UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Kaspar Hollo

Exploring the Value of Weakly-Supervised Deep Learning Approaches for Artefact Segmentation in Brightfield Microscopy Images

Master's thesis (30 ECTS)

Supervisors: Mohammed Ali, MSc
Dmytro Fishman, MSc

Tartu 2021

Exploring the Value of Weakly-Supervised Deep Learning Approaches for Artefact Segmentation in Brightfield Microscopy Images

Abstract:

Brightfield microscopy is of great importance as it offers researchers a relatively simple way to quantify cellular experiments. However, brightfield images often contain a variety of artefacts that should be segmented and thereafter neutralized so that they would not affect the quantitative measurements of cellular experiments. While fully-supervised deep learning models offer state-of-the-art performance in most segmentation tasks in computer vision, it is laborious to acquire the pixel-level labels needed to train these models. Alternatively, segmentation tasks can also be solved using more time- and cost-effective weakly-supervised deep learning models that use image-level labels for training. In this thesis, we compare the performances of fully- (e.g., U-Net) and weakly-supervised approaches (e.g., Score-CAM) to determine whether weakly-supervised approaches could be used as a cheaper but still well-performing solution for segmenting artefacts in brightfield images. Six separate experiments with various fully- and weakly-supervised approaches, image datasets and method ensembles are carried out. The results of the experiments showed that with the number of images and labels currently available, none of the weakly-supervised approaches were able to replicate the performance of the baseline fully-supervised approach. However, some of the weakly supervised approaches, like the combined Score-CAM and U-Net approach, showed promising segmentation results. Moreover, the same approach also showed better generalizability on an unseen dataset than the baseline fully-supervised approach. Future work is required to find the amount of weak supervision signal needed to match the performance of the fully-supervised approaches.

Keywords:

deep learning, neural networks, weakly-supervised learning, brightfield microscopy, artefacts

CERCS:

T111 - Imaging, image processing; P176 - Artificial intelligence; B110 - Bioinformatics, medical informatics, biomathematics biometrics

Nõrgalt juhendatud süvaõppe mudelite tõhusus helevälja mikroskoopiapiltidelt anomaaliate segmenteerimisel

Lühikokkuvõte:

Helevälja mikroskoopia on oluline, sest see pakub teadlastele suhteliselt lihtsa viisi rakukatsete kvantifitseerimiseks. Helevälja mikroskoopiapiltidel leidub aga sageli erinevaid anomaaliaid, mis tuleb esmalt segmenteerida ja seejärel neutraliseerida, et need ei mõjutaks rakukatsete kvantitatiivseid mõõtmisi. Kuigi tugevalt juhendatud süvaõppemudelid pakuvad paljudes tehisnägemisega seotud segmenteerimisülesannetes tipptasemel tulemusi, nõuab selliste mudelite treenimiseks vajalike pikslitäpsusega märgenduste hankimine keerulist ja ajamahukat tööd. Segmenteerimisülesandeid on võimalik aga lahendada ka aja- ja kulutõhusamate nõrgalt juhendatud süvaõppemudelitega, mida treenitakse pilditäpsusega märgenduste abil. Selles lõputöös võrreldakse tugevalt juhendatud meetodite (nt U-Net) ja nõrgalt juhendatud meetodite (nt Score-CAM) segmenteerimistulemusi, et teha kindlaks, kas nõrgalt juhendatud meetodeid võiks kasutada odavama, kuid siiski hästi toimiva lahendusena anomaaliate segmenteerimiseks helevälja mikroskoopiapiltidel. Lõputöö raames viiakse läbi kuus eraldi katset täielikult ja nõrgalt juhendatud meetodite, pildiandmekogumite ja erinevate meetodite kombinatsioonidega. Katsete tulemused näitasid, et praegu saadaolevate piltide ja märgendite arvuga ei suutnud ükski nõrgalt juhendatud meetod korrata tugevalt juhendatud meetodite segmenteerimistulemusi. Sellegi poolest näitasid mõned nõrgalt juhendatud meetodid, nagu näiteks Score-CAM'i ja U-Net'i kombineeritud meetod, paljutõotavaid segmenteerimistulemusi. Nimetatud meetod suutis ka seninägemata pildiandmekogu puhul üldistada paremini kui standardiks määratud tugevalt juhendatud meetod. Selleks, et määrata nõrgalt juhendatud meetodite puhul neile vajaliku andmekogu suurust, mis võimaldaks korrata juhendatud meetodite segmenteerimistulemusi, tuleb teha nõrgalt juhendatud meetoditega lisakatseid.

Võtmesõnad:

süvaõpe, tehisnärvivõrgud, nõrgalt juhendatud õpe, helevälja mikroskoopia, anomaaliad

CERCS:

T111 – Pilditehnika; P176 – Tehisintellekt; B110 - Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika

Table of Contents

1 Introduction

Brightfield microscopy is a light microscopy technique in which the image is created when the dense areas of the specimen have absorbed, scattered, or deflected the illuminating light directed upon the specimen (Mokobi, 2020; Wang & Fang, 2012). It is an integral part of biomedical research as brightfield images can be used to quantify cellular experiments, e.g., brightfield images have been used to examine the growth and movement of cells in healing wounds (Zordan et al., 2011).

However, brightfield images are prone to exhibit different artefacts (e.g., dust particles, bacterial colonies). It is clear that artefacts may affect quantitative downstream analysis by, for example, overlaying and thus hiding a substantial number of cells in the image. Therefore, it is essential to segment (i.e., precisely localize) and thereafter neutralize these artefactual regions in the brightfield images before making any quantitative measurements to get accurate results. To the extent of our knowledge, this problem has largely been overlooked as we did not find any publications in which an effort had been put into segmenting or neutralizing artefacts in brightfield images.

Chan et al. claim in their work (Chan et al., 2020) that state-of-the-art performance in most segmentation tasks is achieved with fully-supervised deep learning models. Fully-supervised learning defines models that are trained on pixel-level labels to solve a segmentation task. Chan et al. add that the need for pixel-level labels is also the most significant drawback of fully-supervised models as manual labelling at the pixel level requires considerable time and effort from the annotator.

The authors of the same work propose weakly-supervised learning as a possible alternative for the segmentation tasks. Namely, weakly-supervised learning requires lower-quality data like image-level labels (i.e., a class tag for each image) to train segmentation models. Image-level labels are much easier to acquire and thus make the training process of a segmentation model cheaper. Research also shows that weakly-supervised deep learning models have been successfully applied to localize artefacts (i.e., details that deviate from normality) in textured surfaces (Defard et al., 2020; Yi & Yoon, 2020; Bergmann et al., 2019). For the abovementioned reasons, we found it worthwhile to investigate if weakly-supervised segmentation models could be used as an alternative to fully-supervised segmentation models to segment artefacts in brightfield images.

This thesis investigates whether weakly-supervised approaches can segment artefactual regions in brightfield images with performance similar to those of fully-supervised approaches.

1.1 Contributions

In this thesis, we have the following contributions:

- We test the artefact segmentation capabilities of two fully-supervised approaches, U-Net and YOLOv5 + U-Net. The superior approach is set as the baseline fully-supervised approach.

- We investigate the potential of segmenting artefacts with a weakly-supervised class activation mapping-based approach called Score-CAM.
- We compare the Score-CAM approach to state-of-the-art artefact segmentation approaches.
- We improve the Score-CAM approach even further by combining it with the fully-supervised U-Net approach.
- We investigate what performance level of the baseline fully-supervised approach can be reached with Score-CAM and Score-CAM + U-Net by varying dataset sizes.
- We ensemble the predictions of the weakly-supervised approaches to see if this results in better performance. The best combined prediction is compared to the baseline fully-supervised approach.
- We test the generalizability of the baseline fully-supervised approach and some of the weakly-supervised approaches by applying them out-of-the-box on a new dataset.

1.2 Outline

Background provides a brief overview of microscopy, artefacts, deep learning and related work in literature.

Datasets and annotations describes the datasets and annotations used to train and test all of the approaches.

Methodology lists the investigated fully- and weakly-supervised segmentation approaches and describes them in more detail.

Experiments and results describes the conducted experiments and their results. A short discussion about the results follows each experiment.

Conclusion summarizes the thesis, lists the limitations and discusses the future directions of this work.

References lists all of the used literature sources.

Appendix includes the license.

2 Background

This chapter introduces the essential concepts needed to understand the thesis. It includes a brief insight into deep learning, microscopy, artefacts and evaluation metrics. Furthermore, it is explored what has already been done regarding artefact segmentation. The background chapter also paves the way for the methodology chapter in which the approaches used in the conducted experiments are explained in more detail.

2.1 Artificial Neural Networks and Deep Learning

Artificial Neural Network (ANN) and deep learning are concepts that are briefly described in this section since the methods used to solve the artefact segmentation task are based on these concepts.

Deep learning is the general definition of machine learning algorithms that are based on ANNs (Marr, 2018). As the field of deep learning is vast, we consider a deep dive into its mechanics to be outside of the scope of this thesis. Hence, in this chapter, deep learning related main principles will only be reviewed on a superficial level to support further discussions.

An ANN is a nonlinear function approximator that essentially tries to mimic how a nervous system processes information (Bilal, 2018). Like in the human nervous system, information is processed in interconnected neurons and then transmitted based on weight parameters that determine the connection strength of the interconnected neurons. ANNs can have different structures, but the simplest example of an ANN (Figure 1) consists of fully-connected neurons in three layers:

- Input layer – the initial data is inserted into the network through the input layer. For example, if the initial data is an image, each pixel value of the image is inserted into a separate neuron of the input layer.
- Hidden layer – in this layer, each neuron sums the incoming weighted signals from the neurons in the previous layer and processes the summed signal with a thresholding function (i.e., the activation function). Depending on the resulting value, the neuron may pass the processed signal to the neurons in the following layer.
- Output layer – in this layer, the network's prediction is returned. The number of neurons in the layer depends on the task at hand. For example, a task may be in the form of determining whether the inserted input image contains a cat or a dog. In that case, the output layer would have two neurons in the output layer. The first neuron would contain the probability that the image contains a cat and the other neuron the probability that the image contains a dog.

An ANN with multiple hidden layers is called a Deep Neural Network (DNN) (Bilal, 2018).

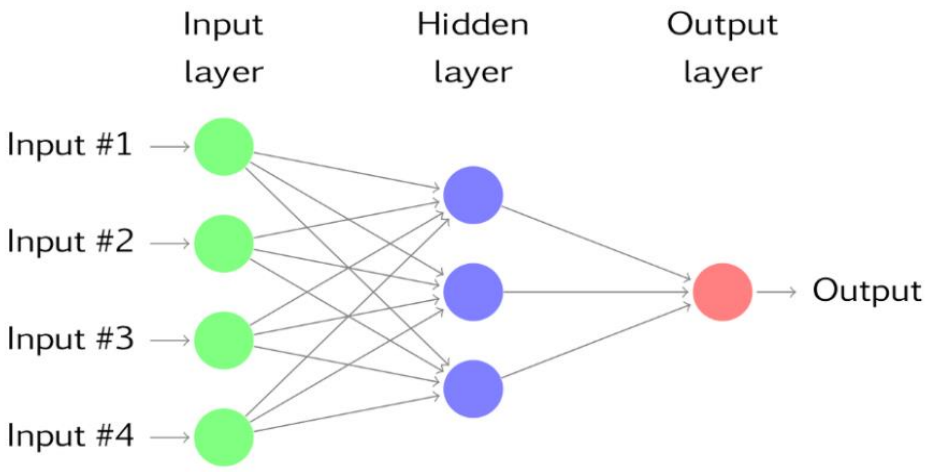


Figure 1. Fully-connected three layer artificial neural network. (Bilal, 2018)

ANNs can be used to solve all kinds of complicated prediction and classification tasks. However, the weights of the ANN must be adjusted before it can make accurate predictions based on the input data (Stanford University, 2015). The weights of an ANN are adjusted during a training process, in which the ANN is shown examples of known input data (e.g., images of cats and dogs) and their respective labels (e.g., cat and dog). The training process itself consists of two subsequently applied algorithms: forward pass and backward propagation (a.k.a. backpropagation). During the forward pass, the input data (e.g., an image of a cat) is fed to the ANN, a prediction is returned by the network (the probability that the image contains a cat), and the error between the prediction and the expected output is measured. Backpropagation computes the gradients of the error with respect to all of the weights in the ANN. Gradients are measures that show how the weights should be adjusted to minimize the error between the prediction and the expected output. Once the gradients are calculated, the weights are also adjusted. This particular training cycle is repeated multiple times to minimize the error between the prediction and the expected output.

2.2 Convolutional Neural Networks

The Convolutional Neural Network (CNN) is a type of ANN designed to analyse images (Brownlee, 2019). The convolutional layer is the key element making CNNs especially suitable for analysing images. The layer consists of matrices of learnable weights called filters that are applied to the input (e.g., an image, output of other convolutional layers) to recognize patterns, make sense of them, and extract features (e.g., lines, shapes, textures, objects) out of the input (Figure 2). Each filter in the convolutional layer extracts a different feature from the input.

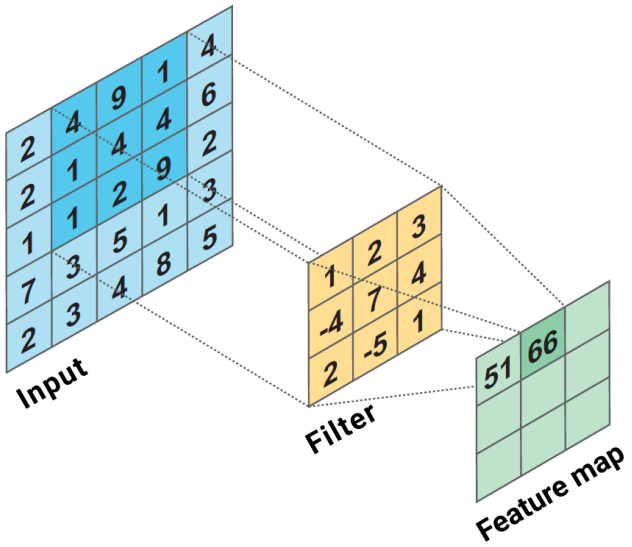


Figure 2. The filter slides horizontally and vertically over the input to extract features from it. At each step, the filter is applied to only a local region of the input that matches the filter's dimensions. This way, each local region of the input is inspected separately for the specific feature the filter is trying to extract. The extracted feature is put into a feature map. (Patel, 2019)

The filters are intentionally designed to be smaller than the input to reduce the number of needed learnable weights and so that the same filter could be applied across the entire input. As the filters are sequentially applied to the input, it can also be determined in which areas of the input does a specific feature reside (Figure 2). The extracted features of a convolutional layer are collected in feature maps (a.k.a. activation maps).

CNN architectures usually contain multiple convolutional layers, which allows the network to detect features of various abstraction levels (Figure 3). Namely, the convolutional layers that are at the beginning of the network will detect lower-level features (e.g., lines, edges). On the other hand, the convolutional layers further down the network will combine features from the first convolutional layers into more complex, higher-level features (e.g., noses, eyes, faces).

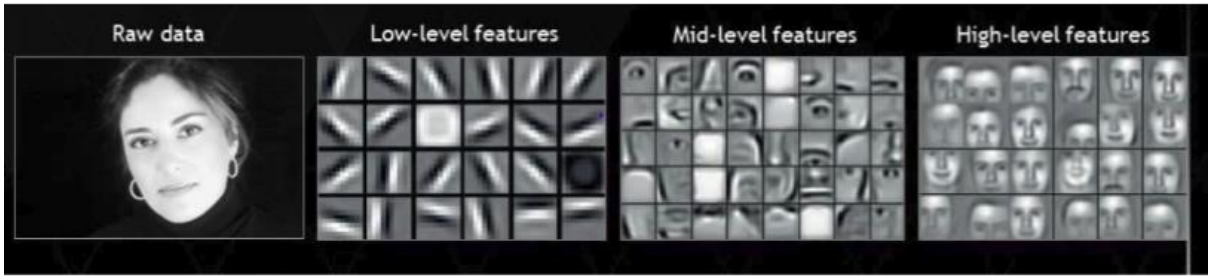


Figure 3. The convolutional layers of CNNs capture features depending on their position in the network. The lower level features are combined to make up the higher-level features. In this example, the lines and edges from the lower-level features are used to construct the eyes, noses and faces in the higher-level features. (Pallawi, 2019)

2.3 Computer Vision Tasks

Herein, we introduce the computer vision tasks that are being solved in this thesis using CNNs. This includes image classification, object detection and semantic segmentation.

Image Classification

Image classification is a task in which a class label is assigned to each input image (Figure 5 left panel). It is considered to be one of the most important computer vision problems - especially in medical image analysis, where the task is often to determine the presence of a disease. For example, different deep learning architectures have been used to diagnose skin cancer (Goyal et al., 2020), breast cancer (Gao et al., 2018), and Alzheimer's disease (Hosseini-Asl et al., 2016). Some of the most widely used deep learning architectures for the classification task include ResNet (He et al., 2016), VGG-16 (Simonyan & Zisserman, 2014), Inceptionv3 (Szegedy et al., 2015), and EfficientNet (Tan & Le, 2019).

Object Detection

Object detection is a computer vision task in which objects from a fixed set of classes are identified and positioned by imaginary bounding boxes within the input image (Figure 5 middle panel). Object detection related tasks also play a vital role in medical image analysis. Deep learning architectures can detect colon cancer in histology images (Sirinukunwattana et al., 2016), and polyps in screening colonoscopies (Urban et al., 2018). State-of-the-art approaches like YOLOv5 (Jocher et al., 2021), YOLOv4 (Bochkovskiy et al., 2020), and Faster R-CNN (Ren et al., 2015) are often used to tackle various object detection tasks.

Semantic Segmentation

Semantic segmentation is a computer vision task in which a class label is assigned to each pixel in the input image. None of the objects in the same class are being differentiated between (Figure 5 right panel). Multiple applications of semantic segmentation can be found in medical image analysis. For instance, the segmentation of skin lesions in dermoscopic images (Ünver & Ayan, 2019), and tumor regions in brain MRI (Kayalibay et al., 2017). Typical deep learning networks include U-Net (Ronneberger et al., 2015), U-Net++ (Zhou, Siddiquee et al., 2018), and SegNet (Badrinarayanan et al., 2015).

