Deep learning solutions for domain-specific image segmentation

Vadineanu, S.

Deep Learning Solutions for Domain-Specific Image Segmentation

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op woensdag 8 oktober 2025
klokke 14:30 uur

door

Șerban Vădineanu
geboren te Corabia, Roemenië
in 1995

Promotor:

Prof.dr. K.J. Batenburg

Co-promotores:

Dr. D.M. Pelt

Dr. O. Dzyubachyk

Promotiecommissie:

Prof.dr. M.M. Bonsangue

Prof.dr. T.H.W. Bäck

Prof.dr.ir. B.P.F. Lelieveldt

Prof.dr. K. Lambers

Dr. L. Cao

Dr. J. Wolterink (Universiteit Twente)

This PhD project was conducted as part of the Society, Artificial Intelligence and Life Sciences (SAILS) program of Leiden University.

Ridderprint, The Netherlands

Contents

Chapter 1

Introduction

Image segmentation is a computer vision task where an image is partitioned into multiple segments or regions. The goal is to assign each pixel in the image to a specific object or region, enabling the categorization of different parts of the image, such as distinguishing objects from the background. This can take the form of finding tumours in medical images [9], identifying traffic signs in self-driving cars [78], or environmental monitoring [75]. Initially, this operation would be performed manually by delineating the border of each segment within the image. However, given the tediousness of this process, there has been a significant push for automating it, with thresholding [76], i.e., the categorization of pixels based on their intensity values, being a first step in this direction (see Figure 1.1 for an overview of the main developments in segmentation techniques). Thereafter came methods such as region-based segmentation [164] and clustering [30] which based the categorization of one pixel on the intensities of the neighbouring pixels. With the advent of machine learning, better performance was achieved by hand-crafting image features and learning the segmentation from the data and its associated ground truth [162, 96, 134], commonly referred to as annotations. Currently, deep learning solutions based on convolutional neural networks [106] produce state-of-the-art results [131, 125, 28] in image segmentation, relinquishing the need for hand-crafted features. However, despite their impressive performance, machine learning, and especially deep learning, techniques require vast amounts of annotated images which are often created manually, making the annotation process a persistent bottleneck [8].

Deep learning is a subset of machine learning that uses artificial neural networks to learn patterns from the data. A network consists of multiple layers of neurons, where

