TECHNICAL UNIVERSITY OF MUNICH

TUM Data Innovation Lab

Machine learning-based image detection for lensless microscopy in life science

Authors	Jan Brunckhorst
	Andreas Pirchner
	Nikhitha Radhakrishna Naik
	Mahalakshmi Sabanayagam
Mentor	Dr. Korbinian Paul-Yuan, Dr. Philipp Paulitschke
	Soft Condensed Matter Group (Faculty of Physics, LMU)
Co-Mentor	Michael Rauchensteiner (Department of Mathematics, TUM)
Project Lead	Dr. Ricardo Acevedo Cabra (Department of Mathematics, TUM)
Supervisor	Prof. Dr. Massimo Fornasier (Department of Mathematics, TUM)

Jul 2019

Abstract

Cell counting and estimating the cell covered area of in vitro cell culture are common tasks in cell biology and hold significant importance in life science research. These tasks are performed manually most often. Recent advancements in detecting the positions of cell nuclei [8] have shown that the deep convolutional neural networks have the potential to automate these tasks. Here, we address both cell counting and covered area detection for lens-free images. We use an annotated dataset for the cell counting task. However, we only have raw bright-field and lens-free images taken at the same instance for the cell covered area estimation. Using the bright-field images we generate annotations by applying standard image processing techniques and consider them as the ground truth for lens-free images to train neural networks. The best model achieves an F1 score of 0.84. We approach the task of cell counting by localizing individual cells rather than estimating merely the object count. We train a neural to learn a distance transform with local maxima corresponding to cell centers. The final model achieves an F1 score of 0.97 and a relative counting error of 2.13%. We compare multiple neural network architectures and show that the LinkNet outperforms the U-Net on both tasks.

Contents

1 Introduction

1.1 Problem Definition and Goals of the Project

Observing the growth and behavior of cells is an important part of many experiments in life sciences. Ranging from medical diagnosis to the development of new medication for curing cancer, highly skilled professionals spend hours of their precious time looking through a microscope to complete tedious tasks such as cell counting as illustrated in Figure 1. The availability of reliable automated tools for monitoring the growth and behavior of cells is essential for speeding up such repetitive tasks, increasing the efficiency of workflows and giving researchers time to focus on more challenging problems.

Figure 1: Manual cell counting with a hemocytometer[1]

This work aims to develop such tools for a very specific kind of imaging device: the lens-free microscope (LFM) (e.g. [4] or [8]). Instead of having optical components, these microscopes utilize a complementary metal-oxide-semiconductor (CMOS) sensor to record interference patterns of cells when illuminated with a light source. While cell monitoring tools are readily available for more common types of microscopes, there has been little work done in the field of LFM. The advantages of LFM are the relatively small price and size compared to optical devices. This also allows monitoring cells in their preferred environment.
The downside, however, is the inferior quality of the images. Despite this, the ability to produce correct results is crucial for the acceptance of LFM and thus is the main goal of our work. More specifically, we want to implement robust LFM image-based algorithms for two common tasks: cell counting and estimating the cell covered area.

Cell counting Counting cells is an essential step in many experiments in microbiology and medicine. For example, a researcher needs to know the concentration of cells in a predefined volume of liquid in order to determine the amount of a chemical agent she has to apply to the cells for an experiment. Most commonly, cells are counted by using a Neubauer counting chamber (also called hemocytometer) - essentially a glass plate with engraved perpendicular lines that form a grid. Usually, a researcher uses the grid to count the cells through a microscope as illustrated in Figure 1.
We strive to provide automated cell counting for lens-free counting chambers. Figure 3 highlights the general scope of the task. Our algorithm has to be able to learn the shape of cells in order to reliably detect cells even in the presence of noise in the form of contamination with e.g. tissue fibers and interference patterns. Another challenge is the separation of distinct cells in crowded portions of images.

Cell-covered area Estimating cell covered area is important to monitor cell growth, for instance it acts as an observable for the effects of different drug dosages. The cell

[1]Source: `https://www.biocompare.com/Editorial%2DArticles/189708%2DAutomated%2DCell%2DCounting%2DSelecting%2Dthe%2DAppropriate%2DSystem`

types considered for this task come in different shapes (pointed, elongated and so on) and sizes. They are also known to be mobile in nature. So, detecting and segmenting individual cells will be more difficult compared to segmenting clusters of cells. In addition, it is essential to have accurate information about the cell growth behaviour in vitro cell culture for harvesting and to avoid over growth. Hence evaluating the cell covered region is significant and beneficial. The goal of the project is to determine the actual cell covered region in lens-free microscopic images.

This report is structured as follows: After this introduction to the project and its goals, we briefly review the literature related to our work. We then go on to introduce and describe the data sets used to train and evaluate our algorithms. After that, we describe both the methods used to solve the tasks but also the metrics to evaluate their performance. Then we show the results of our experiments and compare various approaches. Concluding this report, we interpret the results and put them into context.

1.2 Related Work

Automatic cell detection in traditional light microscopy has received considerable research attention, focusing on cell segmentation (Ronneberger, Fischer, and Brox [9], Buggenthin et al. [2]) or cell detection and counting (Xie, Noble, and Zisserman [11], Khan, Gould, and Salzmann [7], Kainz et al. [6]). Ronneberger, Fischer, and Brox [9] proposed the U-Net as the convolutional neural network for bio-medical image segmentation and won the ISBI cell tracking challenge in 2015. The U-Net is still the neural network architecture most bio-medical researchers choose for image segmentation. For cell counting, or in general object counting, different approaches exist. Counting by detection localizes individual objects in an image which makes counting trivial afterward. Most methods apply non-maximum suppression to extract local peaks corresponding to individual cells. However, this procedure can become difficult for overlapping cells if segmentation masks are used. Therefore, Kainz et al. [6] and Xie, Noble, and Zisserman [11] propose to learn Euclidean distances instead.. Other methods avoid solving the hard detection problem and only estimate the objects count. Similar, Khan, Gould, and Salzmann [7] estimate an image density whose integral over any image region gives the count of objects within that region. However, only a few pieces of literature on cell detection for LFM images exist. Rempfler et al. [8] show that fully convolutional neural networks can achieve high performance in cell detection even for LFM.

2 Data

For the two sub-task of counting cells and estimating the cell-covered area, we work on two separate image datasets. The following section gives a detailed view into the composition of these sets and how we process them. Before that, we also give a brief description of the cell types[2] appearing in the data to give a meaning to the cryptic identifiers. Cells of type 3T3 are isolated from house mice and are mainly used for transfection studies with DNA viruses. Type A549 are lung carcinoma cells obtained from humans and are used

[2]Descriptions are taken from the Leibnitz Institute DSMZ database (`https://www.dsmz.de/`)

for cancer research in mice. Line HuH7 are cells taken from human liver carcinoma[3] and are used for cancer research. Cell type HeLA are human cervix cancer cells and are also used for cancer research.