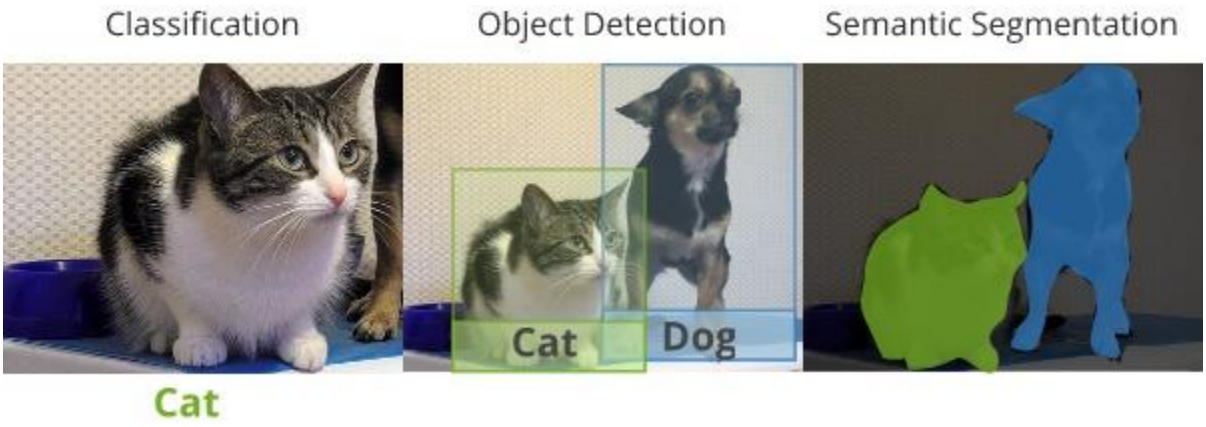


Figure 5. Classification, object detection and semantic segmentation are computer vision tasks that can be solved with deep learning. (Restrepo, 2017)

2.4 Class Activation Mapping

Although people have been able to analyse images using CNNs, the exact reasons behind specific predictions remained unclear. This made it difficult to interpret and debug CNNs and thus hindered their widespread application. In response, the deep learning community designed several approaches to improve the interpretability of CNNs. Here we will discuss one of the developed techniques, called Class Activation Mapping (CAM).

CAM is a CNN visualization algorithm used to offer insights on how a CNN model with millions of parameters makes its predictions (Zhou, Khosla et al., 2015). CAMs try to demystify or at least improve the interpretability of the inner works of CNNs by highlighting the regions or features in an image that are deemed more important by the network while making a prediction. The algorithm's output is a saliency map (a.k.a. heat map), which is an image where the pixel intensities represent the potential importance of the corresponding pixels in the original image to the decision making of the CNN (Ali, 2020; Figure 4). Popular CAM based algorithms include Grad-CAM (Selvaraju et al., 2019), Grad-CAM++ (Chattopadhay et al., 2018), and Score-CAM (Wang, Wang et al., 2019). Score-CAM is the algorithm that is being used in this thesis, and thus it is discussed in more detail in the methodology chapter.

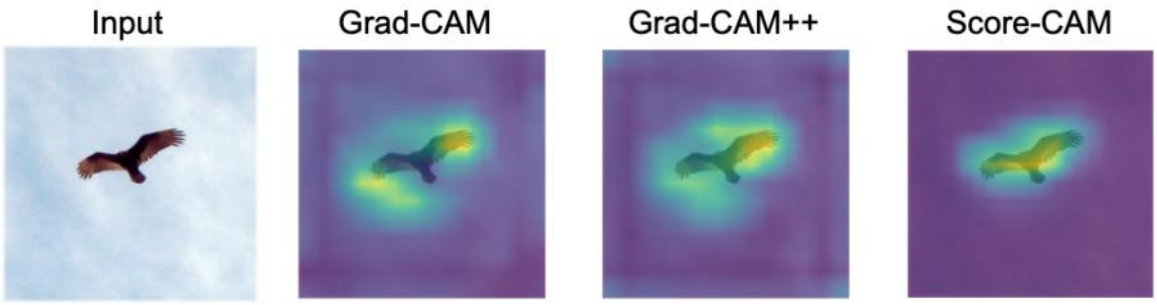


*Figure 4. The example saliency maps of Grad-CAM, Grad-CAM++, and Score-CAM are overlaid on the original image. (*Wang, Wang et al., 2019*)*

2.5 Fully- and Weakly-Supervised Learning

Fully-supervised learning is an approach that uses precise labels for the training of deep learning models. In semantic segmentation related tasks, pixel-level labels (i.e., the annotator assigns a class label to each pixel of the image) are used to train the models. It is the most commonly used method to deliver state-of-the-art performance in semantic segmentation (Chan et al., 2020). However, fully-supervised learning has also its downsides as the training of a complex but well-performing fully-supervised semantic segmentation model requires a vast amount of pixel-level labelled data.

Firstly, the pixel-level labelling of images is time-demanding. It is documented (Lin et al., 2014) that the annotation of an image in the MS COCO dataset took 10.1 minutes on average. In contrast, the image-level labelling (i.e., the annotator assigns a class tag to each image) of an image in the same dataset took only 4.1 seconds on average.

Secondly, in particular fields like cellular biology, any labelling of images must be done by field experts. The inclusion of experts makes the pixel-level labelling process an expensive undertaking, especially in fields where objects have indistinct borders. For example, the pixel-level labelling of borders in histopathology images has even caused disagreements between experts, thereby prolonging the labelling process even further (Xu et al., 2017).

It is possible to alleviate these aforementioned problems by using an alternative approach called weakly-supervised learning. Weakly-supervised learning is an approach where the deep learning models are trained on lower quality data (Roh et al., 2019). In semantic segmentation-related tasks, lower quality data could be in the form of image-level labels that are acquired from experts or imprecise pixel-level labels that are acquired through crowdsourcing. It is considerably easier to acquire these kinds of weak labels. However, a larger quantity of weak labels is usually provided for the training of a model to make up for the impreciseness of the lower quality data.

In this thesis, we compare the performances of fully- and weakly-supervised semantic segmentation methods. The methods used in this thesis are described in the methodology chapter and compared later in the experiments chapter.

2.6 Brightfield and Fluorescence Microscopy

Brightfield and fluorescence microscopy are two types of light microscopy techniques used in cellular biology (Thorn, 2016). In general, fluorescence microscopy is the preferred technique to segment cellular structures in specimens since fluorescent images are created using biomarkers that allow the cellular structures to be easily seen (Figure 6). Biomarkers are molecules introduced in the cellular structures of interest. These molecules emit fluorescence once they have absorbed light of a specific wavelength (i.e., the excitation wavelength), making thereby the cellular structures seen.

Brightfield microscopy is the more basic technique and does not require biomarkers to work (Mokobi, 2020; Wang & Fang, 2012). The contrast in a brightfield image is created when light is passed through the specimen, and the specimen's dense areas have absorbed, scattered, or deflected the light. The cellular structures are not as clearly seen on the brightfield images since the cells are not very dense and, therefore, do not absorb much light (Figure 6).

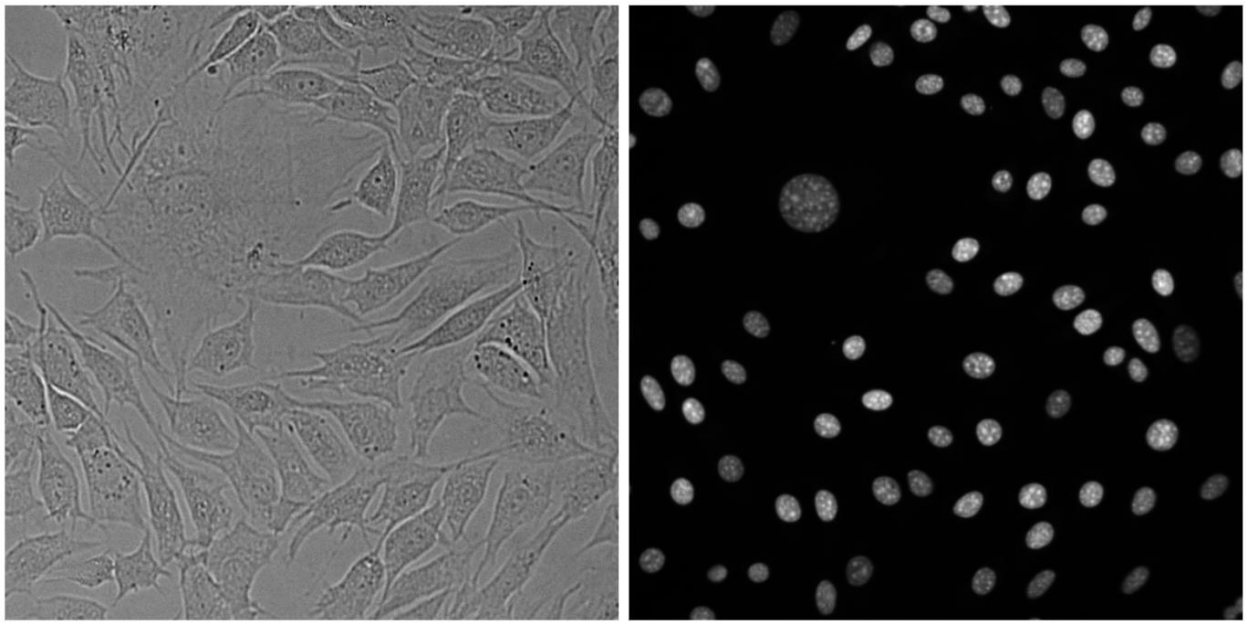


Figure 6. Cellular structures in brightfield (left) and fluorescent (right) images.

Even though it is easier to distinguish the cellular structures in fluorescent images at first glance, they also have downsides compared to brightfield images. Namely, the biomarkers used in fluorescence microscopy are expensive chemicals that require time-demanding preparation, and they might even harm the cellular structures in the specimen (Jensen, 2012; Chazotte, 2011). It has been shown that the combined usage of brightfield images and state-of-the-art segmentation models could bear comparable results to the ones obtained with fluorescent images (Salumaa, 2018). Besides, the application of cheaper, faster, and less invasive brightfield images could potentially save the pharmaceutical companies money and thousands of work hours. For the abovementioned reasons, we are focusing on the brightfield images in this thesis.

2.7 Artefacts

In microscopy, artefacts are objects/details that are unintentionally introduced to the image during the preparation of the specimen or during the imaging process. Some of the most common types of artefacts in light microscopy include dust particles, air bubbles, and wrinkles (Auburn University Department of Pathobiology, 2019; Kent, 2004).

It is clear that quantitative downstream analysis in a cellular experiment (e.g., count the number of nuclei) might be affected by the presence of artefacts in the images. To be specific, artefacts might hide some of the cellular objects or alter the structures of the surrounding cellular objects in an image (Figure 7). This, in turn, might lead to undesirable experiment results.

Certain guidelines can be followed to reduce the number of artefacts in the images (Auburn University Department of Pathobiology, 2019; Koppal, 2013). However, our observations on multiple newly created datasets show that artefacts manifest in the images even when the guidelines are strictly followed (Figure 7). For example, floating dust particles in the air can easily go unnoticed and thereby contaminate the specimen before the imaging process. For reasons like that, automated microscopy systems may conduct many experiments before the presence of artefacts are noticed in the created images. This problem has been overlooked to the extent of our knowledge as we did not find any publications where an effort had been put into segmenting or neutralizing artefacts in brightfield images.

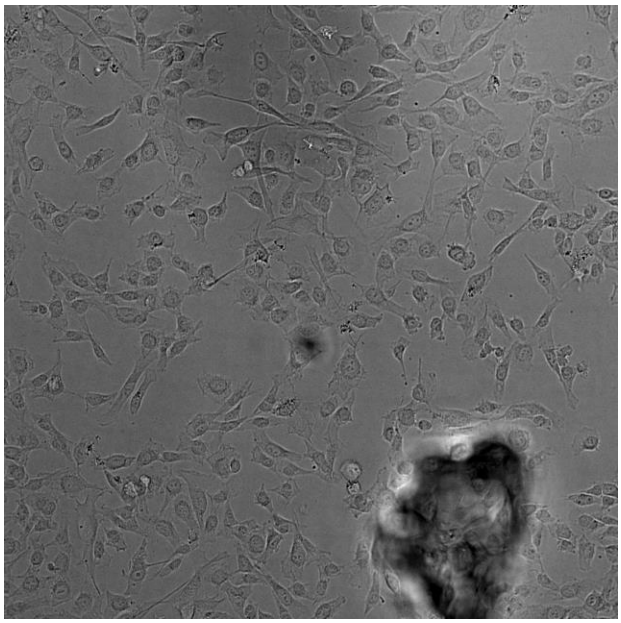


Figure 7. An example brightfield image with artefacts.

Alternatively, industrial quality control is a special field that has the sole purpose of detecting and localizing artefacts (i.e., details that deviate from normality) in textured surface images (Figure 8). The artefact localization deep learning models used in this field are mostly only trained on so-called normal (artefact-free) example images as there is often a lack of artefactual examples that

would represent all possible artefacts (Defard et al., 2020). Research (Defard et al., 2020) shows that there are two categories of artefact localization methods: reconstruction-based and embedding similarity-based methods

Reconstruction-based methods use autoencoders to localize artefacts in images (Bergmann et al., 2021; Bergmann et al., 2019). The idea behind such methods is that the autoencoders trained on normal images would struggle to reconstruct the artefactual features obtained from the artefactual image. The artefacts can be localized when the input image is compared to its reconstructed image. An example reconstruction-based method is AE-SSIM (Bergmann et al., 2019). We describe this method in more detail in the methodology chapter as it is used to segment artefacts in brightfield images.

Embedding similarity-based methods localize artefacts in images with the help of encoders (Defard et al., 2020; Yi & Yoon, 2020). The encoders are used to extract features from image patches and embed them into meaningful vectors. The embedding vectors of a test image are compared based on a similarity metric (e.g., Euclidean distance) to reference embedding vectors obtained from known normal images. A low similarity score between the respective embedding vectors point out the patches with artefacts, and consequently, the artefactual regions can also be localized in the image. Example embedding similarity-based methods include Patch SVDD (Yi & Yoon, 2020) and PaDiM (Defard et al., 2020). Patch SVDD and PaDiM are also used to segment artefacts in brightfield images and thus described in the methodology chapter.

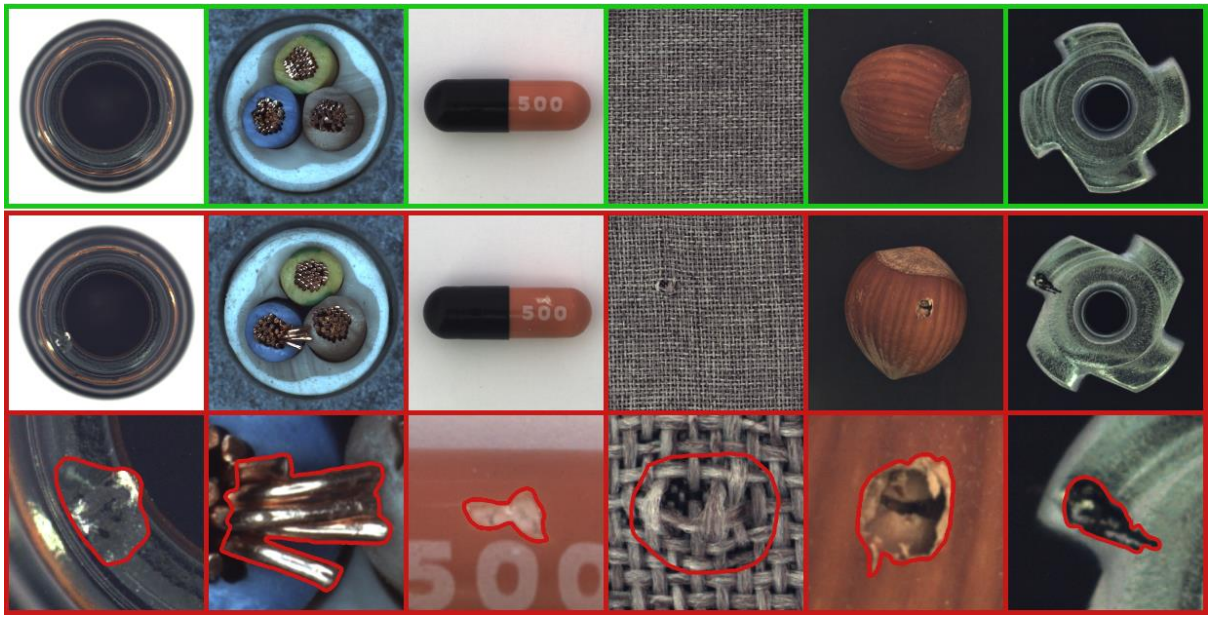


Figure 8. Example normal and artefactual images in the MVTec AD dataset. (MVTec Software GmbH, 2019)

2.8 Metrics

The purpose of this section is to give a short overview of the metrics used to evaluate the experiments conducted in this thesis.

Confusion Matrix and Basic Measures

The segmentation of artefacts is considered a binary classification task in this thesis. Each pixel can be classified as either positive (i.e. artefact) or negative (i.e. background). The evaluation (Saito & Rehmsmeier, 2015) of each predicted pixel can bear four outcomes:

- True positive (TP): pixel was correctly predicted as positive
- False positive (FP): pixel was incorrectly predicted as positive
- True negative (TN): pixel was correctly predicted as negative
- False negative (FN): pixel was incorrectly predicted as negative

The overall performance of the predictions can be visualized in a confusion matrix (Figure 9).

		Predicted	
		Positive	**Negative**
Observed	**Positive**	TP (# of TPs)	FN (# of FNs)
	Negative	FP (# of FPs)	TN (# of TNs)

Figure 9. Confusion matrix. (Saito & Rehmsmeier, 2015)

The confusion matrix is used to calculate the following basic pixel-wise metrics:

- Recall (a.k.a. Sensitivity, True Positive Rate) = TP / (TP + FN)
- Precision = TP / (TP + FP)
- F1 score (a.k.a. Dice Coefficient) = (2 * Precision * Recall) / (Precision + Recall)

Intersection over Union

Intersection over Union (IoU) is a metric used to evaluate performance in semantic segmentation tasks (Tiu, 2019). In this thesis, two IoU metrics are being calculated: pixel-wise IoU and object-wise IoU.

Pixel-wise IoU = TP / (TP + FP + FN)

Object-wise IoU is calculated based on the objects (islands formed by the pixels from the artefact class) in the ground truth and prediction masks, which must first be classified as TP, FP, or FN. An object is classified as TP when the pixel-wise IoU score of a ground truth object with respect to any predicted object surpasses an arbitrarily selected threshold. All of the other ground truth objects are classified as FN and the predicted objects as FP. The final formula of object-wise IoU = TP / (TP + FP + FN).