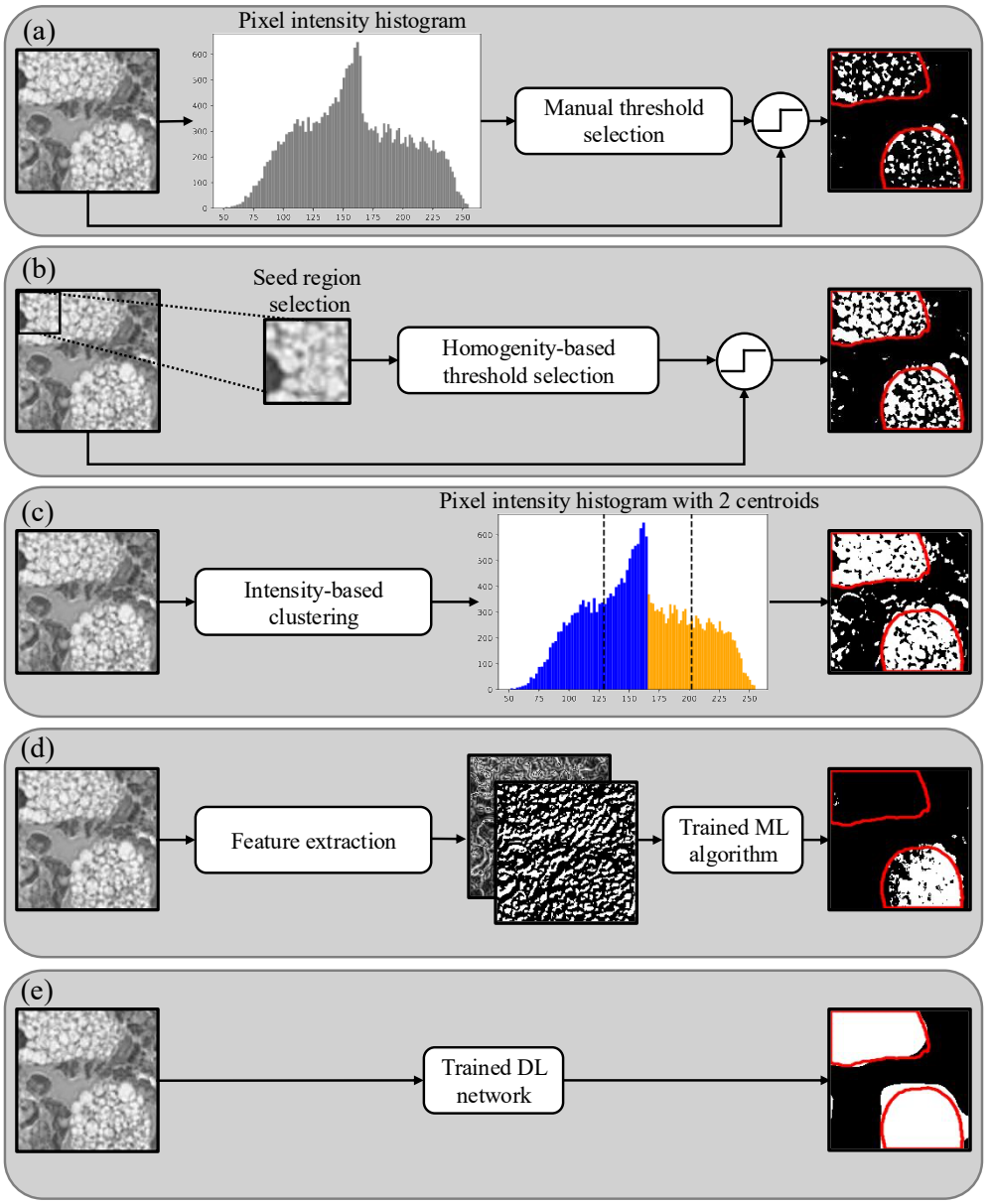


Figure 1.1: The evolution of image segmentation techniques showcased on the segmentation of a cell image. Figure (**a**) shows a segmentation pipeline for manual thresholding, Figure (**b**) corresponds to region-based thresholding, Figure (**c**) to clustering-based methods, while Figures (**d**) and (**e**) illustrate predictions of supervised machine learning and deep learning models, respectively. The red contour corresponds to the ground truth annotation. The input cell image is a crop from Lizard nuclear segmentation data set [59].

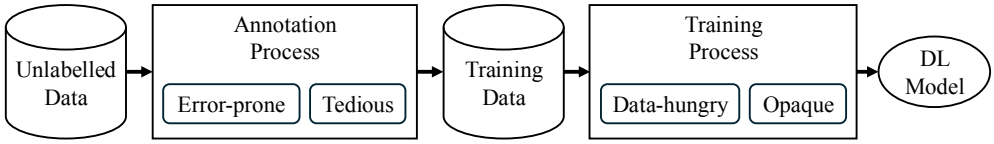


Figure 1.2: Deep learning pipeline from data to trained model and its associated challenges for the human operator.

each layer processes the input data and extracts increasingly complex features. Early layers capture general features such as edges in images, while deeper layers recognize more abstract patterns, for instance, objects. During training, the network requires various examples of input data and ground truth, adjusting its internal parameters to minimize errors. In this way, the network improves its ability to make accurate predictions. In the context of image segmentation, the input data requirement involves the procurement of large collections of images, whereas the annotations consist of pixel-level labels created for every image. A typical supervised deep learning pipeline from the data to the trained model, illustrated in Figure 1.2, consists of an annotation process and a training process.

For segmentation tasks relying on general knowledge, for instance, scene segmentation for self-driving cars [178], the annotation requirement is largely surmounted by the large number of available data sets and by the relative ease with which new data can be annotated, e.g., via crowdsourcing [35]. However, this requirement becomes significantly more demanding in scientific domains. Here, annotations must be created or verified by trained domain experts who are in limited supply. As a result, the adoption of deep learning in these specialized fields progresses more slowly than in general-knowledge domains. To overcome this, it becomes necessary to develop solutions that reduce the burden placed on expert annotators. Considering the deep learning pipeline from Figure 1.2, such solutions can target the annotation process, the training process, or both. On the annotation side, innovative methods are needed to increase the volume of data that can be annotated within a fixed time frame, while preserving their quality. Concurrently, on the training side, there is a demand for networks with transparent learning processes that require less annotated data while still delivering competitive results.

