additionally, both datasets contain only little labeled training data (815 instances[2] for HeLa and 514 for Panc in fully labeled frames), which challenges fully supervised segmentation approaches.

4.2 Results

As argued earlier, segmentation without supervision is an under-constrained problem. As such, InpaintAff alone is unlikely to give rise to a segmentation capturing the intuition of a human annotator. We recall that the main guiding principle for InpaintAff is predictability of pixel intensities. Depending on the distribution of cells in images used to train the inpainting network, this predictability might equally well apply to a background region around each cell. This effect is visible in both datasets (compare Fig. 3 and further images in the supplement Fig. 5 and Fig. 6) and demonstrates that the method is agnostic about the intensity of pixels and merely clusters pixels that are mutually predictable.

We investigate first how well inpainting networks understand image composition by measuring how well InpaintAff *separates* instances. For our quantitative analysis, we decompose the problem of cell segmentation into a foreground/background classification and an instance *separation* task with affinities. We solve the separation task fully self-supervised and evaluate it given a) ideal ground truth (see *Instance Separation*) and b) a foreground/background classifier with minimal supervision (see *Instance Segmentation from Foreground Prediction*).

We report results using the ISBI Cell Segmentation Benchmark segmentation accuracy (SEG score), a metric that is based on the Jaccard similarity index and measures average IoU of all segments that overlap at least 50% with the ground truth (further details are given on the challenge website). The detection score is the percentage of matches that surpass a set IoU threshold.

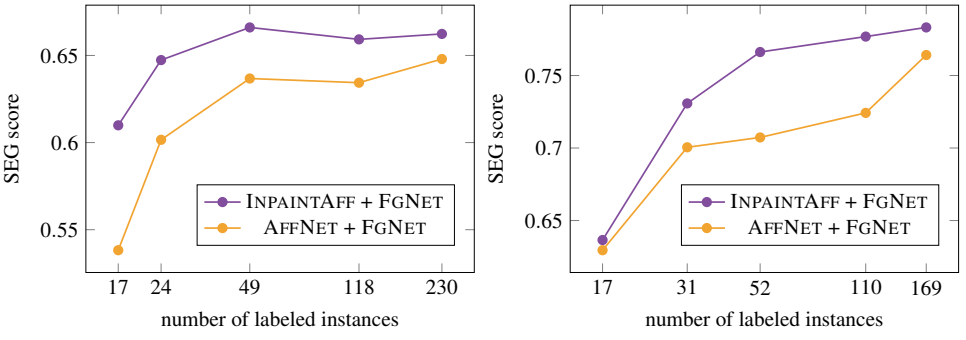
Method		HeLa	Panc
Conn. Comp. + TrueFg		0.785	0.748
InpaintAff + TrueFg		0.858	0.914
InpaintAff + FgNet50		0.766	0.666
Top Entries of the CTC		**HeLa**	**Panc**
HIT-CN*	MU-Lux-CZ*	0.919	0.715
FR-Ro-GE*	CVUT-CZ*	0.903	0.682
PURD-US*	HD-Hau-GE*	0.902	0.665

Table 1: Segmentation scores assuming an accurate foreground detection TrueFg and FgNet50 (trained with 52/49 labeled instances for HeLa/Panc). For reference, we include the official challenge scores of supervised methods on the same datasets (marked with a star), which have been trained on more labeled instances and evaluated on a different testing dataset then our method.

Instance Separation We investigate how well InpaintAff separates instances, assuming that an accurate foreground segmentation is already available. For that, we use the ground-truth segmentation provided in the datasets and convert it into a binary foreground segmentation TrueFg.

As we show in Table 1 (and qualitatively in Fig. 3), connected component analysis on TrueFg alone is not sufficient to achieve an accurate instance segmentation, due to merges of cells in close proximity. Separating those cells using InpaintAff, however, results in an almost perfect instance segmentation, in the case of Panc even significantly exceeding

[2] HeLa has 571 additional instances. In partially labeled frames, which can not trivially be used to train neural networks.



(a) PANC (b) HELA

Figure 4: Segmentation score on the test data of datasets PANC and HELA, for varying amounts of labeled instances used to train FGNET and AFFNET.

the scores of the best performing methods of 0.715 (albeit on different testing data and constrained to the ground-truth foreground). Those results suggest that (1) INPAINTAFF is accurately separating instances, and (2) a foreground segmentation is necessary and sufficient to constrain the boundaries of found objects to obtain a competitive segmentation.

Instance Segmentation from Foreground Prediction Since a foreground segmentation is crucial to capture the application specific notion of what constitutes an object, we next investigate the segmentation accuracy of our method when combined with a foreground prediction network trained on few instances only, which we will refer to as FGNET (details in Section 4.3). We train FGNET on varying amounts of labeled instances to predict a binary foreground mask and use this prediction in combination with our INPAINTAFF to obtain an instance segmentation. As a baseline, we also train a second network AFFNET to predict affinities directly from the same labeled instances used to train the foreground network.