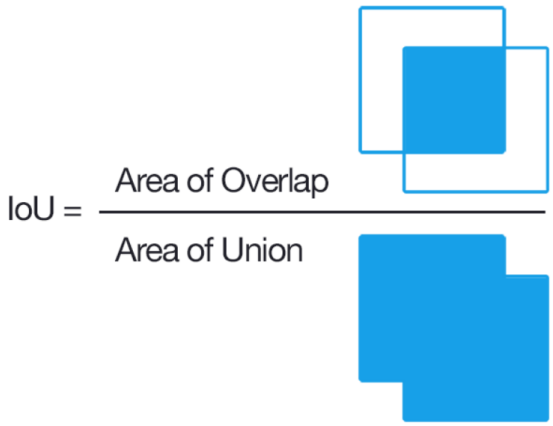


Figure 10. Intersection over Union metric. (Rosebrock, 2016)

3 Datasets and Annotations

Two brightfield microscopy datasets are being used in this thesis - the Seven cell lines dataset and the LNCaP dataset (Fishman et al., 2019). Herein we give a short overview of the two datasets.

During an exploratory search of the datasets, it was discovered that the brightfield images in the datasets contain different types of artefacts, including, but not limited to, bacterial colonies, dust particles, hair, and dead cells. Not all of the images have artefacts in them, but they are common enough to be regarded as a problem. Even though there are several types of artefacts in the brightfield images, they are all considered part of one base class called artefact (in total, there are two classes – artefact and background) in this thesis. For the time being, no differentiation is made between the types of artefacts as the general goal is to segment all of the different artefacts so that they could be avoided while making any quantitative measurements.

The labels for the weakly- and fully-supervised methods are obtained through manual annotation. The pixel-level labels needed by the fully-supervised methods require the annotator to annotate all of the pixels in the images that belong to artefacts. On the other hand, the image-level labels needed by the weakly-supervised methods only require the annotator to add a tag that specifies if an image contains artefacts or not.

Seven Cell Lines Dataset

The main work in this thesis is done on the Seven cell lines dataset (later referred to as the 7cl dataset). The dataset consists of 3024 brightfield images of 1080x1080x1 pixels, and as the name of the dataset suggests, the images contain cells from seven cell lines: canine kidney epithelial cells (MDCK), human fibrosarcoma (HT1080), human cervical adenocarcinoma (HeLa), human hepatocellular carcinoma (HepG2), mouse fibroblasts (NIH3T3), human lung carcinoma (A549), and human breast adenocarcinoma (MCF7).

For most of the experiments in this thesis, we put together a balanced subset of the 7cl dataset that consists of 365 contaminated and 365 clean images (255 contaminated + 255 clean for training, 55 contaminated + 55 clean for validation and 55 contaminated + 55 clean for testing). For an expert in this field, the pixel-level annotation of an image took 6 minutes on average, whereas the image-level annotation of an image took 3-4 seconds on average.

LNCaP

We needed another dataset for testing purposes, and for that, we chose the LNCaP dataset. The dataset consists of 782 brightfield images of 2556x2156x9 pixels, and the images contain human embryonic kidney cells. To make testing easier, we chose the brightfield images from the fifth channel (i.e., focal plane) and center cropped them to a more suitable size of 1024x1024x1.

In the end, we also constructed a balanced subset from the brightfield images of the LNCaP dataset. The subset consists of 51 contaminated and 51 clean images. Unfortunately, we do not have any time estimates for the manual annotation of this dataset.

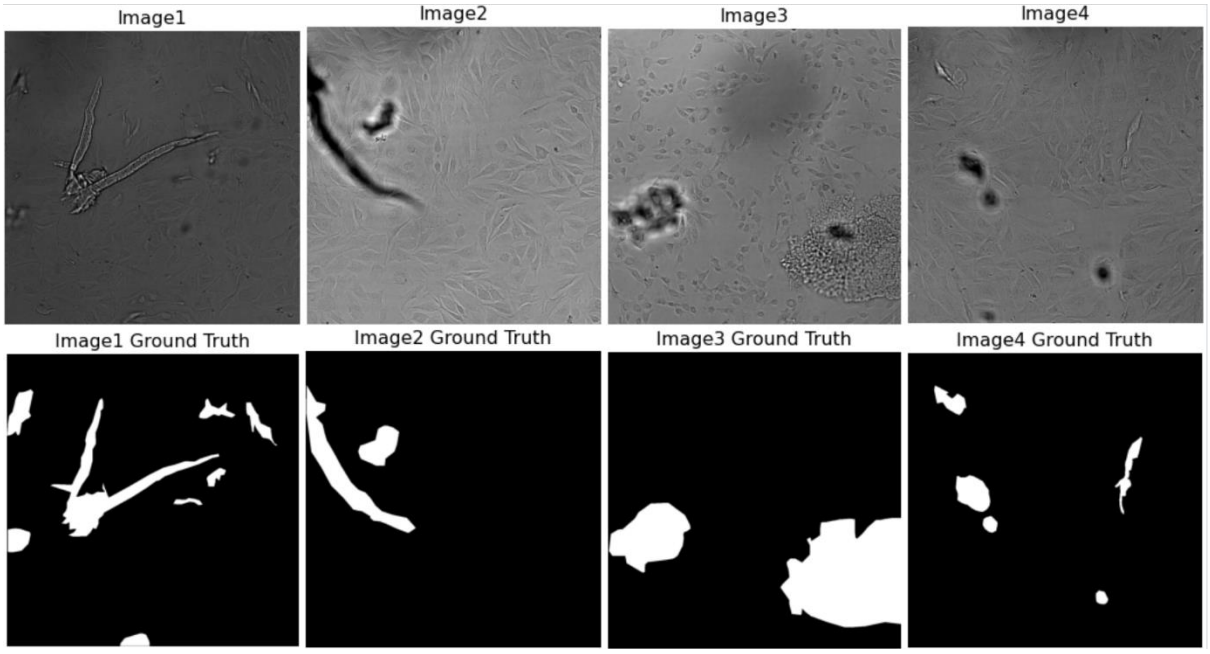


Figure 11. Example 7cl dataset images with according pixel-level labels.

4 Methodology

This chapter introduces the fully-supervised, the weakly-supervised, and the state-of-the-art artefact segmentation approaches investigated to solve the artefact segmentation problem in brightfield microscopy images. All of the models in this thesis were trained on the training images of the 7cl dataset and by using the resources (NVIDIA Tesla V100 GPUs with 32 GB of RAM) provided by the High Performance Computing Center at the University of Tartu.

4.1 Fully-Supervised Approaches

All of the approaches described in this section are fully-supervised approaches that require pixel-level labels for the training.

4.1.1 U-Net

U-Net (Ronneberger et al., 2015) is a fully convolutional neural network initially designed for the segmentation of biomedical images. The network produces a probability map for each input image. The probability map is an image in which a class label with certain probability is assigned to each of its pixels.

The architecture of U-Net is made up of a symmetric encoder-decoder network with skip connections (Ronneberger et al., 2015). The encoder (a.k.a. the contracting path) has the function of capturing the context of the input image into a set of representations (i.e., features). The representations are created by applying a combination of convolutional and max pooling layers on the input. In contrast, the decoder (a.k.a. the expanding path) is used for localizing the representations of the objects of interest that were learned by the encoder. This is achieved with the help of sequentially located upsampling and convolutional layers. The skip connections connect the convolutional layers in matching encoder and decoder levels. The connections allow the transfer of representations from the encoder straight to the decoder. The decoder can thereby recover fine-grained details better in the resulting prediction (i.e., the probability map).

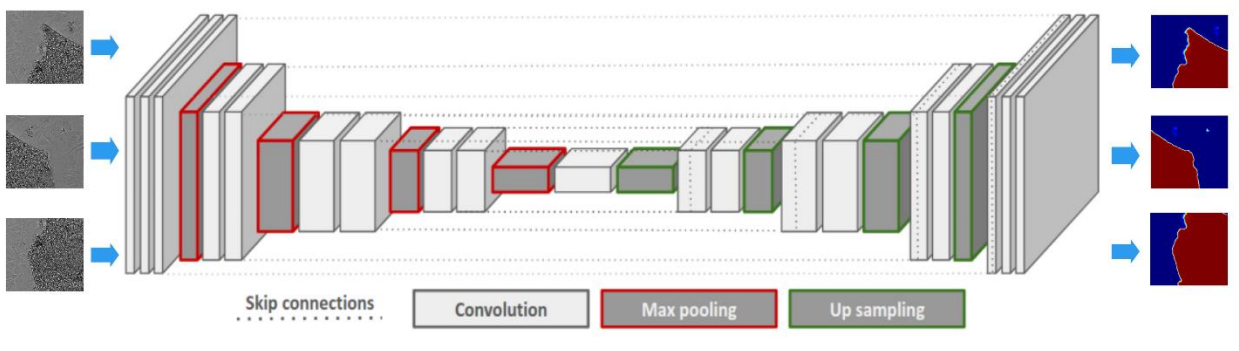


Figure 12. The U-Net architecture consists of a series of down and upsampling steps. The model takes images as inputs and outputs their respective probability maps. (Pryhoda, 2019)

4.1.2 YOLOv5

YOLOv5 is the fifth iteration of the popular "You Only Look Once" family of object detection algorithms (Jocher et al., 2021). The object detector takes an image as input and returns the bounding box coordinates of the objects of interest that were identified and positioned from the input image. It is still under active development but already offers state-of-the-art performance (Solawetz, 2020). The architecture (Solawetz, 2020) of YOLOv5 is comprised of three main building blocks:

- **Backbone** – the feature extractor of the architecture. The architecture in YOLOv5 uses a custom-built backbone which is based on Cross Stage Partial Networks (Wang, Liao, et al., 2019). Features are extracted from lower and higher level convolutional layers to allow features with different granularity levels to be passed to the architecture's next building block.
- **Neck** – this block aggregates the features passed from the different convolutional layers of the backbone by using a special type of feature pyramid called Path Aggregation Network (Liu et al., 2018). The resulting feature maps are fed to the detection part of the architecture.
- **Head** – is responsible for the object detection. Regions of interest that might contain searched objects are extracted from the feature maps. The final output is a vector that contains the predicted bounding box coordinates along with the label and confidence score of each prediction.

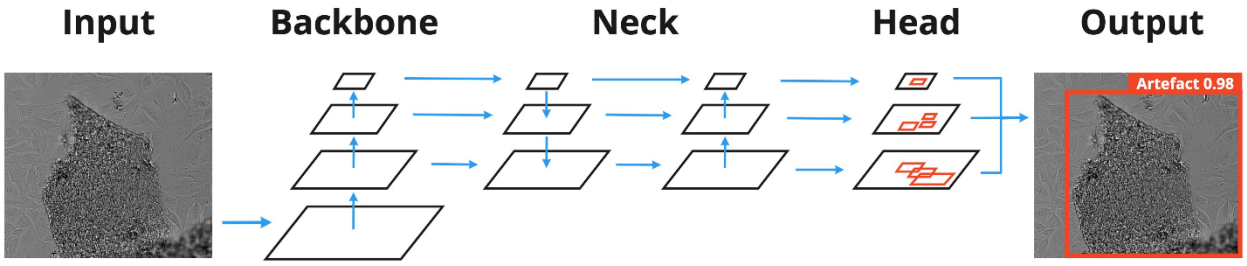


Figure 13. YOLOv5 architecture consists of three sequentially connected building blocks: the backbone, the neck and the head. The backbone extracts features from the input. The neck aggregates the features passed from the backbone. The head is responsible for detecting the objects of interest based on the obtained features.

4.1.3 U-Net Pipeline

The most straightforward solution to the artefact segmentation task is to train a segmentation model to segment artefacts using manually produced pixel-level labels as a ground truth. This particular pipeline aims to do just that with a U-Net model.

Model and Training

The U-Net model constructed in this thesis is largely based on (Ronneberger et al., 2015). The input size of the model for the input images and pixel-level labels is 1024x1024x1. This particular input size was chosen due to memory-related problems that occur with bigger inputs and architectural constraints that accompany the max-pooling layers (the input size has to be divisible by 32 in this model). The binary cross entropy loss and the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.0002 are used to update the weights of the model. A maximum number of 150 epochs are used for training. The learning rate is reduced by a factor of 10 if the validation loss has not decreased for 10 epochs and the training is stopped when the validation loss has not decreased for 30 epochs.

Before a model can be trained, the provided training images and the pixel-level ground truths are center cropped from 1080x1080x1 to 1024x1024x1 to match the model's input size. After that, the cropped images and ground truths can be used to train the U-Net model.

Inference

For inference, the test images are prepared differently from the training stage. Namely, the predictions must be of the same sizes as the test images, and the sizes of the test images do not always match the input size of the model. This is why overlapping patches are extracted from the test images that exceed the input size of the model. The extracted patches are of size 1024x1024x1 and can therefore be fed separately to the segmentation model for inference. Once the model returns the probability maps of the extracted patches, the probability maps are stacked precisely as their respective patches were. The stacked probability maps are then merged/stitched together to make up the probability maps of the test images (the overlapping parts of the probability maps are averaged). On the other hand, patches are not taken for the test images that match the input size of the model and are fed directly into the segmentation model.

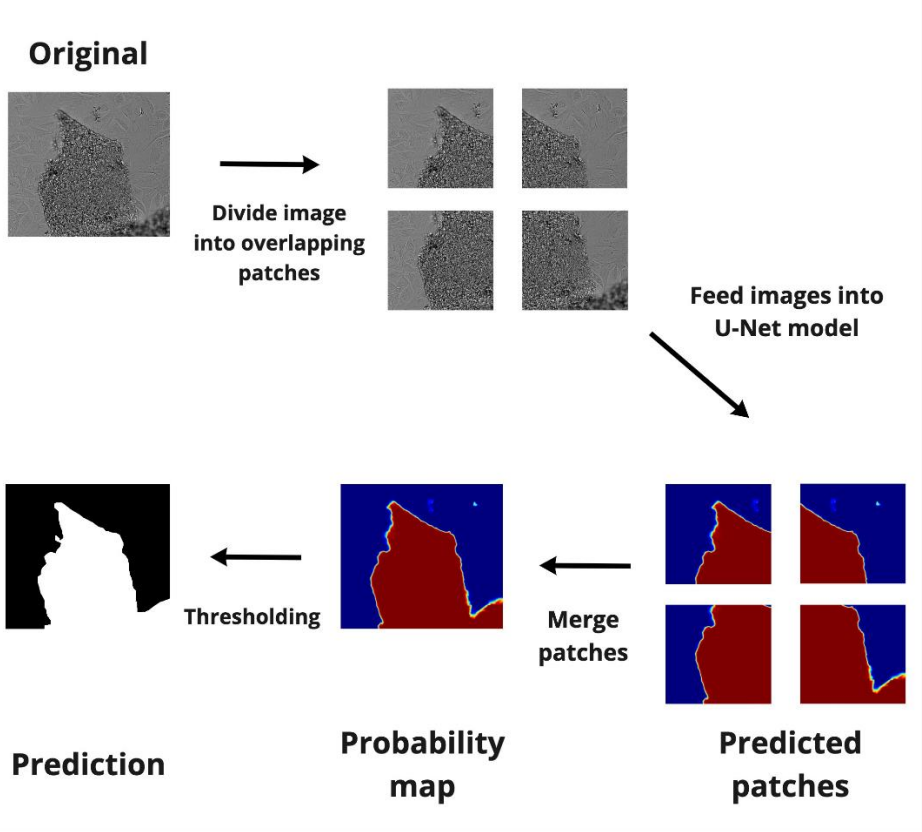


Figure 14. The inference flow of the U-Net pipeline when the test image size exceeds the input size of the model. The input image is divided into overlapping patches so that they could be fed one by one into the U-Net model for segmentation. The patches' probability maps are later stitched together to get the probability map of the entire original image.

4.1.4 YOLOv5 + U-Net Pipeline

One can use an object detection algorithm prior to the segmentation model in order to detect the artefacts before segmenting them. This is done through a two model pipeline where the first model is the YOLOv5 object detection model and the second model is the U-Net segmentation model. Artefacts are at first detected using the trained object detector, while the segmentation approach produces the corresponding probability maps.

Models and Training

The YOLOv5 model used in this thesis was repurposed from (Jocher et al., 2021). The model accepts 1080x1080x1 input images. However, the data needed for the training of a YOLOv5 model differs slightly from the training of a U-Net model as YOLOv5 does not accept pixel-level ground truths as one of its inputs. Besides the full-sized 1080x1080x1 training images, YOLOv5 requires coordinates of bounding boxes that contain artefacts. To accommodate the requirements of YOLOv5, the coordinates are inferred from the pixel-level ground truths using the findContours and boundingRect functions from OpenCV (Intel Corporation et al., 2021) and formatted as labels

so that YOLOv5 can access them. The YOLOv5 model uses the Adam optimizer with a learning rate of 0.0055 and the binary cross entropy loss to update its weights. The maximum number of epochs used for training is 150, during which the best weights of the model are saved. Additionally, the learning rate of the model is altered using the One Cycle Policy (Smith, 2018).

The U-Net model in this pipeline is trained only on small patches instead of almost the entire images used in the previously described U-Net centric pipeline. The U-Net model in this pipeline has an input size of 64x64x1. At first, ground truths of the training images are used to locate the artefacts in the images. Once the artefacts are located, an arbitrary number of small patches that match the input size of the U-Net model are randomly extracted from each located area of its respective training image and ground truth. This sort of random patch extraction allows to artificially increase the size of the training set with new and diverse training examples. In this thesis, 30 patches with each of the size of 64x64x1 are extracted from each located artefactual area. The training set is later balanced out with the same number of artefact-free image and ground truth patches to avoid the problem of having an unequal count of training patches with and without artefacts. All of the other hyperparameters used in this U-Net model are the same as in the previous U-Net centric pipeline.

Inference

YOLOv5 is used to predict bounding boxes around the anomalies in the full-sized test images. The test images are divided into overlapping patches of size 64x64x1 to match the input size of the U-Net model. Only the patches predicted to have anomalies on them by the object detector are inserted into the U-Net model to get according probability maps. The left out patches do not contain anomalies according to the object detector and can therefore be interpreted as blank/empty probability maps (i.e., probability maps that only contain zeros). Once all of the patches' probability maps are returned by the U-Net model, they can be stitched together with the blank probability maps to make up the probability maps of the full-sized test images.