This thesis represents a collection of solutions focusing on streamlining the annotation and training processes for segmentation tasks in two scientific domains with expensive annotation processes, namely cellular imagery and archaeological remote sensing. We tailor methods leveraging the particularities of each domain to increase

the number of available annotations and to train networks at reduced annotation costs. In this chapter, we first describe in Section 1.1 the interplay between the versatility of deep learning solutions and the necessity of expert knowledge, we then provide an overview of the fields of cell imaging and archaeological remote sensing, presented in Section 1.2 and Section 1.3, respectively. In Section 1.4, we then introduce the fundamentals of deep learning methods for computational imaging, with an emphasis on segmentation. Lastly, in Section 1.5, we present the research questions that shape the scope of this thesis.

1.1 Learning Methods and Expert Knowledge

The past two decades have seen a paradigm shift in computational problem-solving, particularly in fields such as image segmentation. Traditionally, each application domain required the algorithms to be tailored specifically to the unique characteristics and challenges of the problem at hand. For instance, segmentation algorithms for medical imaging [5] would differ from those used in autonomous vehicles [50] or environmental monitoring [19]. These domain-specific solutions often demanded significant manual effort and expertise, limiting their adaptability to different fields.

The advent of data-driven techniques, particularly deep neural networks (DNNs), has significantly changed this landscape. Unlike traditional methods, DNNs provide a generic framework for learning features directly from data, without the need for handcrafted rules. This capability allows researchers to develop solutions that are broadly applicable, with minimal customization for specific domains. However, this flexibility comes with a caveat: the success of these methods depends on collaboration between computer scientists and domain experts. Domain experts play an important role in defining the key problems to be solved, curating high-quality data, and interpreting the results generated by the neural networks.

This interplay between generic computational frameworks and domain-specific expertise is exemplified by initiatives such as the Society, Artificial Intelligence and Life Sciences (SAILS) of Leiden University. SAILS is a university-wide interdisciplinary program aiming to disseminate the usage of artificial intelligence throughout the various disciplines within Leiden University. Its projects bring together data, algorithms, and domain experts in collaborative research efforts. This thesis is part of the SAILS initiative, leveraging its multidisciplinary framework to address challenges in image segmentation for cellular imaging and archaeological remote sensing.

1.2 Cell Imaging

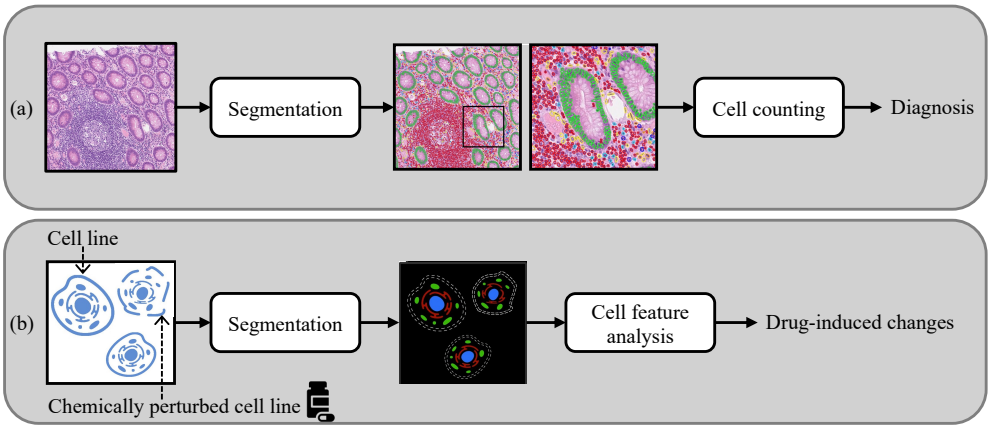


Figure 1.3: Applications of cell segmentation in biomedical research. Figure (a) illustrates the use of segmentation for cell counting, an important step in health diagnostics. Figure (b) demonstrates its application in drug discovery, where segmented cell structures are analysed to assess drug-induced changes. The cell boundary figure and crop in (a) are adapted from [59], and the cell shapes and their segmentations in (b) are from [86].