The segmentation scores for either approach on the test dataset are shown in Fig. 4, for varying amounts of labeled instances used for training. Remarkably, INPAINTAFF consistently outperform trained affinities in terms of the SEG score. This effect is most visible in dataset PANC, where cells tend to cluster more compactly and the separation of individual cells is therefore more challenging. In particular, INPAINTAFF on this dataset in combination with FGNET trained on as few as 24 labeled cells produces a segmentation that outperforms the fully supervised AFFNET using one order of magnitude more training data. As shown in the supplement Fig. 7, this observation also holds in terms of the detection score over varying IoU thresholds. Furthermore, the obtained segmentation score on the PANC dataset using only around 50 labeled instances for the foreground prediction together with self-supervised affinities is on par with the third leading submissions to the ISBI Cell Segmentation Benchmark, which have been trained on 514 instances (albeit evaluated on a different testing dataset than used here).

4.3 Experiment Details

Training and Testing Split We use the inpainting network architecture and training procedure of [19].

Since INPAINTAFF requires a considerable amount of computational resources (see discussion in Section 5) a direct evaluation on the CTC servers on the official testing data is not possible. Therefore, we split the publicly available data for each dataset into a train and testing dataset, each containing one video of sparsely labeled cells. Further details about the network architecture, training and segmentation inference can be found in the supplement.

Affinity-Based Segmentation A segmentation can be derived directly from the affinities with an agglomorative clustering algorithm. We use the MUTEXWATERSHED [30] in our experiments, but other clustering algorithms (*e.g.* GAEC[11] and GF[18]) are equally viable.

We further introduce a single parameter α to control for over- and undersegmentation by multiplying all long range affinities (that are used to split) with α. The optimal α for each evaluated method was determined on the validation dataset. Further details can be found in the supplement.

5 Discussion

It remains an open question as to how far segmentation based on image statistics alone (without any supervision) will find real world applications. As we already observed on the segmentation of cells in microscopy images studied here, an experimentalist's intention of what constitutes a good cell segmentation does not necessarily match the clustering of pixels based on information content. Only at least partially supervised methods with application specific losses can ultimately produce predictions tailored to a specific application, provided enough labeled training data is available.

We see the contribution of this work therefore primarily as a demonstration of the capabilities of inpainting networks to implicitly group pixels of an object without explicit supervision. We observe that this is an especially challenging tasks in microscopy images without obvious cues like color to separate instances: the unsupervised method ReDO [4], which demonstrated impressive results in segmenting flowers of different colors in RGB images, did not converge to produce foreground masks in our experiments[3]. Furthermore, with minimal supervision, our method yields accurate segmentations which is relevant in bioimage analysis scenarios in which labeled training data is scarce. A current limitation of our method is the runtime: INPAINTAFF requires around 48h to process a 700x1100 image on a single GPU. Although inference can be trivially parallelized, the current implementation might be prohibitively slow for many applications. We observed another minor limitation in the early phases of our experiments: Less accurate inpainting networks indeed yield a significantly worse segmentation result. We solved this by adapting the training procedure of [19], training on a variety of random masks. Crucially this network should not be finetuned on the maximally independent regions, since their shape alone suggests the position of other instances.

The goal of our future work is to transfer the segmentation capability, measured in this paper, to networks that directly predict image segments.

[3] All pixels were assigned to background (data not shown). ReDO was trained with default parameters on random crops of our dataset.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33 (5):898–916, May 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.161.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009.

[3] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[4] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. In *Advances in Neural Information Processing Systems*, pages 12705–12716, 2019.

[5] Zeyuan Chen, Shaoliang Nie, Tianfu Wu, and Christopher G Healey. High resolution face completion with multiple controllable attributes via fully end-to-end progressive generative adversarial networks. *arXiv preprint arXiv:1801.07632*, 2018.

[6] Virginia R de Sa. Learning classification with unlabeled data. In *Advances in neural information processing systems*, pages 112–119, 1994.

[7] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[8] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *ACM SIGGRAPH*, pages 303–312. 2003.

[9] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H Adelson. Learning visual groups from co-occurrences in space and time. *arXiv preprint arXiv:1511.06811*, 2015.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[11] Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guillaume Lavoué, Thomas Brox, and Bjorn Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1751–1759, 2015.

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

[13] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534. Springer, 2014.

[14] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019.

[15] Alexander Krull, Tomas Vicar, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *arXiv preprint arXiv:1906.00651*, 2019.

[16] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016.

[17] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017.

[18] Evgeny Levinkov, Alexander Kirillov, and Bjoern Andres. A comparative study of local search algorithms for correlation clustering. In *German Conference on Pattern Recognition*, pages 103–114. Springer, 2017.

[19] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*, 2018.

[20] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

[21] Pavel Ostyakov, Roman Suvorov, Elizaveta Logacheva, Oleg Khomenko, and Sergey I Nikolenko. Seigan: Towards compositional image generation by simultaneously learning to segment, enhance, and inpaint. *arXiv preprint arXiv:1811.07630*, 2018.

[22] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[23] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 181–190, 2019.

[24] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. DeepPermNet: visual permutation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3949–3957, 2017.

[25] Thorvald Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34, 1948.

[26] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. In *ACM SIGGRAPH*, pages 861–868. 2005.

[27] Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*, 2019.

[28] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 391–408, 2018.

[29] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[30] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, Ullrich Kothe, and FredA Hamprecht. The mutex watershed: efficient, parameter-free image partitioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 546–562, 2018.

[31] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pages 341–349, 2012.

[32] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6721–6729, 2017.

[33] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[34] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1486–1494, 2019.

[35] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1476–1485, 2019.

[36] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[37] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.