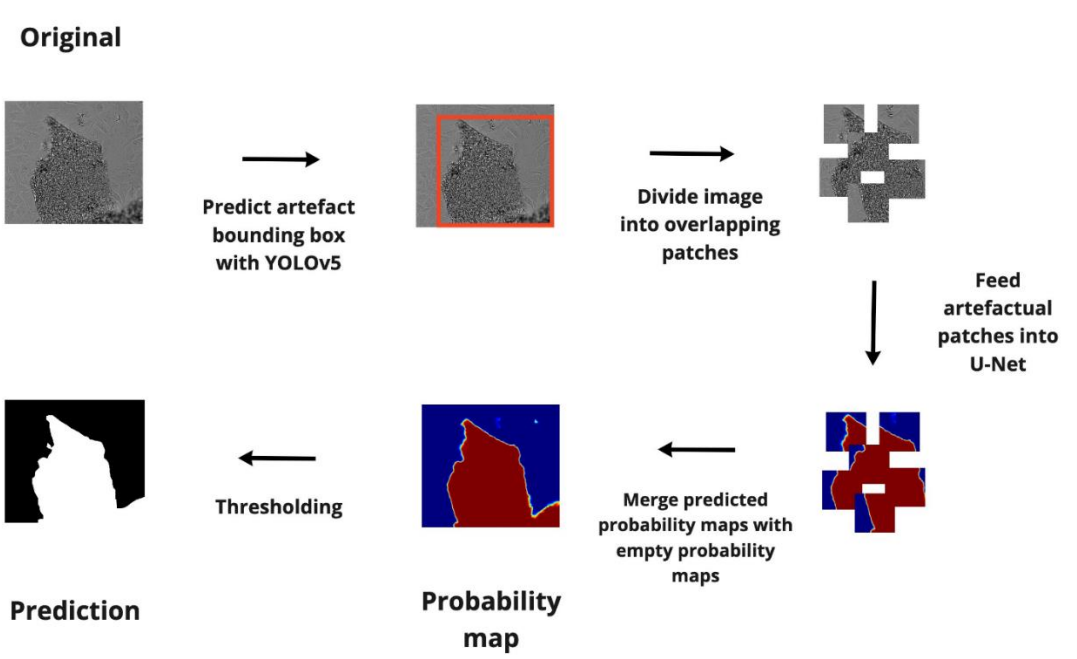


Figure 15. The inference flow of the YOLOv5 + U-Net pipeline. As the first thing, the image is fed into the object detector to detect all of the artefacts. After that, the image is divided into overlapping patches and the patches that contain artefacts are fed into the U-Net model for segmentation. The rest of the patches are interpreted as empty probability maps. The patches' probability maps are later stitched together with the empty probability maps to get the probability map of the entire original image.

4.2 Weakly-Supervised Approaches

This section describes weakly-supervised approaches that are being used in this thesis for segmenting artefacts in brightfield images. The approaches require image-level labels for the training of models that produce pixel-level predictions.

4.2.1 Score-CAM

Score-CAM (Wang, Wang et al., 2019) is a class activation mapping algorithm that requires a pre-trained image classification CNN model to work. The procedure of Score-CAM is divided into two phases. During the first phase, an input image is passed to the pre-trained CNN model and after that the activation maps are obtained from an arbitrarily chosen convolutional layer. Once the activation maps are extracted, they are upsampled to the size of the input image. In the second phase, each upsampled activation map is projected on a separate copy of the input image. Each activation map acts as a mask, and thereby only the highlighted areas in the activation map are visible in the input image. As the final step of the phase, each masked image is fed to the pre-trained CNN model with a softmax function on top of it to get a score-based weight which indicates the class discrimination relevance of the particular masked image. The final saliency map is

acquired by the linear combination of the activation maps from the first phase and the score-based weights from the second phase. The resulting saliency map is essentially a probability map that can later be thresholded and taken as a prediction or pseudo-label.

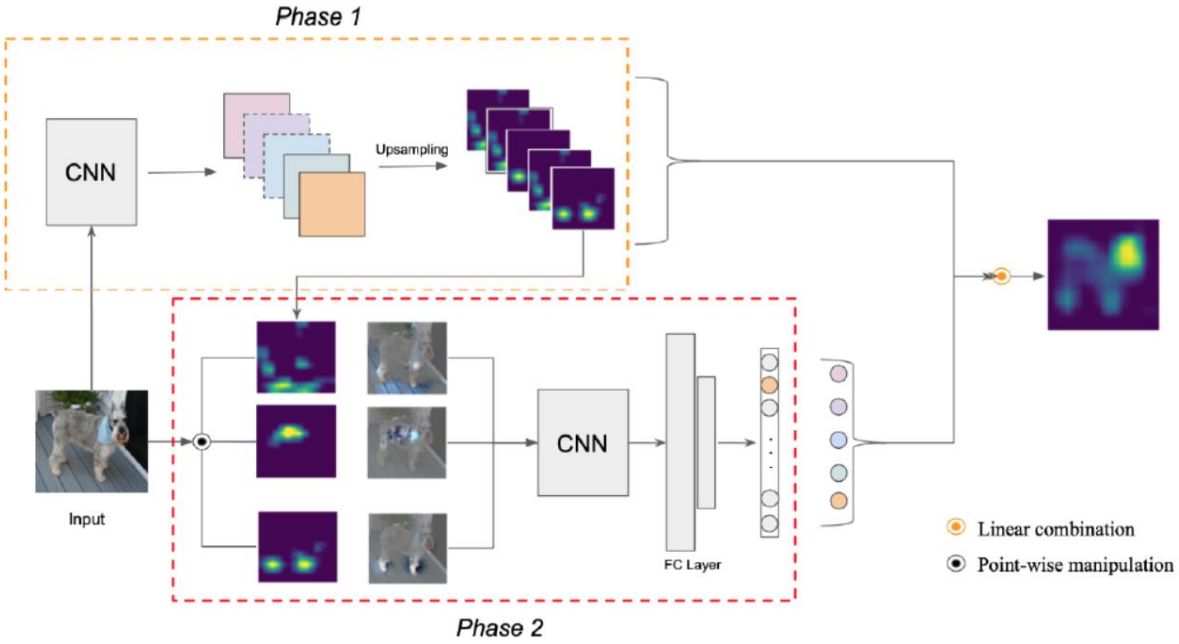


Figure 16. The Score-CAM algorithm (Wang, Wang et al., 2019). In the first phase of the algorithm, activation maps are extracted from a convolutional layer of the chosen CNN model. In the second phase, the activation maps from the first phase are overlain on copies of the original image, and after that, the masked images are fed to the CNN model to get score-based weights. The final saliency map is created from the linear combination of the activation maps from the first phase and the respective score-based weights from the second phase.

4.2.2 Score-CAM Pipeline

The first weakly-supervised approach for segmenting artefacts utilizes the Score-CAM algorithm to produce probability maps. As mentioned in the algorithm description, it requires an image classification CNN model to work. One of the perks of the Score-CAM method is that the classification model can be selected arbitrarily and thus can easily be changed once a more suitable model comes along. The Score-CAM algorithm itself that is being used in this pipeline can be found in GitHub (tabayashi0117, 2020).

Model and Training

The binary classification model used in this thesis is largely based on the ResNet50 (He et al., 2016) architecture that can be found in the Keras framework (Keras Team, 2015). The input size of the model is 1024x1024x1 pixels. The Adam optimizer with the learning rate of 0.002 and the binary cross entropy loss are used to update the model weights. The maximum number of epochs is 150. The learning rate is reduced by a factor of 10 when the validation loss has not decreased

for 10 epochs and the training of the model is stopped when the validation loss has not decreased for 30 epochs.

The ResNet model is trained to solve a binary classification task – to determine whether an image contains artefacts or not. Before training, the training images have to be center cropped to the size of 1024x1024x1 so that they could be used along with their respective image-level labels in the model's training process.

Inference

The inference process of this pipeline consists of two stages. In the first stage, the images are copied, resized and then fed to the trained classification model to determine whether they contain artefacts or not. An empty probability map is returned for each image that gets a negative answer from the classification model. The rest of the images (i.e., the images that contain artefacts) reach the next stage. In the second stage, overlapping patches of 1024x1024x1 are extracted from each test image that exceeds the input size of the classification model and thereafter inserted separately into the Score-CAM algorithm. The algorithm returns a probability map for each inserted patch. Finally, the probability maps of the patches are stitched together to produce the full-sized probability maps that correspond to each test image. Patches are not taken for the test images that match the input size of the classification model and are fed directly into the Score-CAM algorithm.

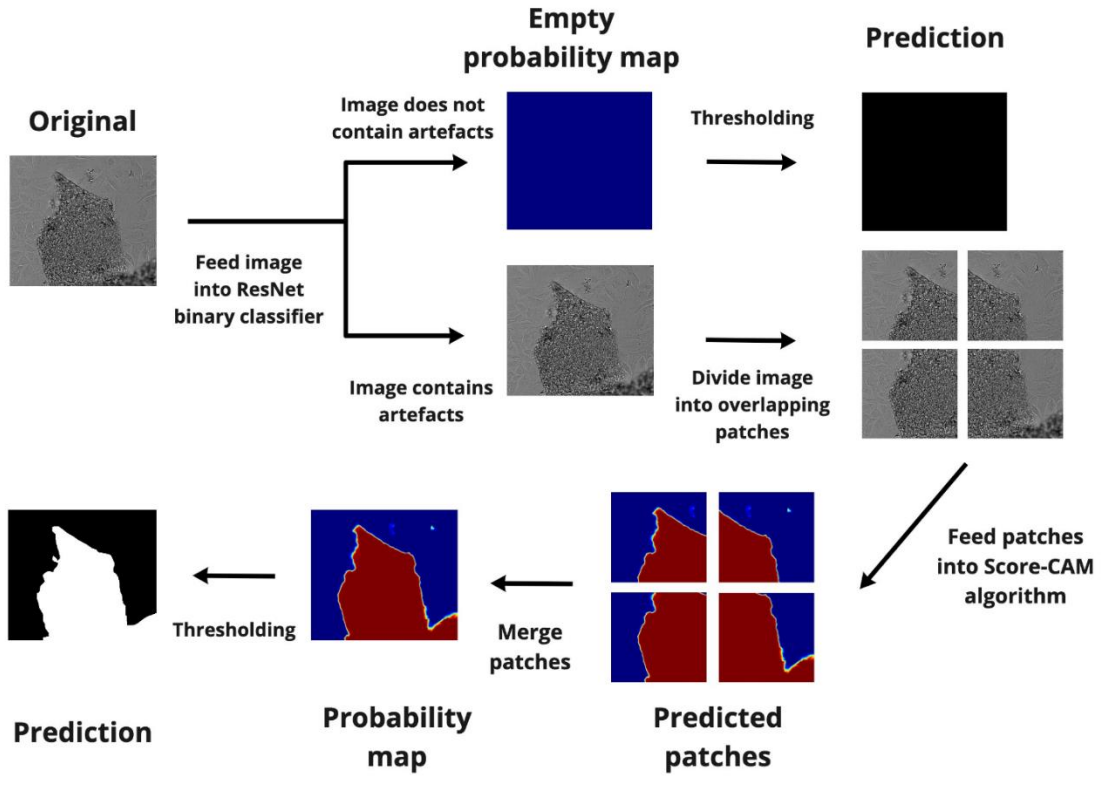


Figure 17. The inference flow of the Score-CAM pipeline when the test image size exceeds the input size of the classification model. The image is initially fed into a binary classifier to determine

if it contains any artefacts or not. If the classifier predicts that the image does not contain any artefacts, an empty probability map with the size of the original image is returned. Otherwise, the image is divided into overlapping patches, fed into the segmentation model and the resulting patches' probability maps are stitched together to get the probability map of the original image.

4.2.3 Score-CAM + U-Net Pipeline

The second weakly-supervised approach is the combined approach of Score-CAM and U-Net. In essence, the Score-CAM approach is used to generate pseudo-labels that would act as the pixel-level ground truths for the training of the U-Net approach.

Training

The training in this pipeline involves the ResNet classification model and the U-Net model. Initially, the ResNet classification model is trained with the training images and their image-level labels precisely as it is described in the Score-CAM pipeline subsection. Thereafter, the training images are fed to the Score-CAM approach to get the respective pixel-level pseudo labels. Finally, the training images are used with the pixel-level pseudo labels to train a U-Net model as is described in the U-Net centric pipeline.

Inference

Only the trained U-Net model is used in the inference process. The detailed inference description can be found in the U-Net pipeline subsection.

4.3 State-of-the-Art Artefact Segmentation Approaches

This section introduces the state-of-the-art artefact segmentation approaches that were found from the literature. We classify these approaches as weakly-supervised as they require artefact-free images for the training process.

4.3.1 Patch Support Vector Data Description

Patch Support Vector Data Description (Patch SVDD) is an embedding similarity-based method that is used to localize artefacts in images (Yi & Yoon, 2020). The idea of this approach is to divide the images into small patches so that each patch could be inspected for the presence of artefacts. An encoder is used to map the projections of the patches into a feature space. Initially, the encoder is fed with known artefact-free patches and is trained to minimize the distances between the projections of these patches. This way, semantically similar projections of the artefact-free patches form clusters in the feature space. These clusters can later be taken as decision boundaries to discriminate between artefactual and artefact-free projections. During inference, the encoder is fed with known artefact-free patches and test patches. The anomaly score of each test patch (i.e., how artefactual is the patch) is determined by the Euclidean distance between the projection of the test patch and the projection of the nearest known artefact-free patch in the feature space. The artefacts in an image can then be localized once all of its artefactual patches are detected. The approach was tested based on the implementation available in GitHub (nuclearboy95, 2020).

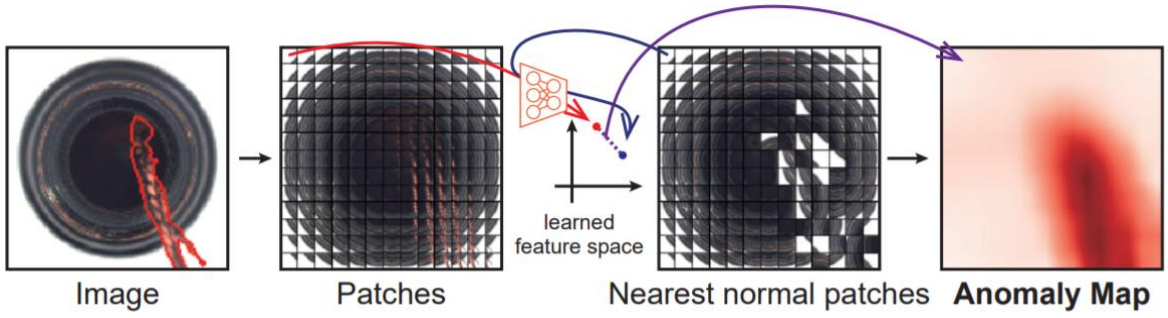


Figure 18. Inference flow of the Patch SVDD method (Yi & Yoon, 2020). *The anomaly score of a test patch is determined by the Euclidean distance of the patch and its nearest artefact-free patch in the learned feature space.*

4.3.2 Patch Distribution Modeling

Similarly to Patch SVDD, Patch Distribution Modeling (PaDiM) is also an embedding similarity-based method that is used for localizing artefacts (Defard et al., 2020). The algorithm uses a pre-trained CNN model to extract features from image patches. The features are extracted from multiple convolutional layers of the model to encode fine-grained details into embedding vectors. The embedding vectors that originate from image patches that reside at the same spatial positions of known artefact-free images are used to learn the parameters of an area in a multidimensional feature space that corresponds to normality (i.e., the multivariate Gaussian distribution parameters at that specific spatial position). Thereafter, the anomaly score of each test patch can be determined by calculating the Mahalanobis distance (McLachlan, 1999) between the embedding vector of the test patch and the learned area in the multidimensional feature space. The detection of the artefactual patches leads to the localization of the artefacts in the images. The approach was tested based on the implementation found in GitHub (xiahaifeng1995, 2021).

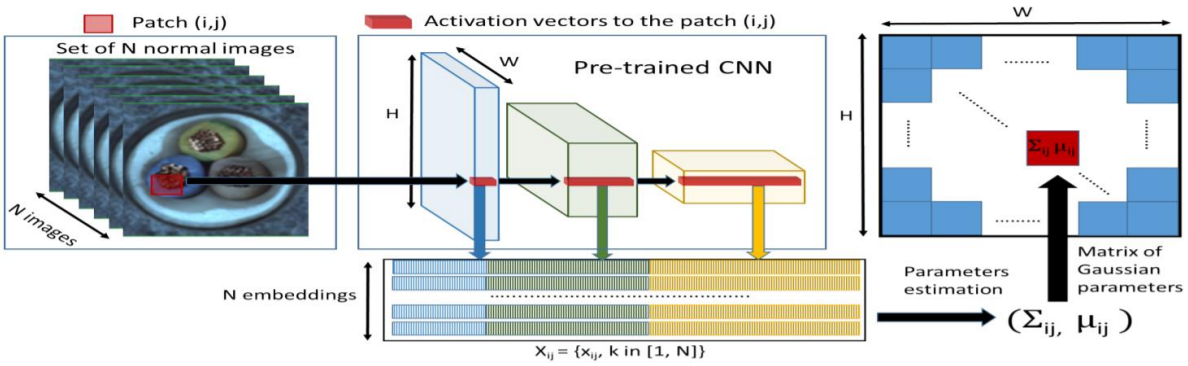


Figure 19. The overall flow of the PaDiM method (Defard et al., 2020). The anomaly score of a test patch is determined by the Mahalanobis distance between the embedding vector of the test patch and the learned distribution that corresponds to normality at that specific spatial position.

4.3.3 Autoencoder with a Structural Similarity Index Metric

Autoencoder with a Structural Similarity Index Metric (AE-SSIM) is a reconstruction-based method used to localise artefacts in images (Bergmann et al., 2019). In essence, the method constitutes an autoencoder that uses a structural similarity-based metric to measure the reconstruction accuracy of an image. The structural similarity metric does not compare the original and reconstructed images per individual pixels but rather per a group of adjacent pixels based on three features - luminance, contrast and structure. These features help improve the reconstruction quality as much attention is put into the fine-grained details present in the original images.

The chosen autoencoder is trained on artefact-free images, and thus, it only learns the characteristics of artefact-free images. During inference, the autoencoder tries to reconstruct the test images. As the autoencoder knows only the characteristics of artefact-free images, it fails to reconstruct the artefactual areas in the test images. The output of the method is a residual map (i.e., probability map) that marks the pixel-wise difference of each test image and its reconstruction. The approach was tested based on an implementation available in GitHub (Boumessouer, 2020).