Cell imaging comprises a set of techniques that enable the visualization and analysis of cellular structures and their dynamics. By monitoring cells' behaviour over prolonged periods of time, researchers can understand how cells react to changes in the local environment or how they respond to various stimuli. This capability is important for advancing efforts in understanding disease pathology and in drug discovery [155] (see Figure 1.3 for an illustrative example). For instance, by tracking the complete blood count from a patient's sample, i.e., counting the white cells, red cells and platelets, the doctors can assess the overall health of that patient [143]. In addition, by tracking the changes appearing in targeted cells, experts can assess the effectiveness of new drugs [86]. For such tasks, the researchers require a good delineation of the cell structures of interest, i.e., cell segmentation, whose manual completion, however, implies tedious work from medical experts. Hence, oftentimes deep learning is perceived as a viable alternative [52]. Given the high annotation demands of deep learning and the challenges in obtaining these annotations—particularly for cell segmentation, where each cell must be individually identified and its shape precisely outlined—many off-the-shelf algorithms, which perform well in domains with abundant annotated data, are not directly applicable in this context.

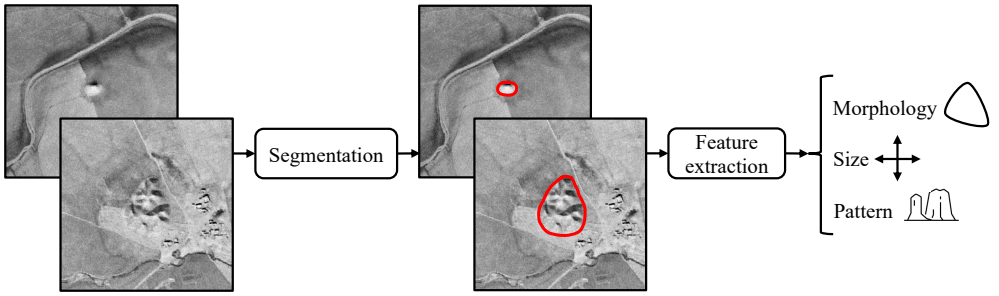


Figure 1.4: The segmentation of archaeological sites as a first step towards a more detailed characterization of the sites within a given area. The aerial images are obtained from CORONA Atlas & Referencing System [26].

1.3 Archaeological Remote Sensing

Archaeological remote sensing is a suite of non-invasive techniques used to detect, map, and analyse archaeological sites and features from a distance, without disturbing the ground. These techniques involve the detection of physical and chemical properties of the Earth's surface, which can indicate the presence of archaeological materials or features such as buried settlements, roads or changes in vegetation patterns caused by human activity [20]. One such task is the identification of settlements from aerial or satellite imagery, which involves the careful analysis of the surrounding landscape before assessing whether a certain image feature represents a settlement. By analysing the distribution patterns of these settlements as well as the local variations in their morphology, the archaeological researchers can gain insights into "the emergence, development, and organization of the first complex human societies." [104]. Here, similarly to cells, the segmentation of the sites can help in extracting morphological patterns which can then be further used to categorize the sites (see Figure 1.4 for an example). Many times, especially in the same geographical area, these settlements share similar visual characteristics, but their widespread distribution makes their manual identification tedious. Here too deep learning can bring considerable advantages by performing the detection of sites in a (semi-)automatic way. Although crowdsourcing is increasingly being used in annotating large archaeological data sets, the involvement of non-experts often leads to issues with data quality [110]. Thus, similar limitations to cell imaging apply here as well with the addition that archaeological research typically receives less funding than the research of medical sciences [151].

1.4 Deep Learning

1.4.1 Machine Learning for Imaging Tasks

Within a machine learning pipeline, an image is typically represented as a real-valued array of pixels $x \in \mathcal{X} \subset \mathbb{R}^{N \times M \times C}$, where $\mathcal{X}$ is the input space, N and M represent the number of rows and columns, respectively, and C corresponds to the number of channels. Coloured images have $C = 3$, corresponding to the red, green and blue channels, whereas grayscale images contain only one channel. The goal is to find the mapping $f : \mathcal{X} \to \mathcal{Y}$ from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$. Machine learning algorithms approximate this mapping with a parametrized function f_δ, whose parameters δ are learned from the set of training images $X \subset \mathcal{X}$ paired with the corresponding expected output $Y \subset \mathcal{Y}$. Unlike the input space $\mathcal{X}$, whose shape is generally fixed, the shape of the output varies depending on the imaging task. For instance, for image classification, where the goal is to categorize images in k different classes, the output space becomes $\mathcal{Y} = \mathbb{R}^k$. Here, the model learns to predict a vector of k probabilities with the highest one providing the predicted class. When it comes to image segmentation with k classes (segments), $\mathcal{Y} = \mathbb{R}^{N \times M \times k}$. Similarly to classification, a probability vector is produced with probabilities for each of the k classes. However, in this case, a classification vector is generated for each pixel of the input image.