2.1 Neubauer Counting Chamber

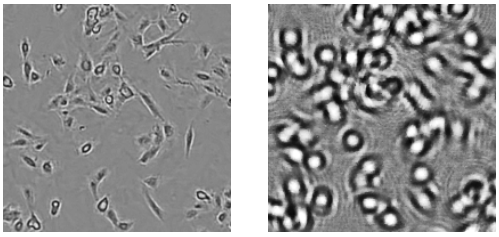
For the cell counting task, we have access to 138 annotated images for training, evaluation and testing, of which 52 are cell line 3T3, 68 are A549 and 19 are HuH7. The images were obtained from a lens-free Neubauer counting chamber, the cell centers were annotated manually by qualified personnel. Due to the preparation of the cell suspension for the counting chamber, one can not really observe differences between the cell lines in a counting chamber. The mean cell counts for the different cell lines are 261.90 for 3T3, 367.36 for A549 and 244.00 for HuH7. All images are of size 1600×1200 and thus are too large to be processed in one run by standard network architectures applied in computer vision such as Resnet [5]. To overcome the issue of having a relatively small number of large images available for training, we employ two separate preprocessing routines during training and test time: For training, we sample random patches of size 224×224 from the large images. This allows us to leverage the available images into a massive number of distinct (albeit possibly highly correlated) training samples. The details regarding the random patch-sampling can be found later in this section.
During test-time, we switch to a scheme that cuts the image into 48 neighboring tiles of size 224×224 that are then put into a single batch for the cell detector's forward pass. After that, the image is reconstructed into its original spatial shape. This allows to utilize the GPU's parallel computing capabilities even on single images, resulting in very fast cell counting.
We split the data randomly into a training set of size 97 to train the cell detection neural networks, an evaluation set of 14 images for detecting over-fitting during training and to determine the optimal threshold for local maximum detection as described later. The remaining images are used as a test set.

2.2 Cell Covered Area

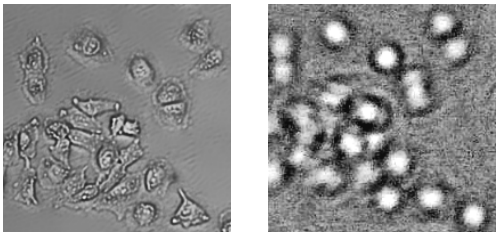
Bright-field and lens-free microscopic images are available for this task. These images are sequence of time lapse captures, wherein images are taken at a given time in parallel using both the microscopes. These images belong to 4 cell types namely 3T3, A549, HeLa and HuH7 among which the images from the first three cell types are time lapse. There are totally 2 sequences each of HeLa and 3T3 cell types and 1 sequence of A549 cell type in the dataset. The number of images available are 295 of which 136 of HeLa, 112 of 3T3, 31 of A549 and 16 of HuH7. The images are approximately 2000×2500, so random patches of 224×224 are extracted from each image which amounts to $\sim$23000 total patches.
Typically, the size of the cells in lens-free images are larger than those in the bright-field microscopic images and differ in shapes as shown in Figure 2. Thus, the correlation between the two microscopic types is not uniform as the shape and size regime of the cells in the bright-field microscopy differ among different cell types whereas the images from lens-free are elliptical in shape and larger in size compared to the actual cells in

[3] see `http://huh7.com/`

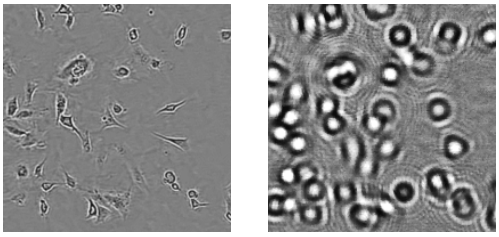
all the cases. As a result, segmenting lens-free images using standard image processing techniques for detecting cell covered area will be erroneous. It is therefore necessary to consider the area covered by cells in bright-field images to build a model that finds the correlation between the bright-field and lens-free images for all the cell types. To find the generalized correspondence between the two microscopes, we first generate annotations from the bright-field images using image processing techniques. Then, use these annotations as the ground truth to build a model with the input being lens-free microscopic images. Important observation is that, there are some discrepancies in the cells presence and location in the images due to continuous movement of the cells, calibration and human errors. Additionally, short intervals in capturing the images on the two microscopes add to this problem.



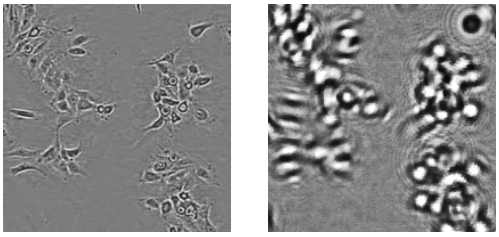
A549 bright-field and lens-free image



HuH7 bright-field and lens-free image



3T3 bright-field and lens-free image



HeLa bright-field and lens-free image

Figure 2: All cell types in the dataset with bright-field and lens-free microscopic images to explain the difference in the size and shape of the cells.

2.3 Data Preprocessing and Augmentation

Image preprocessing for cell counting We convert the raw images to grayscale and perform a background correction and Gaussian blurring. Since the lighting under the lens-less microscopy is red, the red channel is pretty overexposed, the blue channel is hardly signaled, and the green channel is the most contrasted. Therefore the grayscale conversion reduces the number of channels while keeping the relevant information, i.e. the lightness and contrast of pixels. Before feeding the data to the neural network, we normalize all images to zero mean and unit variance, so that the background has values near zero.

Image preprocessing for cell-covered area The cell images from both the microscopes are grayscale and noisy. The lens-free images that are fed into the neural network are denoised using adaptive histogram equalization technique. This gets rid of the obvious interference patterns around the cell region and enables a smooth gradient in the regions

where there is absence of cells. The bright field images are denoised using fast non-local denoising technique. A shift is observed between the bright-field and lens-free images for some of the cell types. To fix this issue, homo-graphic transformations are applied to realign the corresponding images.

Data augmentation We apply random cropping, flipping, and rotation to augment our dataset. Because the detection of cells is invariant to shifts and other linear transformations, we can easily enhance our limited amount of annotated data available. But unlike Ronneberger, Fischer, and Brox [9], we do not use any elastic deformations of the training samples, to keep the circular shape of cells under the lens-less microscope. During training for both lensfree cell counting and cell covered area detection, we extract random crops of size 224 x 224 from the large images and flip or rotate by a random multiple of 90° to randomize the position of cells. For the evaluation or prediction process, the input pipeline applies zero padding to the input images such that the width and height are multiples of 224 and splits the image in non-overlapping patches of shape 224 x 224.

3 Methods

3.1 Lens-free Cell Counting

To locate and count individual cells under the lens-free microscope from the Neubauer Counting-Chamber, we use a two-stage approach. First, we train a fully convolutional neural network to produce a probability map given an image. We then apply non-maximum suppression to extract all local maxima, which should correspond to the cell centers. To evaluate the model performance, we do not use the counting error as a standalone metric, but also precision and recall of cell center predictions. For the training of the probability map, we consider two different approaches. The first one treats the task as a binary segmentation problem, while the second one uses a proximity map based on a Euclidean distance transform of the cell centers.