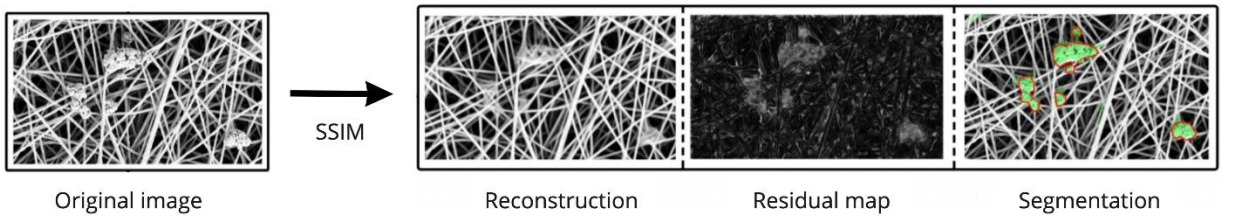


Figure 20. Inference flow of the AE-SSIM method (Yi & Yoon, 2020). *The residual map is created by calculating the pixel-wise difference between the original image and the image reconstructed by the autoencoder that uses a structural similarity-based metric to measure the reconstruction accuracy.*

4.4 Augmentation

It is well known that the training of DNNs requires large amounts of data that should ideally be rather diverse for the models to generalize well (DeVries & Taylor, 2017). Data augmentation is a popular method used to increase the diversity and size of a training dataset by applying image altering functions on the dataset's images and labels. The set of functions used in each domain must be carefully selected as not all of the altered images always have a positive effect on the performance of the model. For example, turning a horse image by 180 degrees is not useful because horses are usually encountered walking upright. On the other hand, turning cellular images by 90 or 180 does not change anything as all the cells in the images remain legit.

Several functions of an image augmentation library called Albumentations (Buslaev et al., 2020) were tested to find the most suitable functions for the 7cl dataset (Figure 21). All sorts of

combinations of the selected functions were applied to the images and used to train U-Net models. The most consistent results were achieved when the models were trained on images that were flipped, randomly rotated and transposed. Therefore these augmentation functions are used to augment the training images in the upcoming experiments.

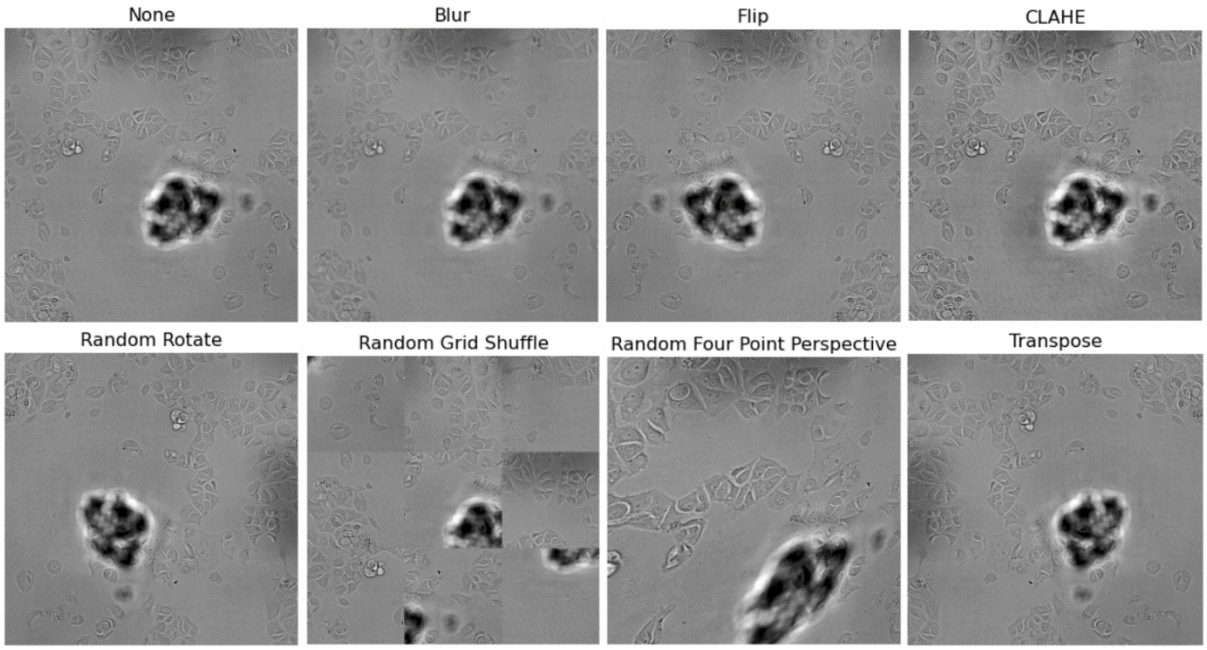


Figure 21. Tested augmentation functions.

4.5 Thresholding of Probability Maps

The probability maps returned by all of the model instances are thresholded to get predictions that can be quantitatively and qualitatively measured. The thresholding values are chosen in a way that would maximize the pixel-wise IoU scores of the predictions.

5 Experiments and Results

The main objective of the thesis is to determine if weakly-supervised methods are able to segment artefactual regions in brightfield images with performance similar to those of fully-supervised approaches. In this chapter, the experiments involving fully- and weakly-supervised approaches are described in detail.

5.1 Baseline Fully-Supervised Approach

In this experiment, we investigate the potential of two slightly different fully-supervised approaches for segmenting artefacts. The first approach uses only the widely-used U-Net model to segment artefacts. In contrast, the second approach uses the state-of-the-art YOLOv5 object detection model to detect the artefacts first and then the U-Net model to segment them. The superior approach is selected as the baseline approach and compared to weakly-supervised approaches in the following experiments. The model instances of the two approaches used in this experiment are trained on the entire training set, validated on the validation set and tested on the test set of the 7cl dataset.

The performances of the two aforementioned approaches are presented in Table 1. These results show that they both got good results across all metrics. The performance comparison suggests that U-Net performs better as it got a higher score in three out of the five compared metrics. This includes better results in pixel-wise IoU (U-Net 0.8563 vs YOLOv5 + U-Net 0.8267) and pixel-wise F1 score (U-Net 0.9226 vs YOLOv5 + U-Net 0.9051) that are the two most widely used metrics for evaluating the quality of semantic segmentation.

Table 1. The quantitative performance comparison of the U-Net approach and YOLOv5 + U-Net approach. The best results are highlighted in yellow.

	Pixel-wise Recall	Pixel-wise Precision	Pixel-wise F1 Score	Pixel-wise IoU	Object-wise IoU
U-Net 510	0.9323	0.9131	0.9226	0.8563	0.2389
YOLOv5 + U-Net 510	0.8914	0.9193	0.9051	0.8267	0.2495

Qualitatively, the predictions of both approaches seem to align well with the majority of the artefacts in the ground truth regardless of the size, shape or structure of the artefacts (Figure 22 first row). The most significant difference between the approaches is perhaps that YOLOv5 + U-Net seems to struggle in some cases with predicting the entirety of the bigger artefacts (Figure 22 second row). However, both approaches only made a few predictions in the clean images of the test set, essentially avoiding false positive mistakes (Figure 22 third row).

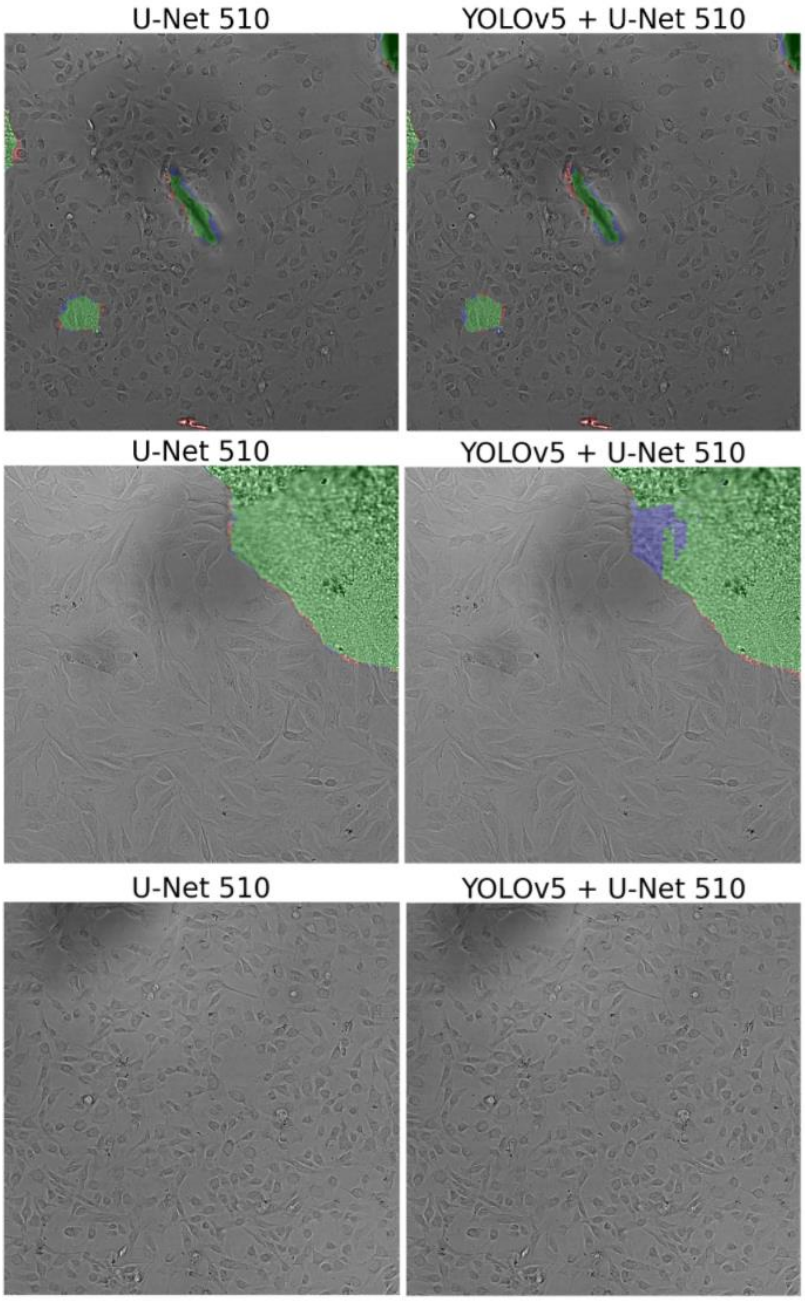


Figure 22. Example artefact segmentation results from the fully-supervised approaches where the thresholded predictions are overlaid on the original images. The green pixels represent the correctly predicted artefactual regions (true positive), the red pixels show areas where the approaches predicted an artefact by mistake (false positive), and the blue pixels indicate areas where approaches failed to predict the underlying artefacts (false negative).

All in all, the results showed that both of these approaches accurately predicted most of the artefacts present in the brightfield images of the 7cl dataset. Therefore, it can be concluded that the fully-supervised approaches offer a reliable solution to the artefact segmentation problem. Despite that, the results suggest that segmenting artefacts using only the widely-used U-Net model

turned out to be the superior fully-supervised approach. Therefore, U-Net is considered the baseline approach.

5.2 Weakly-Supervised Approaches

In the second experiment we investigate the potential of segmenting artefacts in a weakly-supervised manner with the class activation mapping-based method – Score-CAM. Thereafter, we compare the results of the Score-CAM approach to the results of the state-of-the-art artefact segmentation approaches (PaDiM, Patch SVDD and AE-SSIM) to determine the best performing weakly-supervised approach. The state-of-the-art approaches can be considered weakly-supervised approaches as their training processes require only clean images (i.e., images that are artefact-free). Therefore, the model instances of the state-of-the-art approaches are trained on all of the clean images in the 7cl training set (255 images). In contrast, the model instance of Score-CAM is trained on all of the images in the 7cl training set (510 images). All of the model instances are validated on the validation set and tested on the test set of the 7cl dataset.

The quantitative results are presented in Table 2. Score-CAM got the best results in all but one of the measured metrics when compared against the state-of-the-art approaches. Only the object-wise IoU score of AE-SSIM (0.0975) was better than the respective score of Score-CAM (0.0310). AE-SSIM can be considered the best state-of-the-art approach as it got the highest scores amongst the state-of-the-art approaches in metrics like pixel-wise F1 score and pixel-wise IoU.

Table 2. The quantitative performance comparison of the Score-CAM approach and the state-of-the-art approaches. The best results are highlighted in yellow.

	Pixel-wise Recall	Pixel-wise Precision	Pixel-wise F1 Score	Pixel-wise IoU	Object-wise IoU
Score-CAM 510	0.6266	0.6454	0.6359	0.4661	0.0310
PaDiM 255	0.5454	0.2487	0.3416	0.2060	0.0042
Patch SVDD 255	0.4437	0.2475	0.3177	0.1889	0.0030
AE-SSIM 255	0.4715	0.6138	0.5333	0.3636	0.0975

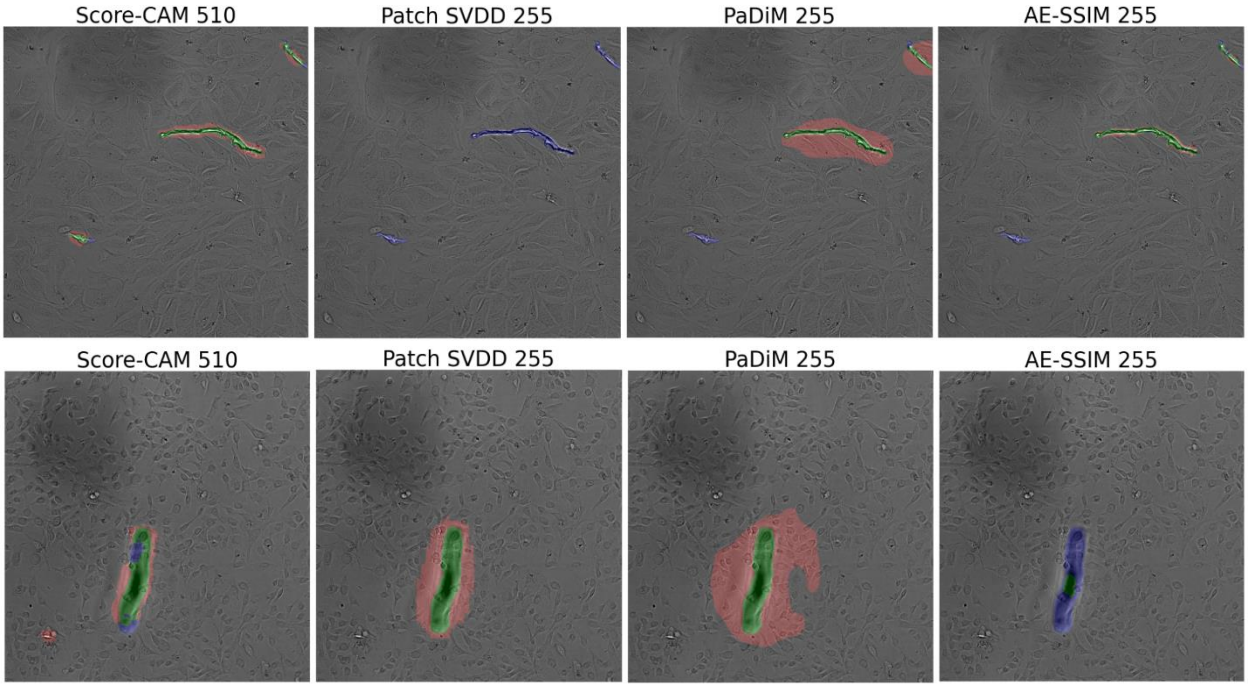
Qualitatively, Score-CAM localized most of the smaller artefacts, but by overpredicting them to a small extent (Figure 23 first row). On the other hand, the segmentation of the bigger artefacts turned out to be less than ideal as Score-CAM struggled to segment them as a whole (Figure 23 third row). Even though Score-CAM made occasional false positive predictions in the clean images of the test set, the majority of the clean images were predicted correctly as having no artefacts (Figure 23 fourth row).

The qualitative results (Figure 24) also explain the bad object-wise IoU score of Score-CAM (Table 2) as the metric seems to be very sensitive to any overprediction and underprediction. The weakly-supervised approach overpredicted small artefacts and underpredicted bigger artefacts,

consequently resulting in a low object-wise IoU score. However, the slight overpredictions might actually be beneficial as the nuclei that reside at the borders of the artefacts might also be affected by the artefacts. Therefore, we do not consider the object-wise IoU metric to be the best metric to evaluate the performance of the artefact segmentation approaches.

PaDiM and Patch SVDD could not segment the artefacts in the brightfield images very well. Even though both of the approaches were quite often able to localize the artefacts in the images, they did it by overpredicting the artefacts to a large extent (Figure 23 first, second and third row). Overpredictions of this magnitude call the usefulness of these two approaches into doubt as big chunks of the actual clean parts of the images are discarded.

Although AE-SSIM segmented the artefacts much better than the other state-of-the-art approaches, the predictions of the approach can still be characterized as inconsistent. In some cases, the predictions of AE-SSIM aligned almost perfectly with the ground truth (Figure 23 first row). In other cases, the approach was totally unable or barely able to predict the artefacts present in the images (Figure 23 second and third row). The fact that the predictions aligned well with the ground truth in some cases helped to achieve a better object-wise IoU than Score-CAM and Score-CAM + U-Net. On the negative side, AE-SSIM made occasional false positive predictions in the clean images of the test set (Figure 23 fourth row).