1.4.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) have experienced widespread adoption in recent years, owing to their state-of-the-art performance and remarkable versatility across various imaging tasks [91, 93]. This type of neural network is comprised of filters organized in layers which are applied to the input image generally in a sequential manner, i.e., the output of a set of filters (layer) becomes the input to the next. Each layer i creates an intermediate image $z_i \in \mathbb{R}^{N_i \times M_i \times C_i}$, called a feature map, that reflects the importance of certain image features. The shape of z_i can change depending on the desired spatial dimensions and the number of filters (C_i) that the layer contains. The feature map z_i^j from filter j of layer i is obtained as

$$z_i^j = \sigma\left(\sum_{l=0}^{c_{i-1}} (z_{i-1}^l \circledast h_{ij}^l) + b_{ij}\right), \tag{1.1}$$

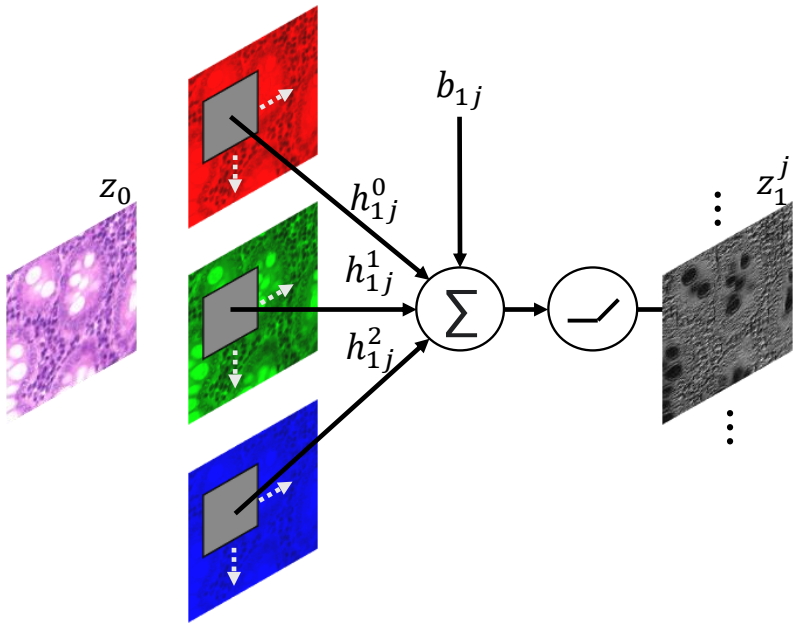


Figure 1.5: Example of how a feature map is obtained for a filter j in the first layer of a CNN with a coloured image as input. Each of the three channels of the input is convolved with a corresponding set of weights of the filter, the results are summed, a bias term is added and a non-linear function is applied to the summation result. This process is repeated for every filter of the layer. The input cell image is a crop from Lizard nuclear segmentation data set [59].

where each channel of the previous feature map z^l_{i-1} is convolved with a corresponding set of weights of the filter $h^l_{ij} \in \delta$ and the results are summed across all channels of the previous layer. After adding the bias term $b_{ij} \in \delta$, an activation function σ, oftentimes the rectified linear unit (ReLU) [2], is applied to introduce non-linearity, allowing the network to capture more complex patterns. A schematic representation of this process is presented in Figure 1.5.

In a supervised setup, in order to estimate the CNN parameters $\widehat{\delta}$, a training step is performed wherein the CNN's predictions of a set of input images, called the training set X, are compared against the ground truth output Y via a loss function $L : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The goal is to find the optimal parameters

$$\widehat{\delta} = \underset{\delta}{\operatorname{argmin}} \sum_{(x,y)\in X\times Y} L(f_\delta(x), y) \tag{1.2}$$

that minimize the loss. Due to the high number of parameters and the non-linearity

introduced by the activation functions, analytical solutions are difficult to apply to CNNs. Hence, the optimization step usually involves an optimization algorithm, such as ADAM [82], which iteratively adjusts the parameters by using the partial derivatives of L with respect to them. Given that the training step is performed on a finite, and often small, subset of $\mathcal{X}$, a good fit on the training data does not guarantee the same performance on unseen images. Thus, to avoid such discrepancy, referred to as overfitting, a separate set, called validation set, can be employed. In this way, the optimization is still performed on the training set, while the performance of the CNN will be evaluated on the validation set, with the parameters being updated only if this results in a lower error on the validation set.