3.1.1 Ground Truth Generation

Let $A \in \mathbb{R}^{m \times n}$ be a microscopy image containing cells and $C = \{c_i\}_{i \in I}$ the set of annotated cell coordinates. Following [10], let D_C be the corresponding Euclidean distance transform with $D_C(x)$ being the distance from location $x \in \mathbb{R}^2$ to the nearest cell center in C. In the binary segmentation approach, a model has to predict whether a pixel belongs to a cell or the background. We generate the corresponding ground truth matrix by assigning each location x that is within a given distance d to a cell center to class 1 and vice versa:

$$p(x) = \begin{cases} 1 & \text{if } D_C(x) \leq d, \\ 0 & otherwise. \end{cases} \tag{1}$$

We choose $d = 8$, which is smaller than the average radius of a cell under the lens-less microscopy, to reduce the overlap of close cells. To exploit the additional context of the distance to the nearest cell and produce unique local extrema corresponding to detected cell centers, the Euclidean distance transform seems like a reasonable choice. However, this

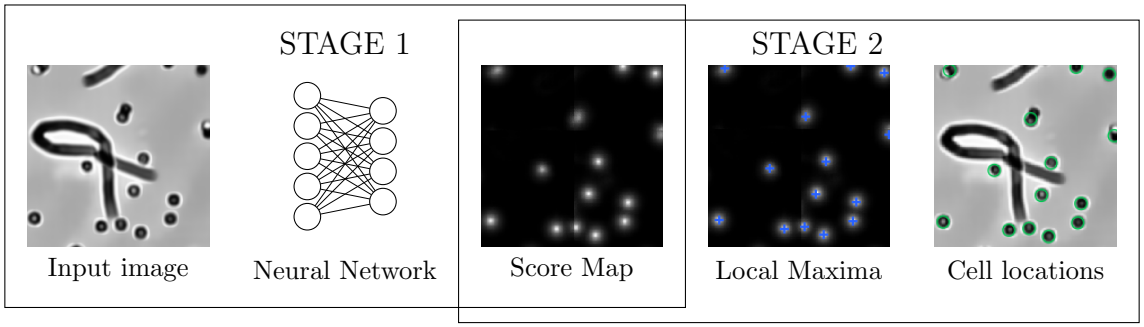


Figure 3: Illustration of the two-stage cell-location process. The work flow starts at the left with the preprocessed lens-free microscopy image and ends at the right with the predicted cell centers. The neural network estimates a Euclidean distance transform from section 3.1.1.

would lead to high and varying scores for different regions in the background. Therefore, [6] and [10] propose a transformed Euclidean distance map that is flat in the background and has distinctive peaks at all cell centers:

$$d(x) = \begin{cases} \exp(\alpha(1 - \frac{D_C(x)}{d_{max}})) - 1 & \text{if } D_C(x) \leq d_{max}, \\ 0 & otherwise \end{cases}, \tag{2}$$

where α and d_{max} control the shape of the exponential function. As with the binary mask we generate the ground truth matrix by applying d over each location in A. Kainz et al choose $\alpha = 5,\ d_{max} = 16$ for the ICPR and $\alpha = 3,\ d_{max} = 39$ for the BM dataset, which contain microscopy images. In the following, we will work with two specifications for our dataset, namely $\alpha = 5,\ d_{max} = 16$ and $\alpha = 4,\ d_{max} = 24$.

3.1.2 Locating Cells

We approach the task of locating cells in two stages rather than training a single model to find cell centers in an end-to-end fashion. Training the neural network cell detector on whole maps rather than just lists of (x, y)-pairs makes it easier for the model to filter for cell features as the expected shape of the cells is implicitly given by the ground truth map. We illustrate the two-stage procedure in Figure 3.

Stage 1 In a first step, a fully convolutional neural network takes an input image and produces either a probability map (with the binary segmentation objective) or a Euclidean distance map (see 3.1.1 for details). In the segmentation case, a high value corresponds to the cell detector's confidence in finding a cell at this position. After training with the Euclidean distance map, the value corresponds to the position's predicted distance to the nearest cell center - the higher the value, the closer the center.

Stage 2 The second stage takes the map produces by the cell detector as input. Local maxima of this map indicate positions where the cell detector network has the highest confidence in finding a cell center there within a local neighborhood. Therefore the local maxima of a well-trained detector should correspond to the true cell centers. We single out

local maxima by dilating the map with a maximum filter. To ensure that local maxima are at least d pixel apart, the filter has to be of size $(2d+1) \times (2d+1)$. After dilation, all positions but the local maxima will have changed in value. So by comparing the original with the dilated map, we obtain the predicted cell center positions as a list of (x, y)-pairs. When looking at the map in the center of Figure 3, one notices that the cells are quite far apart. While the values in this space should be equal to zero, the cell detector will assign some small positive value. By just naively looking for local maxima, one would obtain lots of spurious cell center detections due to this noise. We solve this by considering only maxima above some threshold, which is determined experimentally as will be discussed in 4.1. For convenience, we use the implementation `peak_local_max()` in the Python package `scikit-image`[4].

3.1.3 Metrics used for Model Evaluation

When thinking of a way to evaluate cell counting, it is tempting to purely consider a metric such as the mean squared (MSE) or absolute (MAE) counting error. While they certainly are the most important metrics, in the end, MSE and MAE do not allow to check whether the cell detector actually locates cells. Imagine there are n cells in an image. A good cell detector locates all of them, while a random detector may just also yield n cell centers - from a counting perspective, this means that both are equally suited for the task. A strong cell detector allows robust cell counting. A correct cell count, on the other hand, does not imply that the cells were actually detected. To build confidence and trust in our method, it is thus imperative to first evaluate the performance in locating cells. We outline an alternative approach for the model evaluation in the following paragraph.

Precision, recall and F1 Calculating precision and recall for the predicted maps in a familiar pixel-wise fashion is not suitable for the cell counting problem for two reasons: First of all, the annotations tend to miss the cell centers by a few pixels, resulting in inaccurate ground truth masks. Precision and recall would then be biased - the scores would be less than perfect even if the predicted mask is exactly covering the cell. Secondly, precision and recall are only well-defined for the (in our case, binary) classification problem. Training the model with distance maps does not allow to view the task as such. Thus, a task-specific redefinition of precision and recall is required.

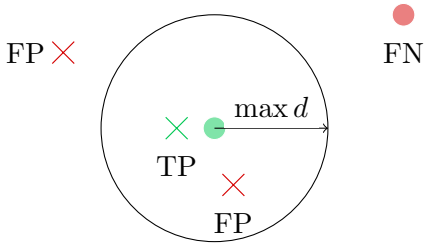


Figure 4: Definition of true positives (TP, green cross), false positives (FP, red crosses) and false negatives (FN, red circle) in the neighborhood of an annotated cell (green circle) for the cell counting task.

[4]https://scikit-image.org/

Figure 4 visualizes our definition of what is considered a detected cell. A detection within d pixels of an annotated cell center is marked as true positive (TP). In case the cell detector produces additional center predictions for the same cell, those spurious detections are considered false positives (FP). Similarly, a detection outside of any neighborhood of an annotated cell is also considered a false positive. Annotated cells with no matching prediction are counted as false negatives (FN). With these definitions, one can compute precision, recall, and F1-score as usual.