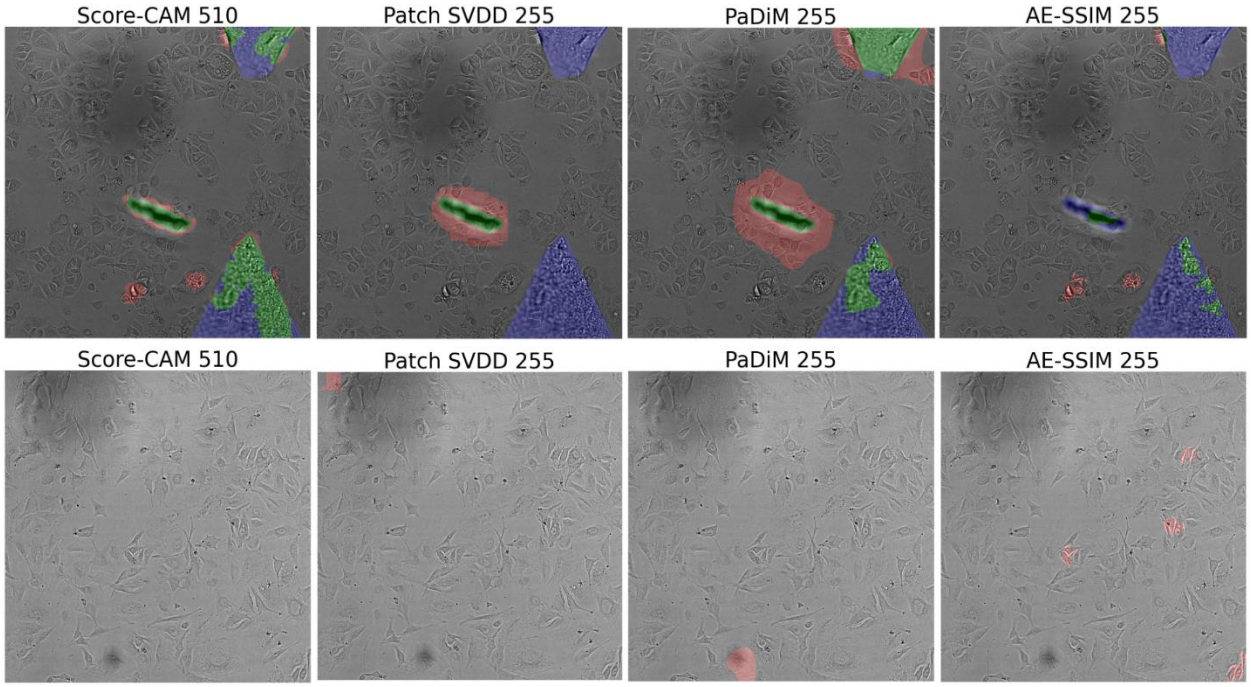


Figure 23. Example artefact segmentation results from the weakly-supervised approaches where the thresholded predictions are overlaid on the original images. The green pixels represent the correctly predicted artefactual regions (true positive), the red pixels show areas where the model predicted an artefact by mistake (false positive), and the blue pixels indicate areas where models failed to predict the underlying artefacts (false negative).

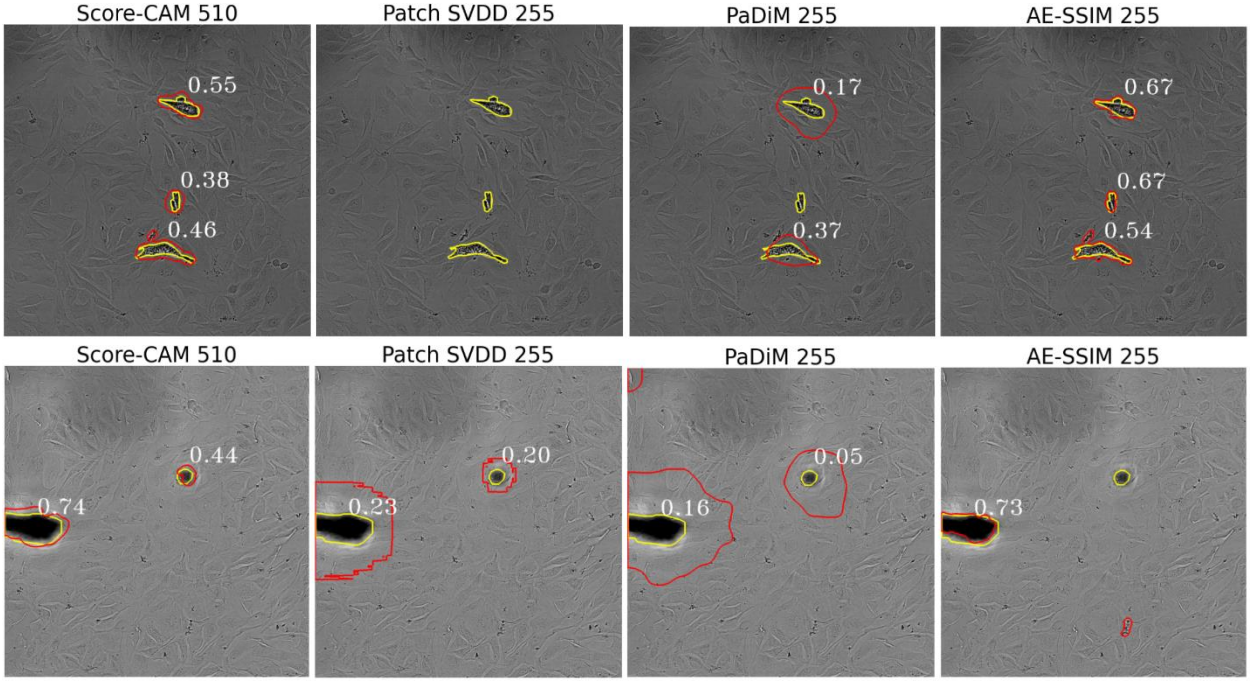


Figure 24. Example segmentation results from the weakly-supervised approaches where the yellow contours represent the ground truths and the red contours represent the predictions. An object is

regarded as correctly predicted (true positive) in the object-wise IoU metric when the pixel-wise IoU score of a ground truth object with respect to a predicted object surpasses the threshold score of 0.5.

The quantitative and qualitative results indicate that Score-CAM performed relatively well as it was able to segment most of the smaller and large parts of the bigger artefacts in the brightfield images. Besides, Score-CAM performed better than the state-of-the-art artefact segmentation approaches and is thereby the best weakly-supervised approach in this experiment. Consequently, Score-CAM is used in the following experiment as a part of a newly constructed pipeline.

5.3 Score-CAM vs. Score-CAM + U-Net

We conduct the third experiment to determine if combining the weakly-supervised Score-CAM approach with the fully-supervised U-Net approach would result in better performance than that of the Score-CAM approach. To be specific, the qualitative results from the previous experiment showed that Score-CAM overpredicted smaller and underpredicted bigger artefacts. Therefore, we hypothesized that a segmentation model like U-Net could generalize and produce accurate predictions even when trained on slightly inaccurate pixel-level pseudo-labels that are generated by Score-CAM. The model instances of the two approaches are trained on the training set, validated on the validation set and tested on the test set of the 7cl dataset.

The quantitative results are displayed in Table 3. The results show that Score-CAM + U-Net performed better than Score-CAM in four out of the five measured metrics. Even though Score-CAM + U-Net produced better results than Score-CAM, the results were only marginally better. For example, the pixel-wise IoU of Score-CAM + U-Net was 0.4756, whereas the same metric for Score-CAM was 0.4661. The qualitative results (Figure 25) also show that the predictions of the two approaches are very similar; perhaps Score-CAM + U-Net segments the bigger artefacts a little better.

Table 3. The quantitative performance comparison of the weakly-supervised Score-CAM and Score-CAM + U-Net approaches. The best results are highlighted in yellow.

	Pixel-wise Recall	Pixel-wise Precision	Pixel-wise F1 Score	Pixel-wise IoU	Object-wise IoU
Score-CAM 510	0.6266	0.6454	0.6359	0.4661	0.0310
Score-CAM + U-Net 510	0.6605	0.6294	0.6446	0.4756	0.0318

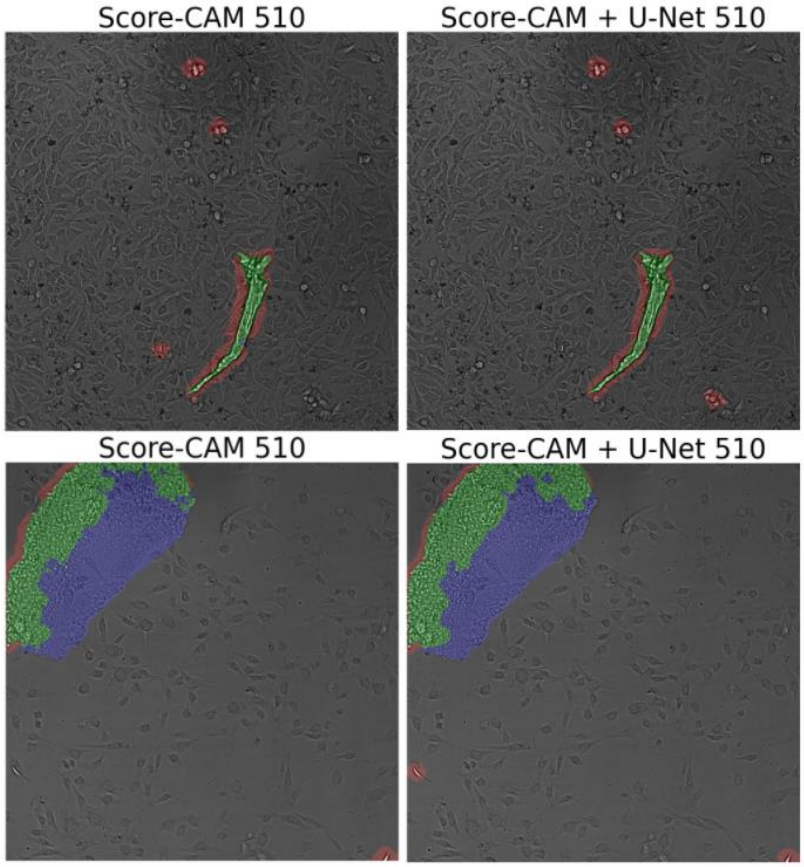


Figure 25. Example artefact segmentation results from the weakly-supervised Score-CAM and Score-CAM + U-Net approaches. The thresholded predictions of the approaches are overlaid on the original images. The green pixels represent the correctly predicted artefactual regions (true positive), the red pixels show areas where the model predicted an artefact by mistake (false positive), and the blue pixels indicate areas where models failed to predict the underlying artefacts (false negative).

The quantitative results established that Score-CAM + U-Net performed better than Score-CAM. However, the U-Net model could not generalize to the expected levels when trained with slightly inaccurate pixel-level pseudo-labels. The hoped performance boost was most likely not achieved since the model was not only trained but also validated on pixel-level pseudo-labels. Giving the model manually labelled pixel-level labels for validation would have made it a semi-supervised

method, and therefore it would not have met the weakly-supervised method criteria anymore. Nevertheless, as the results of Score-CAM + U-Net were still marginally better than that of Score-CAM, it was decided to compare both of these weakly-supervised approaches to the baseline approach in the next experiment.

5.4 Baseline Fully-Supervised Approach vs. Weakly-Supervised Approaches

We designed this experiment to see what performance level of the baseline approach can be achieved with Score-CAM and Score-CAM + U-Net. This is assessed by comparing the performance levels of the two weakly-supervised approaches to the performance levels of the baseline fully-supervised approach when their model instances are trained on a varying number of images.

At first, an equal number of clean and contaminated images from 7cl are randomly sampled without replacement into 41 image sets - five sets for each sample size of 2, 4, 8, 16, 32, 64, 128, 256 images, and one set with the complete set of 510 images (5 * 8 + 1 = 41 image sets). The five sets that are sampled for each sample size of images are used to account for some of the error that entails random sampling (some sets might only contain perfectly suitable images and other sets might only contain unsuitable images). After that, each set is used to train a separate model instance of the baseline and the weakly-supervised approaches. All of the model instances are validated on the entire validation set and tested on the entire test set of the 7cl dataset.

The pixel-wise comparison unsurprisingly shows that all the approaches achieve better results when trained on more images (Figure 26). The figure also demonstrates that the pixel-wise scores of the weakly-supervised approaches increase at a slower rate than the pixel-wise scores of the baseline approach. As expected, none of the approaches performs well when trained on a couple of images. The performance of the baseline approach starts to slowly increase when trained on eight images. In contrast, the weakly-supervised approaches' performances start to increase when trained on 32 images. The best model instances of Score-CAM and Score-CAM + U-Net that are trained on the entire training set of 510 images are approximately on par with the median U-Net model instance trained on 32 images in the pixel-wise IoU metric.

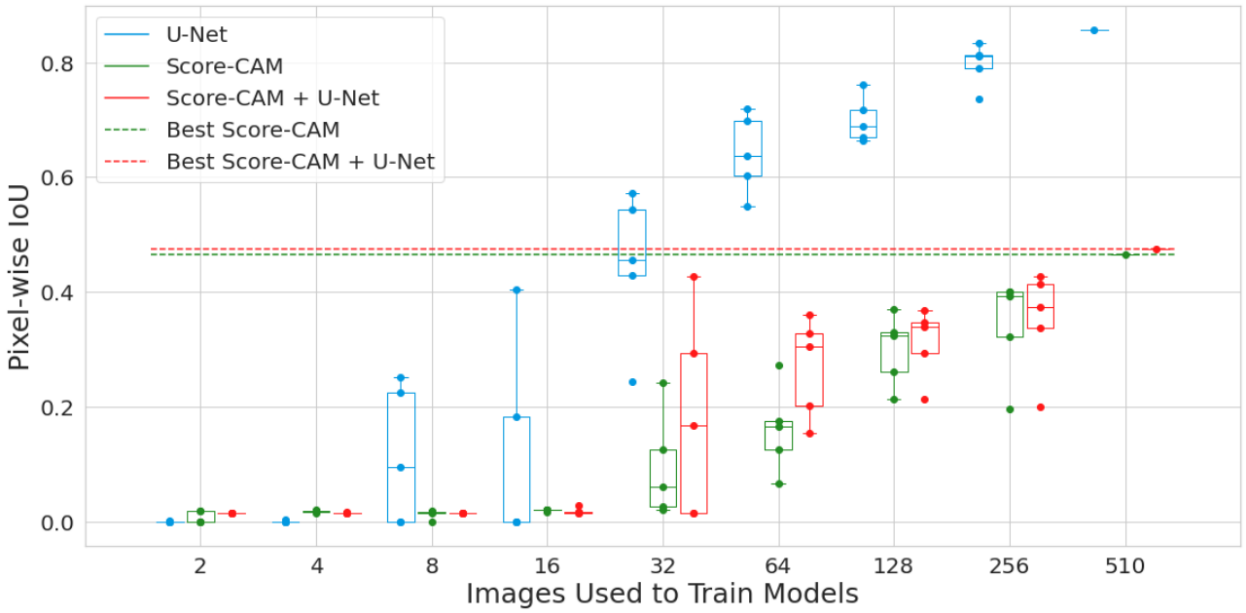


Figure 26. Pixel-wise IoU comparison of the baseline fully-supervised and the weakly-supervised Score-CAM and Score-CAM + U-Net approaches. Each dot in the figure represents the pixel-wise IoU score of a model instance trained with the number of images shown on the x-axis.

The quantitative results shown in Table 4 demonstrate that the best U-Net model instance achieved by far better results in all of the measured metrics than the best model instances of Score-CAM and Score-CAM + U-Net. For example, the pixel-wise IoU score of U-Net was 0.8563, which is a lot higher compared to 0.4661 for Score-CAM and 0.4756 for Score-CAM + U-Net. The most significant difference between the median U-Net model instance and the best weakly-supervised model instances lies in the object-wise IoU score in which the median U-Net model instance has a considerably higher score.

Table 4. The quantitative performance comparison of the baseline and the weakly-supervised Score-CAM and Score-CAM + U-Net approaches. The best results are highlighted in yellow.

	Pixel-wise Recall	Pixel-wise Precision	Pixel-wise F1 Score	Pixel-wise IoU	Object-wise IoU
U-Net 510	0.9323	0.9131	0.9226	0.8563	0.2389
Median U-Net 32	0.5561	0.7171	0.6264	0.4561	0.1321
Score-CAM 510	0.6266	0.6454	0.6359	0.4661	0.0310
Score-CAM + U-Net 510	0.6605	0.6294	0.6446	0.4756	0.0318

The qualitative results in Figure 27 display that the best U-Net model instance segments the artefacts the best. The median U-Net and the best weakly-supervised model instances segment most of the smaller artefacts, but the median U-Net model instance does it without overpredicting (Figure 27 first and second row). On the other hand, the best weakly-supervised model instances

perform better than the median U-Net model instance when segmenting the bigger artefacts (Figure 27 third row).

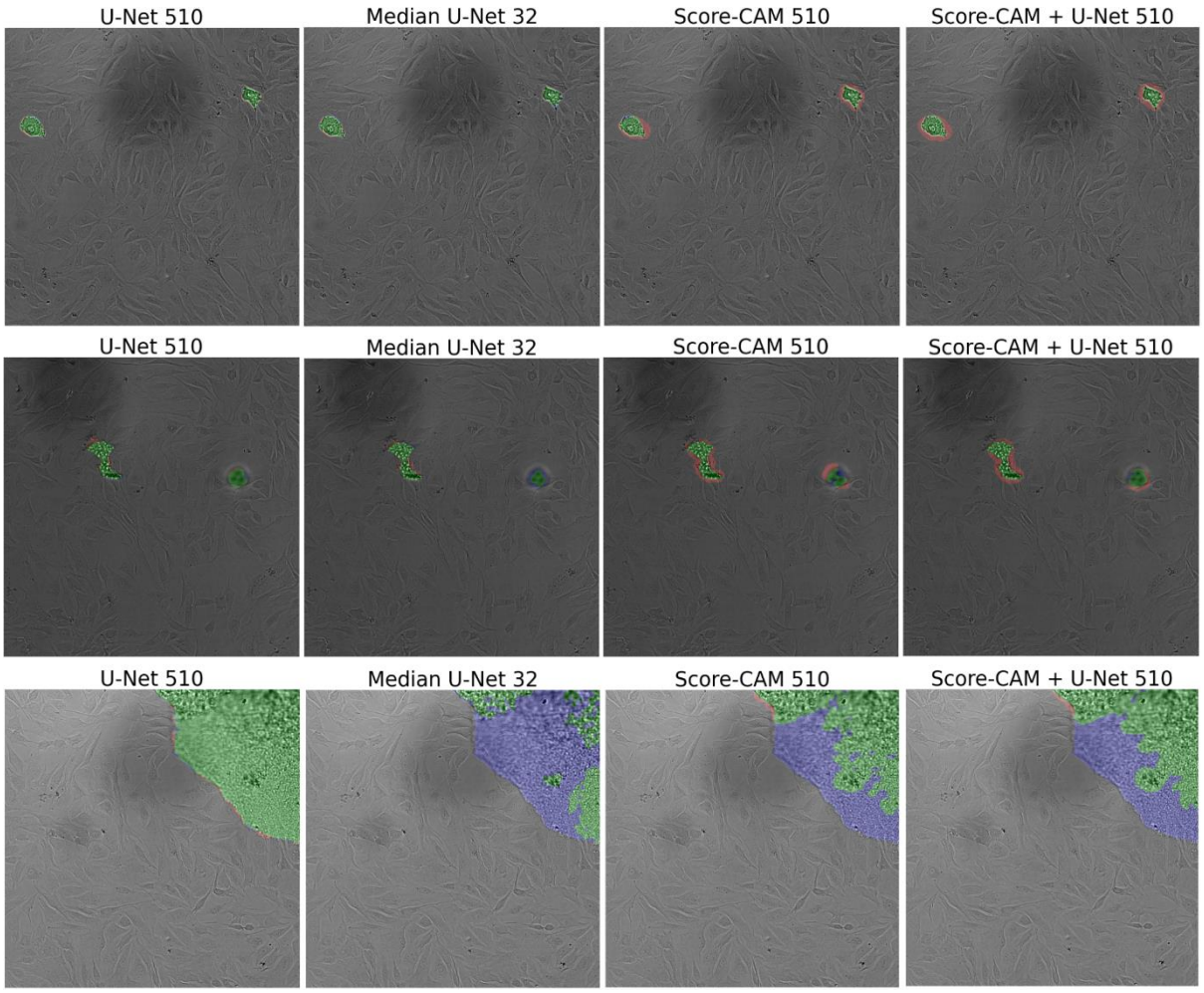


Figure 27. Example artefact segmentation results from the baseline approach and the weakly-supervised Score-CAM and Score-CAM + U-Net approaches. The thresholded predictions of the approaches are overlaid on the original images. The green pixels represent the correctly predicted artefactual regions (true positive), the red pixels show areas where the model predicted an artefact by mistake (false positive), and the blue pixels indicate areas where models failed to predict the underlying artefacts (false negative).