1.4.3 Challenges for the Human Operator

Although, when successfully trained, deep learning networks offer substantial benefits in automating image segmentation, obtaining these advantages comes with challenges for the human operators both during the annotation and training processes. These challenges are visually represented in Figure 1.2.

Firstly, not only is the large-scale annotation of images for segmentation a time-consuming task but it can also result in inconsistencies being introduced due to, for instance, fatigue, low image quality, or the ambiguity of the segmented structures. In many specialized domains, there can be a lack of consensus between experts when it comes to defining the category of an object, e.g., whether it is an archaeological site or a hill. Moreover, disagreements can also appear when defining the boundary of structures, for instance, when delineating cell nuclei from the surrounding cytoplasm. Thus, the effort required in annotating a data set and the potential errors it may contain pose significant apriori challenges to the deployment of deep learning by the human operator.

Secondly, training deep learning networks is a process that typically is susceptible to overfitting which can be caused by insufficient training data. Thus, an already annotated set of images does not guarantee a successful training process. Additionally, even after being successfully trained, the adoption of deep learning is challenged also by the opacity of its decision process. Due to the large number of parameters (generally, in the order of millions) involved in generating the output of deep learning networks, the steps undergone for producing it cannot be traced in a way comprehensible to humans. This can hinder trust and limit the integration of deep learning in less technical fields such as archaeology.

1.5 Research Questions

In this thesis, we address these challenges by focusing on segmentation tasks where the integration of standard deep-learning networks is suboptimal due to the limited availability of annotated samples. We propose solutions targeting both the annotation and the training processes with applications in cellular imaging and archaeological remote sensing.

An overview of the chapters of this thesis together with the research questions that they answer is presented below.

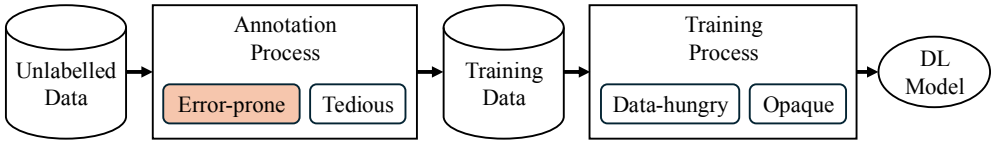


Figure 1.6: Our first research question examines the consequences of errors in the annotation process.

Research question 1. *How do different types of annotation errors impact the performance and robustness of deep learning models in the task of cell segmentation?*

It is commonly assumed that, when training deep learning networks, errors in annotations can severely degrade their performance, given the sensitivity of these networks to input quality (emphasized in orange in the deep learning pipeline in Figure 1.6). The annotations for cell segmentation are susceptible to errors which can be difficult to find and correct. Thus, understanding the specific impact of various annotation errors would allow us to develop more robust deep-learning networks or decide which inconsistencies require the most care to be prevented.

In Chapter 2, we analyse the impact of annotation inconsistencies on deep-learning-based cell segmentation. We introduce perturbations (see Figure 1.7) to emulate errors typical to the annotation of cells and we measure how they affect the performance of different network architectures designed for segmentation.

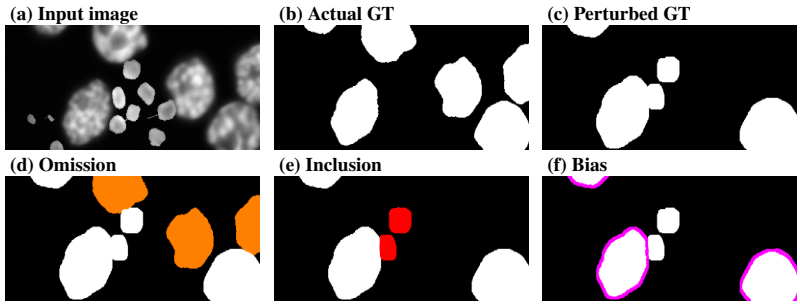


Figure 1.7: Example of the perturbations we perform. Figures (**a**, **b**) were generated with a virtual microscope from [166].