3.2 Cell Covered Area

We approach this problem in two steps. First, generate the annotations from bright-field microscopic images as the ground truth annotations are not available. Second, use state-of-the-art models U-Net and LinkNet to segment the lens-free images using the generated annotations from step 1 as the ground truth as described in Section 2.2.

3.2.1 Ground truth annotation generation

There are several tools available for segmenting bright-field microscopic images. The tool from the paper [2] is used with a few parameters modified suitable for our problem. Masks for the denoised bright-field images are obtained using the tool. The result is the segmented masks for cells that aren't split as our focus is to identify clusters of cells rather than individual cells. Obtained masks are processed further as the contrast patterns in the bright-field images were not captured. These patterns appear to be cells that are not accounted in the first step of annotation generation. This can be observed in the Figure 5 from the first two images.

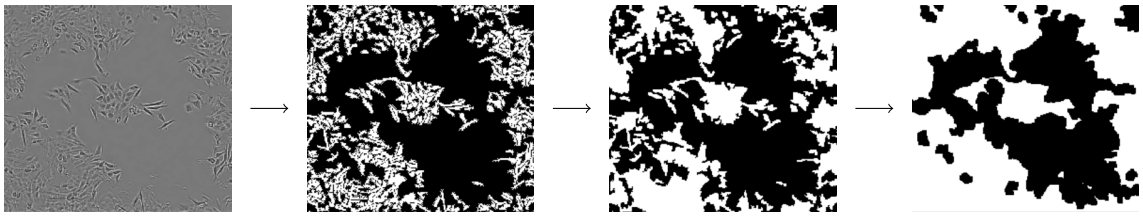


Figure 5: Illustration of ground truth annotation generation from a bright-field image. The image is segmented using a tool and cell clusters are obtained by series of processing steps.

To obtain the first stage mask, the mask from the tool is dilated with (2,2) elliptical kernel and external contours are identified and filled. To capture the contrast patterns, we worked on filling the non cell region (that is holes) by taking advantage of the smooth pixel value observed in the hole region in denoised bright-field image. Potential holes are chosen from the peak histogram pixel value of the bright-field image corresponding to the hole region in the first stage mask. In the chosen hole region, only the connected components with area greater than 500 pixels are considered to be actual holes. Thus, we reduce the hole region from the first stage mask by this process. Figure 6 depicts the overlay of the final mask on the bright-field image.

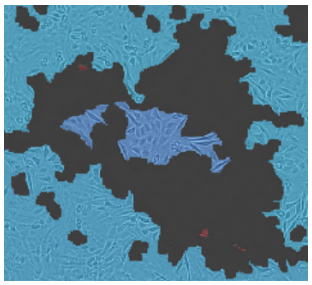


Figure 6: Overlay of final mask on the denoised bright-field image

3.2.2 Predictions from the Model output

The predictions from the fully convolutional neural network models are probabilistic maps as shown in 7. As a step towards final evaluation of the model results, these predictions need to be converted to binary masks. To achieve this, a Gaussian blur followed by smoothing is first applied to the predictions from the models. A threshold of 0.5 is then applied to the maximum pixel value found in the image. A pixel value higher than 0.5 of this maximum is considered as a cell as shown in Figure 7.

Prediction from model Processed prediction

Figure 7: First picture is the probabilistic prediction map obtained from the model and the second is obtained after applying post processing and threshold.

3.2.3 Metrics for Model Evaluation

It is tricky to decide on the metric for cell covered area detection as there are discrepancies observed in the bright-field and lens-free images as mentioned in Section 2.2. There are some instances where the cells appearing in bright-field are not seen in lens-free and vice versa, and thus we cannot rely only on Precision or Recall scores. So, we consider F1 score which takes into consideration both Precision and Recall and also Mean Squared Error (MSE) that is computed on the area coverage in the ground truth annotations and that in the predictions. Considering MSE alone for evaluation would be inaccurate as well, since some cells might be incorrectly segmented at some parts of the image (False positives) and this would be compensated by portions of the image where the model failed to detect cells (False Negatives). Thus, F1 score in combination with Mean Squared Error that conveys the variation in the overall area coverage is used as a measure for performance of cell covered area detection task.

4 Results

4.1 Lens-free Cell Counting

We evaluate the performance on the different ground truths we generated in section 3.1.1 using different loss functions. The Euclidean proximity map significantly improves the performance compared to the binary segmentation mask as a ground truth choice. The LinkNet achieves the best results while being the fastest method as well. The prediction of the LinkNet and local maxima extraction takes 0.13 seconds on an NVIDIA Quadro P5000 for a microscopy image with 1600x1200 pixels, which is $4x$ faster than using the U-Net architecture. We use our definition of precision and recall from section 3.1.3, and the mean percentage counting error, i.e. the average relative error, as metrics for comparison. We choose the relative counting error over the absolute counting error to equally weight sparse and crowded microscopy images and increase the interpretability of results.

4.1.1 Experimental Setup

We train three different neural network architectures using the standard cross entropy, balanced cross entropy, and the dice loss (see Appendix A.1 for definitions): A fully convolutional network with ResNet-50 truncated at the 23rd layer as backend (see Appendix A.2), the U-Net [9] with 28x28 pixels in the lowest resolution, and the LinkNet [3] with 14x14 pixels in the lowest resolution. We choose the percentage of pixels with value zero as β for each ground truth type. For training, we use 96 images of our dataset and split the remaining images resulting in 14 evaluation and 28 test images. We train all models for 7000 steps (parameter updates) with a batch size of 32 using the Adam optimizer. While we use a learning rate of 1e$-$4 for ResNet and LinkNet, we train the U-Net with a learning rate of 1e$-$5, because higher learning rates result in unstable training for U-Net. Nevertheless, each model converged before the 7000 steps. After the training is complete, we apply non-maximum suppression with thresholds $\kappa = 0.1, 0.11, ..., 0.9$ to find the optimal threshold choice. We use our definition of precision and recall from section 3.1.3 with a maximum distance of 9 pixels for the evaluation of the models.

4.1.2 Evaluation

F1 score	Binary segmentation			$\alpha = 4,\ d_{max} = 24$			$\alpha = 5,\ d_{max} = 16$		
	CE	BCE	DL	CE	BCE	DL	CE	BCE	DL
ResNet	0.8914	0.8854	0.7960	0.9644	0.9647	0.9632	0.9637	0.9635	0.9670
U-Net	0.9402	0.9068	0.9601	0.9511	0.9592	0.9609	0.9229	0.9620	0.9644
LinkNet	0.8921	0.8901	0.8168	0.9687	0.9694	0.9695	0.9677	0.9690	0.9677

Table 1: F1 scores for ResNet, U-Net and LinkNet trained on binary segmentation and Euclidean proximity maps with the three loss choices, cross entropy (ce), balanced cross entropy (bce) and dice loss (dl) as defined in Appendix A.1. We calculate the optimal threshold on the evaluation dataset and present the F1 score on the test images.