All things considered, Score-CAM and Score-CAM + U-Net performed worse compared to the baseline approach, at least with the number of training images currently available. Based on Figure 26, the pixel-wise IoU scores were still steadily increasing for all approaches. Hence, it cannot be ruled out that the performance of the weakly-supervised approaches could match the performance of the baseline approach if given more images. It is unclear how many images with weak labels should be added to the training set of the weakly-supervised approaches to match (or if it is even possible) the pixel-wise IoU score of the baseline approach.

The analysis also showed that the best Score-CAM and Score-CAM + U-Net model instances were approximately on par with the median U-Net model instance trained on 32 images. Even though these three approaches showed comparable performance, the weakly-supervised approaches can be considered better as the manual annotation does not take as much time. Based on the time estimates given by the expert (6 minutes per pixel-level and 4 seconds per image-level label), the manual data annotation for the weakly-supervised model instances took 34 minutes (510 images * 4s), whereas the manual data annotation for the median U-Net model instance took ~97 minutes (16 contaminated images * 360s + 16 clean images * 4s). The disparity is even bigger between the best weakly-supervised model instances and the best U-Net model instance. The manual data annotation for the best U-Net model instance took 25 hours and 47 minutes (255 contaminated images * 360s + 255 clean images * 4s).

5.5 Ensembled Predictions of the Weakly-Supervised Approaches

We conducted the fifth experiment to see if ensembling the predictions of the weakly-supervised approaches would result in better performance. Ensembling could help reduce the variance of the predictions and thereby result in a smaller generalization error. The ensemble predictions are calculated by averaging the prediction pairs of all of the weakly-supervised approaches used in the previous experiments.

The quantitative results are presented in Table 5 and Table 6. The ensembling proved its effectiveness as one of the ensembled predictions turned out to be better than the predictions of any of the weakly-supervised approaches separately. To specify, the ensemble prediction of AE-SSIM and Score-CAM + U-Net achieved a pixel-wise IoU of 0.5052 and a pixel-wise F1 score of 0.6712, whereas the respective scores of AE-SSIM were 0.3636 and 0.5333, and the respective scores of Score-CAM + U-Net were 0.4756 and 0.6446. None of the other ensemble predictions produced better results than the weakly-supervised approaches separately.

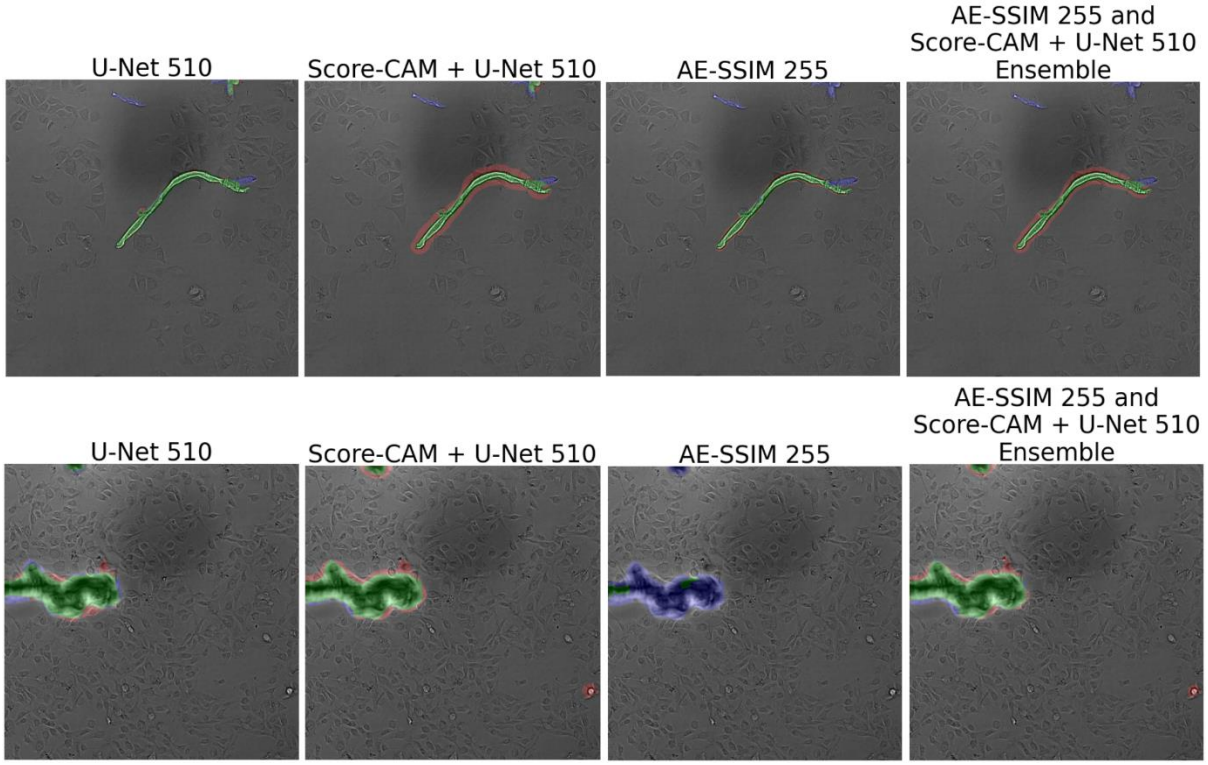
Table 5. The pixel-wise IoU scores of the ensembled predictions. The separate pixel-wise IoU scores of the weakly-supervised approaches are highlighted on the diagonal in gray. The best score is highlighted in yellow.

	Score-CAM 510	Score-CAM + U-Net 510	Patch SVDD 255	PaDiM 255	AE-SSIM 255
Score-CAM 510	0.4661				
Score-CAM + U-Net 510	0.4753	0.4756			
Patch SVDD 255	0.4661	0.4755	0.1889		
PaDiM 255	0.3238	0.4528	0.2060	0.2060	
AE-SSIM 255	0.4614	0.5052	0.3636	0.4460	0.3636

Table 6. The pixel-wise F1 scores of the ensembled predictions. The separate pixel-wise F1 scores of the weakly-supervised approaches are highlighted on the diagonal in gray. The best score is highlighted in yellow.

	Score-CAM 510	Score-CAM + U-Net 510	Patch SVDD 255	PaDiM 255	AE-SSIM 255
Score-CAM 510	0.6359				
Score-CAM + U-Net 510	0.6444	0.6446			
Patch SVDD 255	0.6359	0.6445	0.3177		
PaDiM 255	0.4892	0.6233	0.3416	0.3416	
AE-SSIM 255	0.6314	0.6712	0.5333	0.6169	0.5333

The qualitative analysis demonstrates that the predictions of AE-SSIM and Score-CAM + U-Net rectify each other's mistakes in the ensembled prediction of these approaches. For instance, AE-SSIM helped reduce the overprediction margin of Score-CAM (Figure 28 first row). In other cases, Score-CAM helped segment some of the artefacts missed by AE-SSIM (Figure 28 second row). Although ensembling made the prediction better, the segmenting of the bigger artefacts remained somewhat a problem (Figure 28 third row).



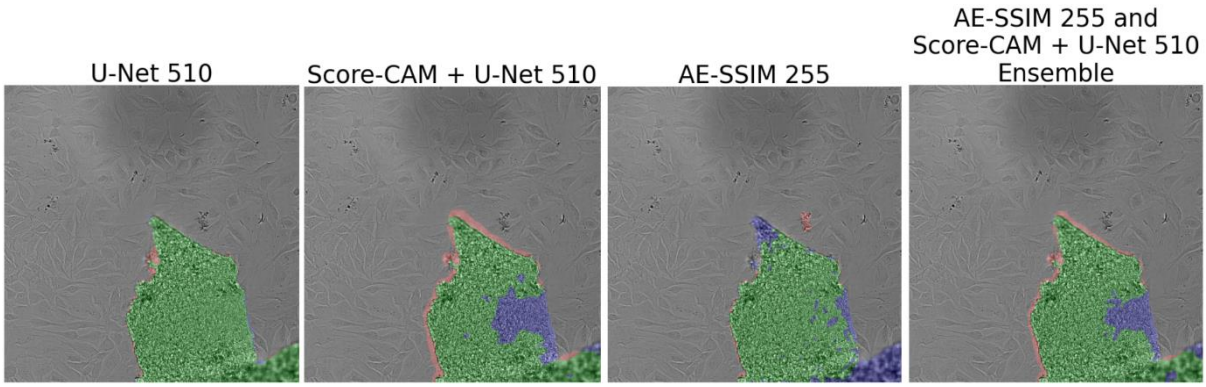


Figure 28. Example artefact segmentation results from the ensembled weakly-supervised approaches where the thresholded predictions are overlaid on the original images. The green pixels represent the correctly predicted artefactual regions (true positive), the red pixels show areas where the model predicted an artefact by mistake (false positive), and the blue pixels indicate areas where models failed to predict the underlying artefacts (false negative).

Ensembling helped construct a prediction from AE-SSIM and Score-CAM + U-Net that returned better results than any other prediction of the tested weakly-supervised approaches. If put into use, a more sophisticated pipeline should be formed to combine the training and inference processes of the two weakly-supervised approaches. Also, even though ensembling helped to improve the prediction, its performance is still worse than that of the baseline fully-supervised approach (Table 7).

Table 7. The quantitative performance comparison of the baseline approach and the best ensemble. The best results are highlighted in yellow.

	Pixel-wise Recall	Pixel-wise Precision	Pixel-wise F1 Score	Pixel-wise IoU	Object-wise IoU
U-Net 510	0.9323	0.9131	0.9226	0.8563	0.2389
AE-SSIM 255 and Score-CAM + U-Net 510 Ensemble	0.6704	0.6721	0.6712	0.5052	0.0454

5.6 Generalizability of Approaches

In the last experiment we explore the generalizability of some of the approaches by applying them out-of-the-box on the images of another dataset. For this, the best performing approaches from the previous experiments (i.e., U-Net, Score-CAM + U-Net, AE-SSIM, AE-SSIM and Score-CAM + U-Net ensemble) are trained on the images of the 7cl dataset and used to segment the artefacts in the images of the LNCaP dataset.

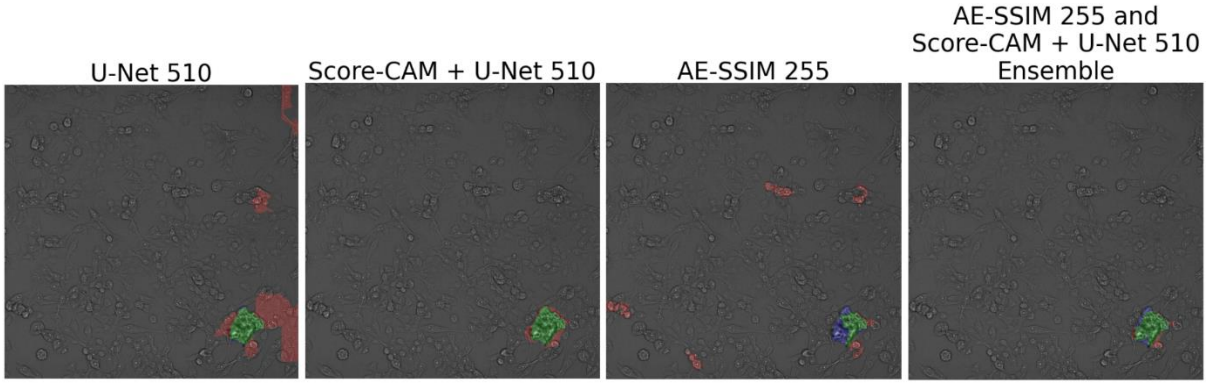
The quantitative results can be found in Table 8. The weakly-supervised approach Score-CAM + U-Net recorded the best results by having a pixel-wise IoU of 0.4721 and a pixel-wise F1 score of 0.6414. The fully-supervised U-Net approach failed to generalize as well as Score-CAM + U-Net,

and only achieved a pixel-wise IoU of 0.0362 and a pixel-wise F1 score of 0.0699. AE-SSIM got the worst results in all but one of the measured metrics.

Table 8. The quantitative results of the fully- and weakly-supervised approaches. The best results are highlighted in yellow.

	Pixel-wise Recall	Pixel-wise Precision	Pixel-wise F1 Score	Pixel-wise IoU	Object-wise IoU
U-Net 510	0.2325	0.0411	0.0699	0.0362	0.0117
Score-CAM + U-Net 510	0.6375	0.6455	0.6414	0.4721	0.0651
AE-SSIM 255	0.0571	0.0538	0.0554	0.0285	0.0021
AE-SSIM 255 and Score-CAM + U-Net 510 Ensemble	0.3243	0.7882	0.4595	0.2983	0.0532

Qualitatively, U-Net produced many false positive predictions as the nuclei in the images of the LNCaP dataset were often predicted as artefacts (Figure 29 first column). U-Net was occasionally also able to predict the artefacts in the images correctly. However, the quantitative results of U-Net were more affected by all of the false positive predictions and thereby, the overall performance was poor. On the other hand, Score-CAM + U-Net was more often than not able to partially segment the artefacts in the brightfield images (Figure 29 second column). The approach seldomly produced false positive predictions, which positively affected the quantitative results.



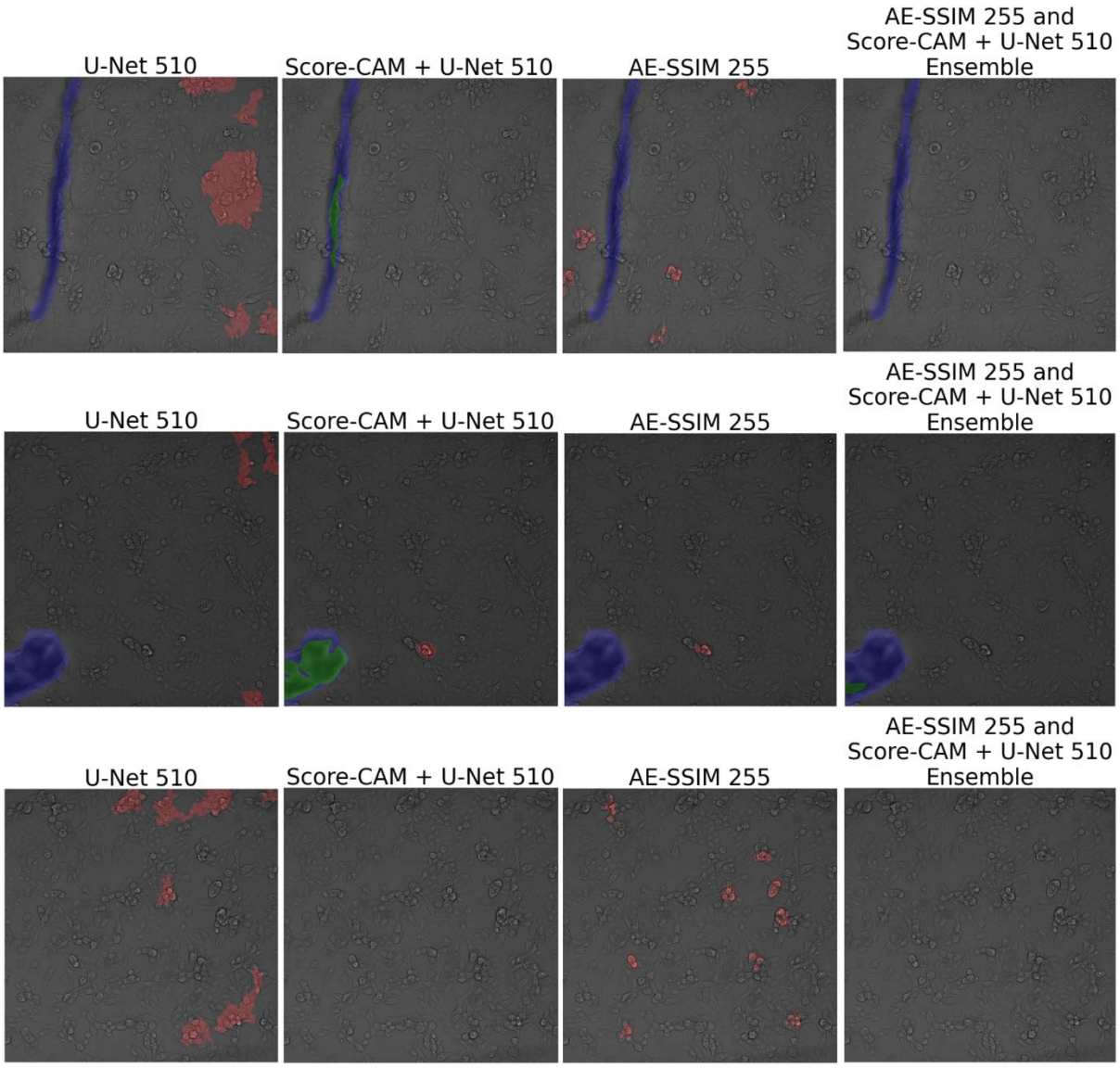


Figure 29. Example artefact segmentation results from the fully- and weakly-supervised approaches where the thresholded predictions are overlaid on the original images. The green pixels represent the correctly predicted artefactual regions (true positive), the red pixels show areas where the model predicted an artefact by mistake (false positive), and the blue pixels indicate areas where models failed to predict the underlying artefacts (false negative).

The quantitative and qualitative results established that the weakly-supervised Score-CAM + U-Net approach generalized best out of the tested approaches. In fact, Score-CAM + U-Net generalized so well on the LNCaP dataset that it was able to match the performance of the same approach when it was trained and tested on the 7cl dataset (Table 9). However, the approach should be tested out-of-the-box on additional datasets before it can be considered a universal approach for segmenting artefacts. None of the other tested approaches could extrapolate and should therefore not be used out-of-the-box on the images of another dataset.

Table 9. The quantitative results of the weakly-supervised Score-CAM + U-Net approach when trained on the 7cl dataset and tested on the 7cl and LNCaP datasets. The best results are highlighted in yellow.

	Pixel-wise Recall	Pixel-wise Precision	Pixel-wise F1 Score	Pixel-wise IoU	Object-wise IoU
Score-CAM + U-Net 510 Tested on 7cl	0.6605	0.6294	0.6446	0.4756	0.0318
Score-CAM + U-Net 510 Tested on LNCaP	0.6375	0.6455	0.6414	0.4721	0.0651

6 Conclusion

The main objective of the thesis was to determine whether weakly-supervised methods are able to segment artefactual regions in brightfield images with performance similar to those of fully-supervised approaches. The conclusion to this question was meant to be reached through the analyses of six experiments.

In the first experiment, we tested the capabilities of two fully-supervised approaches, U-Net and YOLOv5 + U-Net. We concluded based on the results that the two fully-supervised approaches are reliable solutions to the artefact segmentation problem - both approaches correctly predicted most of the artefacts regardless of their size, shape, or structure. However, U-Net performed marginally better and thus was selected as the baseline fully-supervised approach.

In the second experiment, we investigated the potential of segmenting artefacts with the weakly-supervised class activation mapping-based method Score-CAM. Score-CAM performed well but left some room for improvement. Namely, the approach was able to segment the majority of the smaller artefacts, but it was only able to partially segment the bigger artefacts in the brightfield images of the 7cl dataset. Score-CAM was also compared to the state-of-the-art artefact segmentation approaches. The state-of-the-art artefact segmentation approaches were not up to par with the Score-CAM approach as they got considerably worse results in four out of the five measured metrics, with the fifth measure being rather uninformative for such type of experiments.

In the third experiment, we combined the weakly-supervised Score-CAM approach with the fully-supervised U-Net approach. The results indicated that the joint approach of Score-CAM and U-Net could not generalize to the hoped level and was only able to produce marginally better results than the Score-CAM approach. Qualitatively, the predictions of the two approaches were also very similar.

We conducted the fourth experiment to see what performance level of the baseline approach could be reached with the weakly-supervised Score-CAM and Score-CAM + U-Net approaches by varying dataset sizes. The analysis clearly showed that the weakly-supervised approaches could not perform as well as the baseline fully-supervised approach when the models were trained on an equal number of images. However, the best performing weakly-supervised models (trained on 510 images) were performance-wise approximately on par with a baseline model trained on fewer images (trained on 32 images). Thus, it was possible to compare these models based on the time spent to annotate the images to determine which of these approaches was more time-efficient. While the annotation for the 32 images of the baseline approach took ~97 minutes, it took only 34 minutes to annotate the 510 images of the weakly-supervised approaches.

In the fifth experiment, we ensembled the predictions of the weakly-supervised approaches to see if this would result in better performance. Ensembling proved its effectiveness as the combined prediction of Score-CAM + U-Net and the state-of-the-art artefact segmentation approach AE-SSIM returned better results than any other weakly-supervised approach in this thesis. However, the ensembling of the two approaches could not produce a prediction that would segment the

bigger artefacts as a whole. In addition, the performance of the ensembled prediction was still notably worse than the performance of the baseline fully-supervised approach.

In the last experiment, we tested the generalizability of the baseline fully-supervised approach and some of the weakly-supervised approaches by applying them out-of-the-box on a new dataset. The Score-CAM + U-Net approach produced the best results and showed that it was regularly able to at least partially segment the artefacts in the brightfield images. None of the other approaches were able to extrapolate and should therefore not be used out-of-the-box on other datasets.

All things considered, with the number of images currently available for training, the weakly-supervised approaches were not able to segment the artefactual regions in brightfield images with performance similar to the baseline fully-supervised approach. Nevertheless, some of the weakly-supervised approaches returned acceptable performance in our opinion. They could even be the preferred option when there is a limited amount of time to prepare the training data as it is much cheaper to annotate image-level labels than pixel-level labels. In addition, Score-CAM + U-Net showed that the weakly-supervised approaches were able to extrapolate better than the baseline approach and therefore also have a greater potential to generalize on new unseen datasets.

6.1 Limitations

In reality, the usability of the weakly-supervised approaches depends heavily on the images in the dataset. Namely, the Score-CAM and Score-CAM + U-Net approaches cannot be trained on the full-sized images when all of the images in the dataset turn out to be artefactual as the approaches require artefactual and clean images. The same goes for the state-of-the-art artefact segmentation approaches (PaDiM, Patch SVDD, and AE-SSIM), as they need clean images for training. For such datasets, the fully-supervised approaches are still applicable.

6.2 Future Work

This thesis has laid the basis for a future publication. The publication tries to shed light on the artefact problem in brightfield images and demonstrate how they affect downstream analysis. It offers the Score-CAM approach as a possible solution for the artefact problem.

Beyond the publication, there are still some ideas that can be implemented in the future. The first idea was already mentioned in the experiments chapter – acquiring more training data for the weakly-supervised approaches. The additional data might improve the performance and demonstrate how many weak labels are needed to reach the performance of the baseline fully-supervised approach.

Secondly, the weakly-supervised approaches can still be improved. The pipelines are currently based on standard architectures like ResNet50 and U-Net. They were used just for the proof of concept that weakly-supervised approaches can be used to segment artefacts in brightfield images. The next step would be to replace the standard architectures with more advanced ones like EfficientNet (Tan & Le, 2019) or U-Net++ (Zhou, Siddiquee et al., 2018) to improve the results even further.

7 References

Ali, S. (2020, April 23). What is Saliency Map? Retrieved April 25, 2021, from GeeksforGeeks website: https://www.geeksforgeeks.org/what-is-saliency-map/

Auburn University Department of Pathobiology. (2019). BIOL 0509 Histology Lab Introduction II Basic Microscopy. Retrieved January 28, 2021, from Auburn.edu website: https://www.auburn.edu/academic/classes/zy/hist0509/html/02basmic.html

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. Retrieved January 23, 2021, from arXiv.org website: https://arxiv.org/abs/1511.00561

Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., & Steger, C. (2021). The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-020-01400-4

Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., & Steger, C. (2019). Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. https://doi.org/10.5220/0007364503720380

Bilal, A. (2018, January 31). Artificial Neural Networks and Deep Learning - Becoming Human: Artificial Intelligence Magazine. Retrieved January 16, 2021, from Medium website: https://becominghuman.ai/artificial-neural-networks-and-deep-learning-a3c9136f2137

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. Retrieved January 23, 2021, from arXiv.org website: https://arxiv.org/abs/2004.10934?

Boumessouer, A. (2020, September 20). AdneneBoumessouer/MVTec-Anomaly-Detection. Retrieved March 4, 2021, from GitHub website: https://github.com/AdneneBoumessouer/MVTec-Anomaly-Detection

Brownlee, J. (2019, April 16). How Do Convolutional Layers Work in Deep Learning Neural Networks? Retrieved January 19, 2021, from Machine Learning Mastery website: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. *Information*, *11*(2), 125. https://doi.org/10.3390/info11020125

Chan, L., Hosseini, M. S., & Plataniotis, K. N. (2020). A Comprehensive Analysis of Weakly-Supervised Semantic Segmentation in Different Image Domains. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-020-01373-4

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. https://doi.org/10.1109/wacv.2018.00097

Chazotte, B. (2011). Labeling Nuclear DNA Using DAPI. *Cold Spring Harbor Protocols*, *2011*(1), pdb.prot5556–pdb.prot5556. https://doi.org/10.1101/pdb.prot5556

Defard, T., Setkov, A., Loesch, A., & Audigier, R. (2020). PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization. Retrieved February 16, 2021, from arXiv.org website: https://arxiv.org/abs/2011.08785

DeVries, T., & Taylor, G. W. (2017). Dataset Augmentation in Feature Space. Retrieved February 26, 2021, from arXiv.org website: https://arxiv.org/abs/1702.05538

Fishman, D., Salumaa, S.-O., Majoral, D., Peel, S., Wildenhain, J., Schreiner, A., … Parts, L. (2019). *Segmenting nuclei in brightfield images with neural networks*. https://doi.org/10.1101/764894

Gao, F., Wu, T., Li, J., Zheng, B., Ruan, L., Shang, D., & Patel, B. (2018). SD-CNN: A shallow-deep CNN for improved breast cancer diagnosis. *Computerized Medical Imaging and Graphics*, *70*, 53–62. https://doi.org/10.1016/j.compmedimag.2018.09.004

Goyal, M., Knackstedt, T., Yan, S., & Hassanpour, S. (2020). Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities. *Computers in Biology and Medicine*, *127*, 104065. https://doi.org/10.1016/j.compbiomed.2020.104065

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.90

Hosseini-Asl, E., Keynton, R., & El-Baz, A. (2016). Alzheimer's disease diagnostics by adaptation of 3D convolutional network. *2016 IEEE International Conference on Image Processing (ICIP)*. https://doi.org/10.1109/icip.2016.7532332

Intel Corporation, Willow Garage, & Itseez. (2021, February 26). OpenCV. Retrieved April 19, 2021, from OpenCV website: https://opencv.org/

Jensen, E. C. (2012). Types of Imaging, Part 2: An Overview of Fluorescence Microscopy. *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology*, *295*(10), 1621–1627. https://doi.org/10.1002/ar.22548

Jocher, G., Stoken, A., Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, … Ingham, F. (2021). ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration. *Zenodo*. https://doi.org/10.5281/zenodo.4418161

Kayalibay, B., Jensen, G., & van. (2017). CNN-based Segmentation of Medical Imaging Data. Retrieved January 23, 2021, from arXiv.org website: https://arxiv.org/abs/1701.03056

Kent, M. (2004). Advanced Biology. Retrieved January 28, 2021, from Google Books website: https://books.google.ee/books?hl=en&lr=&id=OM3KDwAAQBAJ&oi=fnd&pg=PP1&dq=advanced+biology+Michael+Kent&ots=Fg1LItEzla&sig=YrtytBKtJcj5mseE-hTBpdNkiDM&redir_esc=y#v=onepage&q&f=true

Keras Team. (2015). Keras: the Python deep learning API. Retrieved April 29, 2021, from Keras.io website: https://keras.io/

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Retrieved May 4,

2021, from arXiv.org website: https://arxiv.org/abs/1412.6980

Koppal, T. (2013, October 9). How to Avoid Contamination in the Microbiology Lab. Retrieved April 17, 2021, from Lab Manager website: https://www.labmanager.com/ask-the-expert/ask-the-expert-how-to-avoid-contamination-in-the-microbiology-lab-9882

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., … Dollár, P. (2014). Microsoft COCO: Common Objects in Context. Retrieved January 26, 2021, from arXiv.org website: https://arxiv.org/abs/1405.0312

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance Segmentation. Retrieved February 9, 2021, from arXiv.org website: https://arxiv.org/abs/1803.01534

Marr, B. (2018, December 12). What Is Deep Learning AI? A Simple Guide With 8 Practical Examples. *Forbes*. Retrieved from https://www.forbes.com/sites/bernardmarr/2018/10/01/what-is-deep-learning-ai-a-simple-guide-with-8-practical-examples/?sh=59edbf4a8d4b

McLachlan, G. J. (1999). Mahalanobis distance. *Resonance*, *4*(6), 20–26. https://doi.org/10.1007/bf02834632

Mokobi, F. (2020, May 15). Brightfield Microscope (Compound Light Microscope). Retrieved January 10, 2021, from Microbe Notes website: https://microbenotes.com/brightfield-microscope/

MVTec Software GmbH. (2019). MVTec AD: MVTec Software. Retrieved April 1, 2021, from Mvtec.com website: https://www.mvtec.com/company/research/datasets/mvtec-ad/

nuclearboy95. (2020, October 19). nuclearboy95/Anomaly-Detection-PatchSVDD-PyTorch. Retrieved March 4, 2021, from GitHub website: https://github.com/nuclearboy95/Anomaly-Detection-PatchSVDD-PyTorch

Pallawi. (2019, April 16). AI Starter- Build your first Convolution neural network in Keras from scratch to perform multi-class classification. Retrieved January 21, 2021, from Medium website: https://medium.com/@pallawi.ds/ai-starter-build-your-first-convolution-neural-network-in-keras-from-scratch-to-perform-a059eaa6d4ff

Patel, K. (2019, September 8). Convolutional Neural Networks — A Beginner's Guide - Towards Data Science. Retrieved January 18, 2021, from Medium website: https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022

Pryhoda, O. (2019). *Tissue Segmentation in Histopathological Whole-Slide images with Deep Learning*. Retrieved from website: https://s3.eu-central-1.amazonaws.com/ucu.edu.ua/wp-content/uploads/sites/8/2019/12/Oleksandr-Pryhoda.pdf

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Retrieved January 23, 2021, from arXiv.org website: https://arxiv.org/abs/1506.01497

Restrepo, R. (2017). What is semantic segmentation? Retrieved January 23, 2021, from Ronny.rest website: http://ronny.rest/tutorials/module/seg_01/segmentation_01_intro/

Roh, Y., Heo, G., & Whang, S. E. (2019). A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE Transactions on Knowledge and Data Engineering*, 1–1. https://doi.org/10.1109/tkde.2019.2946162

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Retrieved January 23, 2021, from arXiv.org website: https://arxiv.org/abs/1505.04597

Rosebrock, A. (2016). Intersection over Union (IoU) for object detection - PyImageSearch. Retrieved February 17, 2021, from PyImageSearch website: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Saito, T., & Rehmsmeier, M. (2015). Basic evaluation measures from the confusion matrix. Retrieved February 17, 2021, from Classifier evaluation with imbalanced datasets website: https://classeval.wordpress.com/introduction/basic-evaluation-measures/

Salumaa, S.-O. (2018). Convolutional Neural Networks for Cellular Segmentation. *Dspace.ut.ee*. https://doi.org/http://hdl.handle.net/10062/66166

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved January 22, 2021, from arXiv.org website: https://arxiv.org/abs/1409.1556

Sirinukunwattana, K., Raza, S. E. A., Tsang, Y.-W., Snead, D. R. J., Cree, I. A., & Rajpoot, N. M. (2016). Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images. *IEEE Transactions on Medical Imaging*, *35*(5), 1196–1206. https://doi.org/10.1109/tmi.2016.2525803

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay. Retrieved May 4, 2021, from arXiv.org website: https://arxiv.org/abs/1803.09820

Solawetz, J. (2020, June 29). YOLOv5 New Version - Improvements And Evaluation. Retrieved February 9, 2021, from Roboflow Blog website: https://blog.roboflow.com/yolov5-improvements-and-evaluation/

Stanford University. (2015). CS231n Convolutional Neural Networks for Visual Recognition. Retrieved April 25, 2021, from Github.io website: https://cs231n.github.io/neural-networks-1/

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. Retrieved January 22, 2021, from arXiv.org website: https://arxiv.org/abs/1512.00567

tabayashi0117. (2020). tabayashi0117/Score-CAM. Retrieved April 2, 2021, from GitHub website: https://github.com/tabayashi0117/Score-CAM

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Retrieved January 22, 2021, from arXiv.org website: https://arxiv.org/abs/1905.11946

Thorn, K. (2016). A quick guide to light microscopy in cell biology. *Molecular Biology of the Cell*, *27*(2), 219–222. https://doi.org/10.1091/mbc.e15-02-0088

Tiu, E. (2019). Metrics to Evaluate your Semantic Segmentation Model. Retrieved February 19, 2021, from Medium website: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

Ünver, H. M., & Ayan, E. (2019). Skin Lesion Segmentation in Dermoscopic Images with Combination of YOLO and GrabCut Algorithm. *Diagnostics*, *9*(3), 72. https://doi.org/10.3390/diagnostics9030072

Urban, G., Tripathi, P., Alkayali, T., Mittal, M., Jalali, F., Karnes, W., & Baldi, P. (2018). Deep Learning Localizes and Identifies Polyps in Real Time With 96% Accuracy in Screening Colonoscopy. *Gastroenterology*, *155*(4), 1069-1078.e8. https://doi.org/10.1053/j.gastro.2018.06.037

Wang, C.-Y., Liao, H.-Y. M., Yeh, I-Hau., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2019). CSPNet: A New Backbone that can Enhance Learning Capability of CNN. Retrieved February 9, 2021, from arXiv.org website: https://arxiv.org/abs/1911.11929

Wang, G., & Fang, N. (2012). Detecting and Tracking Nonfluorescent Nanoparticle Probes in Live Cells. *Imaging and Spectroscopic Analysis of Living Cells - Optical and Spectroscopic Techniques*, 83–108. https://doi.org/10.1016/b978-0-12-391857-4.00004-5

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., … Hu, X. (2019). Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. Retrieved February 14, 2021, from arXiv.org website: https://arxiv.org/abs/1910.01279

xiahaifeng1995. (2021, March). xiahaifeng1995/PaDiM-Anomaly-Detection-Localization-master. Retrieved March 4, 2021, from GitHub website: https://github.com/xiahaifeng1995/PaDiM-Anomaly-Detection-Localization-master

Xu, Y., Jia, Z., Wang, L.-B., Ai, Y., Zhang, F., Lai, M., & Chang, E. I-Chao. (2017). Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. *BMC Bioinformatics*, *18*(1). https://doi.org/10.1186/s12859-017-1685-x

Yi, J., & Yoon, S. (2020). Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation. Retrieved February 4, 2021, from arXiv.org website: https://arxiv.org/abs/2006.16067

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Learning Deep Features for Discriminative Localization. Retrieved February 14, 2021, from arXiv.org website: https://arxiv.org/abs/1512.04150

Zhou, Z., Siddiquee, Md Mahfuzur Rahman, Tajbakhsh, N., & Liang, J. (2018). UNet++: A Nested U-Net Architecture for Medical Image Segmentation. Retrieved January 23, 2021, from arXiv.org website: https://arxiv.org/abs/1807.10165

Zordan, M. D., Mill, C. P., Riese, D. J., & Leary, J. F. (2011). A high throughput, interactive imaging, bright-field wound healing assay. *Cytometry Part A*, *79A*(3), 227–232. https://doi.org/10.1002/cyto.a.21029

Appendix

I. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Kaspar Hollo**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Exploring the Value of Weakly-Supervised Deep Learning Approaches for Artefact Segmentation in Brightfield Microscopy Images,

supervised by Mohammed Ali and Dmytro Fishman.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Kaspar Hollo

14/05/2021