In Table 1 we analyze the F1 scores of the different neural network architectures using the different ground truths from Section 3.1.1. Further, we compare different loss choices: cross entropy, balanced cross entropy and dice loss. Using the euclidean ground truth maps shows a significant improvement over the binary segmentation mask. The U-Net and ResNet achieve their best F1 scores (0.9644 and 0.9670) trained with dice loss on the euclidean distance map with specifications $\alpha = 5, d_{max} = 16$. For the LinkNet architecture the euclidean distance map with $\alpha = 4, d_{max} = 24$ worked best. While the LinkNet outperforms the other models on almost all specifications, it is far more crucial to choose an appropriate ground truth mask and loss function than the most complex model.

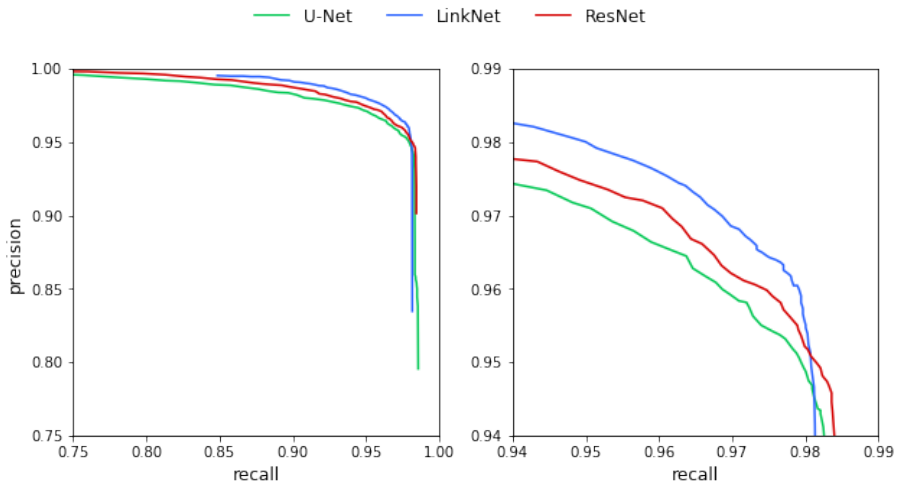


Figure 8: We compute the precision-recall curve by varying the threshold κ. For each architecture we use the best ground truth and loss specification, i.e. dice loss with $\alpha = 5, d_{max} = 16$ for ResNet and U-Net and $\alpha = 4, d_{max} = 24$ for the LinkNet architecture. The right plot is s zoomed-in version of the left one.

For further analysis of the best specifications for U-net, ResNet, and LinkNet, we compare the precision-recall curves. In Figure 8, we compute precision and recall by varying the threshold κ. Higher thresholds result in higher precision and lower recall and vice versa. This allows to fine-tune for better precision or recall performance. The right plot is a zoomed-in version of the left one and shows that usually, the LinkNet achieves higher precision with the same recall than the other two models. Only if, very high recall scores are required the ResNet outperforms the LinkNet. As mentioned in section 3.1.3, we propose to use a combination of F1 score and counting error for the task of cell counting. In Table 2, we compute the mean percentage counting error corresponding to Table 1. Again, the LinkNet is the winner with a percentage counting error of 2.13%. The counting error is usually lower than the proposed error by precision and recall because false positives and false negatives can cancel out each other.
In Figure 9 we analyze the predicted score maps of the LinkNet trained with dice loss on binary segmentation and Euclidean proximity maps with $\alpha = 4$, $d_{max} = 24$ and $\alpha = 5$, $d_{max} = 16$. The produced probability maps for binary segmentation exhibit multiple local maxima with large distances to the cell center. For the two overlapping cells in the second row we see one local maximum in the middle of the cells and therefore only one cell detection. The predicted proximity have a local unimodal structure and

%-error	Binary segmentation			$\alpha = 4,\ d_{max} = 24$			$\alpha = 5,\ d_{max} = 16$		
	CE	BCE	DL	CE	BCE	DL	CE	BCE	DL
ResNet	9.92%	12.51%	44.28%	2.68%	2.92%	2.37%	2.44%	2.24%	2.34%
U-Net	2.75%	8.12%	2.54%	3.14%	3.10%	2.75%	9.37%	3.04%	2.86%
LinkNet	16.06%	17.76%	38.57%	2.12%	2.43%	2.13%	2.53%	2.35%	2.63%

Table 2: Mean percentage counting error for ResNet, U-Net and LinkNet trained on binary segmentation and Euclidean proximity maps with the three loss choices, cross entropy (ce), balanced cross entropy (bce) and dice loss (dl) as defined in Appendix A.1.

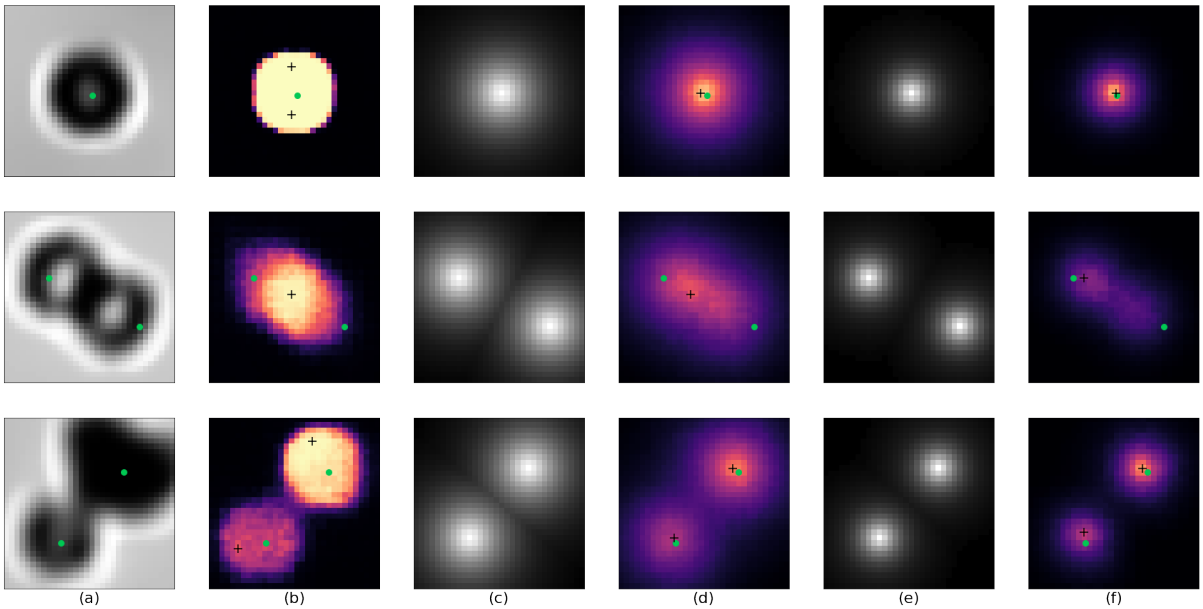


Figure 9: Comparison of estimated score maps for binary segmentation and Euclidean proximity maps. (a) Small patches of raw images from the lens-free microscopy. Green dots indicate annotated cell center. (b) Probability maps predicted by model trained on binary segmentation with local maxima (black markers). (c) The ground truth proximity map with specifications $\alpha = 4,\ d_{max} = 24$ and (d) being the predicted score map. (e) The ground truth proximity map for $\alpha = 5,\ d_{max} = 16$ with predictions (f).

therefore unique local maxima that match the cell centers much better, for the second row even better than the annotated coordinates.

4.2 Cell Covered Area

In this section, we evaluate the annotations generated using the proposed method, why we use fully convolutional neural networks and not deterministic image processing based segmentation approaches and the performance of neural networks for cell covered area detection.

4.2.1 Validation of generated annotations

To validate how good the generated annotations are, we chose one image from HuH7, HeLa and 3T3 each, in a way that the cells are neither sparsely covered nor densely covered, so that it represents the dataset well. We then manually segmented the chosen images and compared it with the generated annotations. The example of HuH7 cell image segmentation is illustrated in Figure 10.

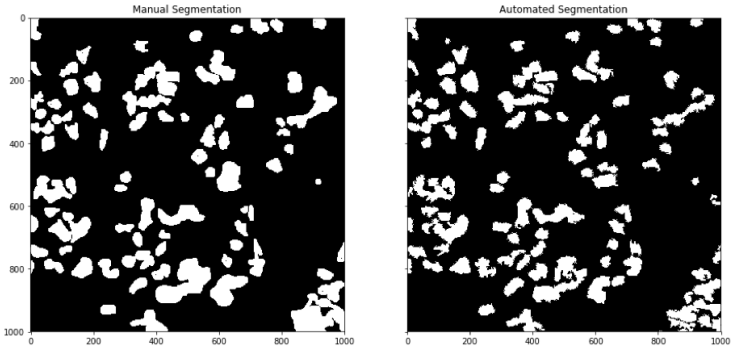


Figure 10: Manually segmented and Generated Annotations of HuH7 bright-field image.

The accuracy of generated annotations in comparison with manual segmentation is 0.934, recall and precision are 0.786 and 0.856 respectively. The area covered estimations are 0.188 and 0.173 for manual and generated annotations respectively. The low recall score is due to the generous segmentation of the cells manually as humans are not good at visually identifying the boundaries precisely to the pixel scale. Validation on other chosen images are observed to be inline with the illustrated one.

4.2.2 Evaluation of non neural net based approach

Before diving into neural network based models, we evaluate the deterministic image processing based segmentation approach. We formalized a simple segmentation method tuned well on HeLa and used the same method on other cell types as well. It is illustrated in the Figure 11 which shows area coverage plots for HeLa and A549. We made two observations - first, the segmentation approach couldn't be tuned well enough to identify the original cell clusters as the lens-free images have larger cell size compared to the bright-field images and there is no fixed correspondence between them. So, this method overestimates the cell coverage always as seen in the plots. It matches the original annotation in high cell density regime as the entire image is covered with cells. Second, the simple segmentation performs very badly on other cell types as the method is fine tuned for HeLa. Significant drift is seen between the original annotation and the masks from simple segmentation in the right plot. It can also be observed that the F1 scores are not consistent across different cell types. For example, F1 score for HeLa is around 0.81 and that of A549 is 0.62. Therefore, it is a challenge to generalize the simple segmentation approach for obtaining best results irrespective of the cell types. In general, most of the segmentation done using image processing techniques requires specific parameter setup for different types of images. This led us to consider neural network based models to learn the correspondence between lens-free and bright-field images.

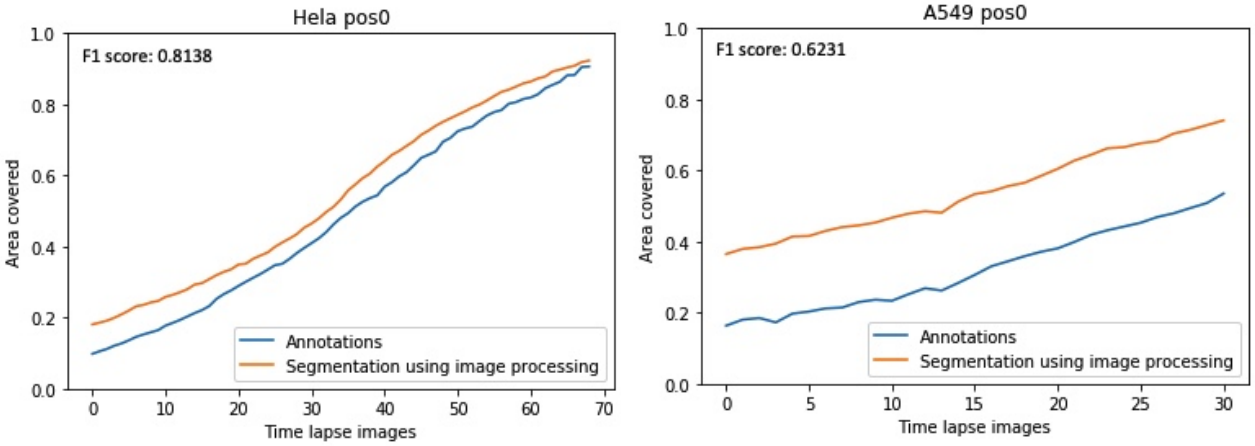


Figure 11: Area coverage graphs for annotations and simple segmentation for HeLa and A549 along with their F1 scores.

4.2.3 Experimental Setup

Fully convolutional network architectures U-Net [9] and LinkNet [3] are trained with three losses namely, binary cross-entropy, dice loss and a combination of binary cross-entropy and dice losses (Appendix A.1). All the six models are trained for 10000 steps with a batch size of 32 using the Adam optimizer. The dataset is randomly split to train, evaluation and test set each with 205, 45 and 45 images respectively. Patches of 224×224 are extracted from each image and is used for training the model. Same train, evaluation and test set is used for evaluating all the models. The learning rates used for U-Net is that of 1e-5 and for LinkNet with a higher rate of 1e-3. We choose a higher learning rate for LinkNet as the loss did not converge with lower rates even with 50 epochs. With the above setup, all the models converged. After training each model, the predictions on the test set which are probabilistic maps are obtained. The obtained predictions are processed as mentioned in Section 3.2.2 to get binary prediction masks. We use F1 score with MSE based on area coverage to evaluate the models and decide on the best one.

4.2.4 Evaluation

Models	BCE		BCE + DICE		DICE	
	F1	MSE	F1	MSE	F1	MSE
LinkNet	0.8247	0.001434	0.8363	0.001566	0.8270	0.001594
U-Net	0.7042	0.051104	0.6988	0.054393	0.8129	0.001222

Table 3: F1 Score and MSE computed on testset for two models: LinkNet and U-Net for each of the three losses namely Binary Cross Entropy (bce), combination of Binary Cross Entropy and Dice loss (bce+dl) and Dice Loss (dl) as defined in Appendix A.1.

In Table 3, we measure the F1 scores and MSE for different model-loss combinations on the testset. We observe that the LinkNet with bce+dl loss and U-Net with dice loss

perform the best with F1 scores 0.8363 and 0.8129, and MSE 0.001566 and 0.001222 respectively. Between the best models in two different architectures, we choose LinkNet over U-Net based on the F1 score although the MSE for U-Net is lesser than LinkNet as F1 score shows the accuracy of the models as argued in the Section 3.2.3. We observe that U-Net with bce and bce+dl losses perform poorly with F1 scores close to 0.70 as opposed to 0.84 in the LinkNet best model. This is explained by the fact that U-Net predictions are quite conservative and have very low confidence in terms of the prediction probabilities. Figure 12 shows the F1 scores obtained on LinkNet and U-Net.

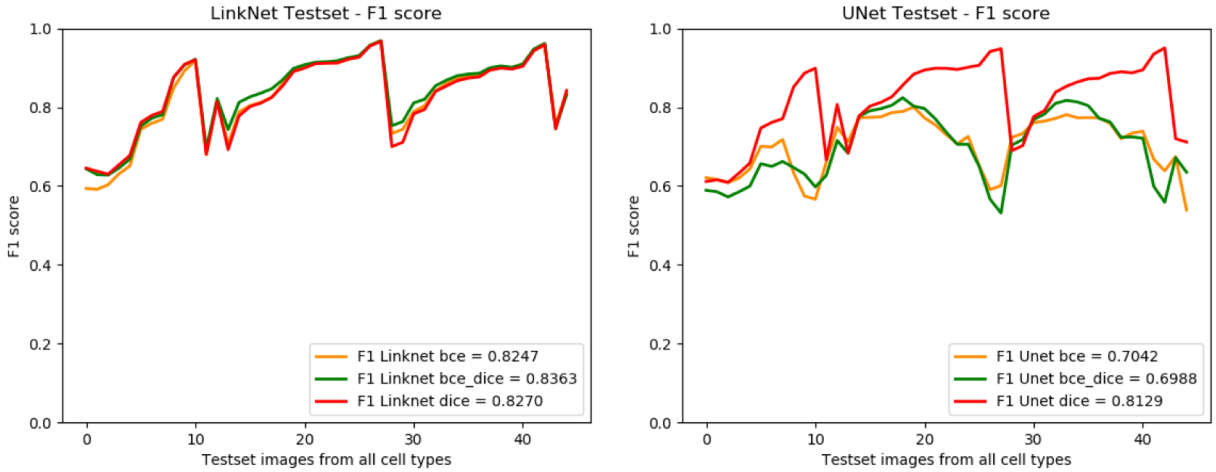


Figure 12: F1 score plot on testset for LinkNet and U-Net models.

Since we have time lapse microscopic images for each cell type except HuH7, we compare the area coverage estimations between the ground truth annotations and the predictions from the best model established, i.e. LinkNet with bce+dl loss.

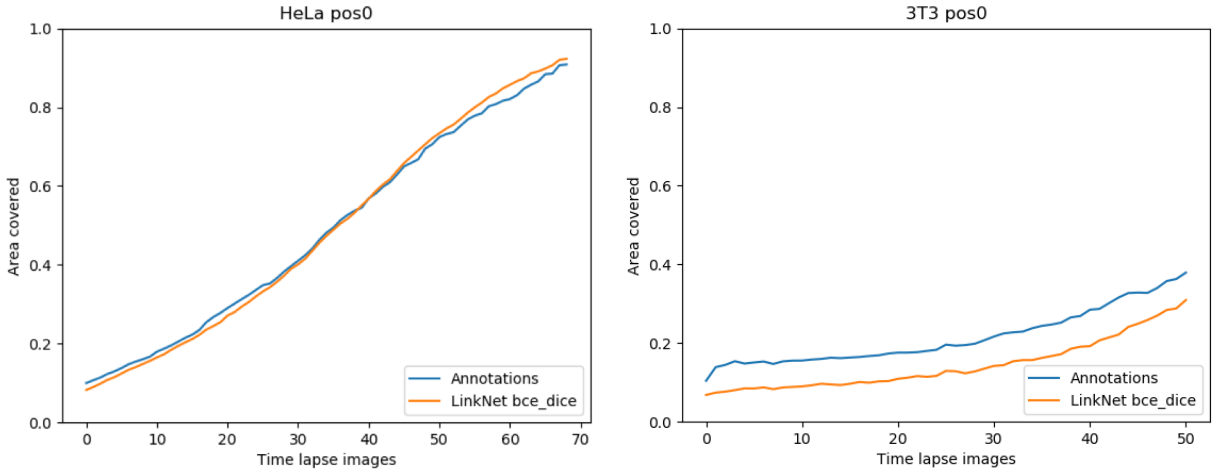


Figure 13: Representative plots of area coverage on time lapse sequences

Figure 13 shows representative plots as both the time lapse sequences from HeLa, one of the two sequences from 3T3 and the sequence from A549 pretty much resemble the plot

with title 'HeLa pos0' and the only sequence where we observed significant variance is with one other sequence of 3T3 as shown in the plot with title '3T3 pos0'. The significant shift is due to the inclusion of extra portions as cells while generating annotations as we observe lot of small spots which are not seen in lens-free images. Thus, the area predicted is lesser than the ground truth for this particular sequence.

Training with raw lens-free images As mentioned earlier, the above experiments are performed by training the network with denoised lens-free images. To measure the value of denoising the input as it takes considerable amount of time, we trained the best model with raw lens-free images as input. The F1 score and area coverage plots on testset using LinkNet with bce+dl loss, with denoised and raw lens-free images as input is as shown in Figure 14. The F1 scores for the models trained with denoised and raw images are 0.8363 and 0.8232 respectively and the MSE for area coverage are 0.001566 and 0.001708. Although, these scores are close, it can be observed that a model trained with denoised images with reduced interference patterns perform slightly better than the model that is trained with raw images. It should also be noted that denoising an image is an additional step in the data processing pipeline and in turn be an overhead.

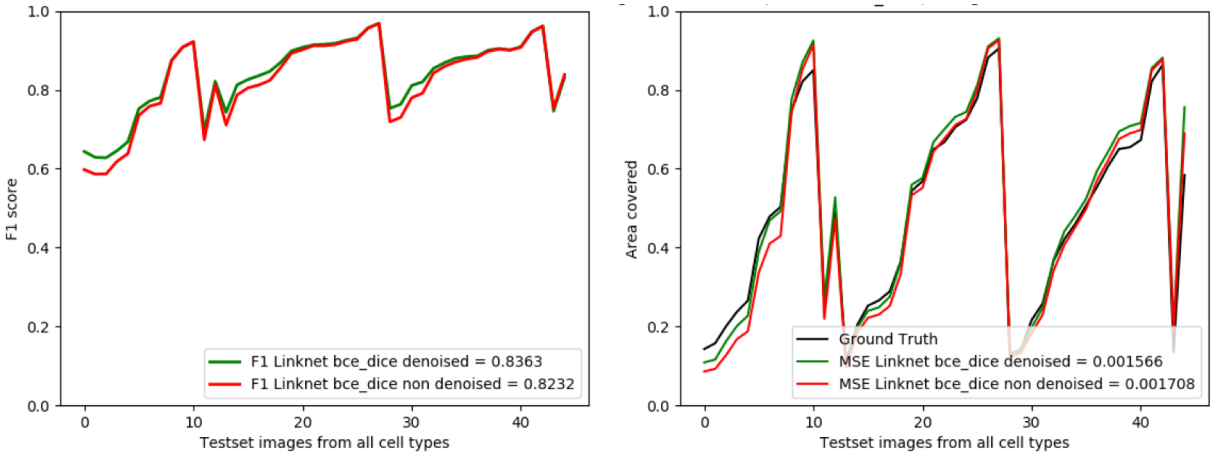


Figure 14: F1 scores and area coverage on testset using LinkNet models trained with denoised and raw lensfree images

5 Conclusion

We have described in detail the several approaches taken to solve the two tasks at hand in the previous sections namely Cell Counting and Cell-covered area detection. Cell counting from lens-free images involved localizing individual cells. For accomplishing this, we proposed a distance transform for the ground truth and two stage process to locate the cells where we train the model in the first stage and use the local maxima to detect the cell centers in the second stage. We experimented with the mentioned deep neural network architectures and established that the robust model is LinkNet with dice loss. This model achieved an F1 score of 0.97 and a relative counting error of 2.13% which is as good as manual counting.

Detecting cell-covered area from lens-free images comprised of generating annotations from bright-field images and determining a suitable model that learns the correspondence between bright-field and lens-free images with high accuracy. We establish from the experiments that LinkNet with bce+dice loss performs the best with an F1 score of 0.836. We suggest some improvements in these methods that could lead to better results. Firstly, refine the annotations to minimize irregularities in the segmentation. This includes accurate distinction between a cell and background with noise. Secondly, reduce the discrepancies between the reference bright-field and lens-free images that occur in the time lapse sequences in the dataset as discussed in Section 2.2. Lastly, further research can be done on Long Short Term Memory (LSTM) with the U-Net [1] or LinkNet as the dataset contains temporal microscopic images.

References

[1] Assaf Arbelle and Tammy Riklin Raviv. "Microscopy cell segmentation via convolutional LSTM networks". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE. 2019, pp. 1008–1012.

[2] Felix Buggenthin et al. "An automatic method for robust and fast cell detection in bright field images from high-throughput microscopy". In: *BMC bioinformatics* 14.1 (2013), p. 297.

[3] Abhishek Chaurasia and Eugenio Culurciello. "LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation". In: *CoRR* abs/1707.03718 (2017). arXiv: 1707.03718. URL: http://arxiv.org/abs/1707.03718.

[4] Alon Greenbaum et al. "Wide-field computational imaging of pathology slides using lens-free on-chip microscopy". In: *Science Translational Medicine* 6.267 (2014), 267ra175–267ra175. ISSN: 1946-6234. DOI: 10.1126/scitranslmed.3009850.

[5] K. He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[6] Philipp Kainz et al. "You Should Use Regression to Detect Cells". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 276–283. ISBN: 978-3-319-24574-4.

[7] Aisha Khan, Stephen Gould, and Mathieu Salzmann. "Deep Convolutional Neural Networks for Human Embryonic Cell Counting". In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Gang Hua and Hervé Jégou. Cham: Springer International Publishing, 2016, pp. 339–348.

[8] Markus Rempfler et al. "Tracing cell lineages in videos of lens-free microscopy". In: *Medical Image Analysis* 48 (2018), pp. 147–161. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2018.05.009.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[10] A. Sironi, V. Lepetit, and P. Fua. "Multiscale Centerline Detection by Learning a Scale-Space Distance Transform". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2697–2704. DOI: 10.1109/CVPR.2014.351.

[11] Weidi Xie, J. Alison Noble, and Andrew Zisserman. "Microscopy cell counting and detection with fully convolutional regression networks". In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 6.3 (2018), pp. 283–292. DOI: 10.1080/21681163.2016.1149104.

A Appendix

A.1 Losses

Let $P \in \mathbb{R}^{m \times n}$ be the ground truth matrix and $Q \in \mathbb{R}^{m \times n}$ the predicted score map, we then define cross entropy of P and Q as:

$$CE(P, Q) = -\sum_{i}^{m} \sum_{j}^{n} (p_{ij} \log q_{ij} + (1 - p_{ij}) \log (1 - q_{ij})).$$

To handle class imbalance we define the balanced cross entropy via:

$$BCE(P, Q) = -\sum_{i}^{m} \sum_{j}^{n} (\beta p_{ij} \log q_{ij} + (1 - \beta)(1 - p_{ij}) \log (1 - q_{ij})),$$

where $\beta \in]0, 1[$ defines the balance between foreground and background. We further define the dice loss as:

$$DL(P, Q) = -\frac{2\langle P, Q \rangle}{\|P\|_F^2 + \|Q\|_F^2} = -\frac{\sum_i^m \sum_j^n p_{ij} q_{ij}}{\sum_i^m \sum_j^n p_{ij}^2 + \sum_i^m \sum_j^n q_{ij}^2},$$

where $\|\cdot\|_F$ defines the Frobenius norm.
We define the combination of BCE and DICE as:

$$BCE\_DICE(P, Q) = BCE(P, Q) + DL(P, Q).$$

A.2 Network Architectures

ResNet-based Fully-convolutional Architecture

The architecture uses a ResNet-50 as feature extractor and is closely related to the network proposed in [8] (see Figure 15). First, the image is convolved with 64 filters of dimension 7×7 and a stride of 2. Followed by a strided max-pooling operation and 7 residual blocks to produce an output of $28 \times 28 \times 512$ as the high level features of an image. To up-sample back to the spatial dimension of the original image patch, a transposed convolution of filter size 8×8 and stride 8 is applied. ReLU is used as activation function after all convolutions except for the transposed convolution, where a sigmoid activation is applied.

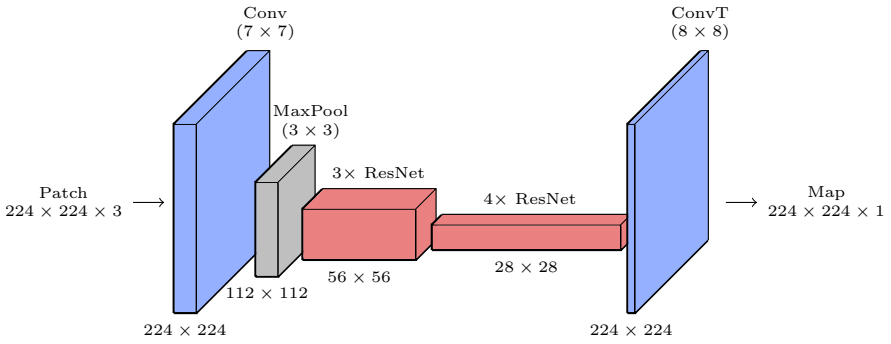


Figure 15: Fully-convolutional cell detector architecture with ResNet backbone.