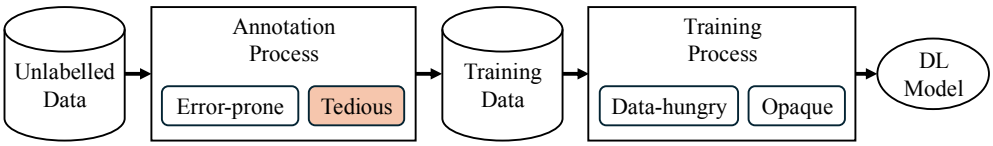


Figure 1.8: Our second research question focuses on reducing the human effort required for producing annotations.

Research question 2. *To what extent can we reduce the human effort required for cell segmentation annotation by using a convolutional neural network to improve lower-quality annotations?*

Given the labour-intensive nature of cell segmentation, reducing manual annotation effort is crucial for expanding deep-learning applications in biomedical fields (emphasized in orange in the deep learning pipeline in Figure 1.8). One promising approach is to relax the strict quality standards traditionally applied to annotations, thereby enabling a greater volume of annotated samples within a fixed time frame. However, these lower-quality annotations may not be immediately suitable for direct training of segmentation networks and may require refinement to be fully effective during training.

In Chapter 3, we propose a solution to enhance the annotation process by reducing the human effort in training deep learning algorithms. We achieve this by automatically enhancing the quality of noisy annotations, produced with low effort. We propose a learning pipeline in which a CNN is trained to upgrade low-quality annotations. For

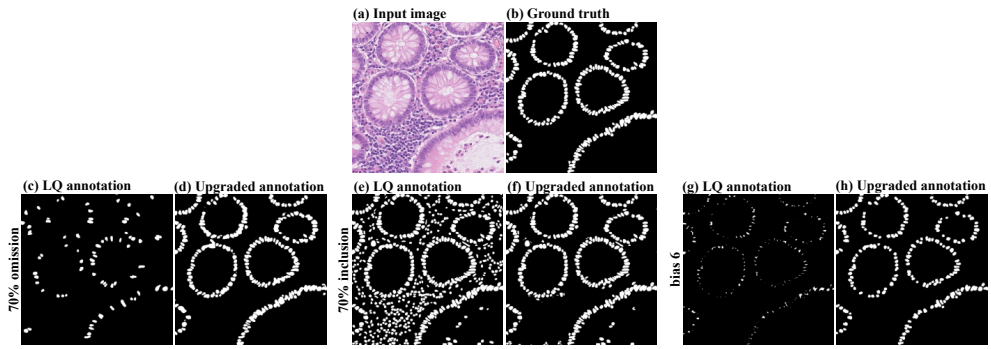


Figure 1.9: Example of perturbed annotations and their corresponding upgraded versions. Figures (**a, b**) are from [59].

this, we employ a small set of well-annotated samples whose annotations we perturb, similarly to Chapter 2, such that the CNN can learn a mapping from different versions of low-quality annotations to high-quality ones. We then use the initial set with high-quality annotations together with the upgraded noisy annotations to train segmentation networks on this larger combined set. In Figure 1.9, we show different types of perturbations applied to annotations together with the results after applying the upgrading CNN.

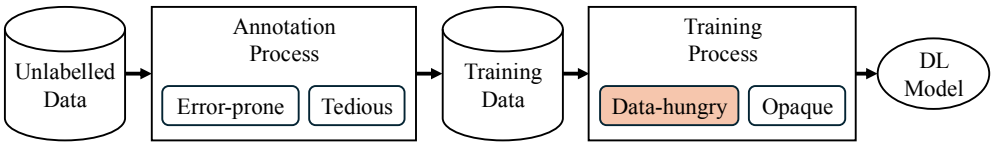


Figure 1.10: Our third research question is aimed at reducing the data requirements for training deep learning models.

Research question 3. *To what extent can the few-shot learning paradigm be effectively applied to cell microscopy image segmentation?*

An alternative approach to reducing the human effort in applying deep learning to specialized domains is to design algorithms that require less data than traditional CNNs (emphasized in orange in the deep learning pipeline in Figure 1.10). Few-shot learning, which aims to perform tasks with minimal labelled data, shows promise in addressing data scarcity in cell microscopy. However, its applicability to cell segmentation is uncertain due to particularities inherent to cell imagery compared to natural images where few-shot learning is more commonly applied.

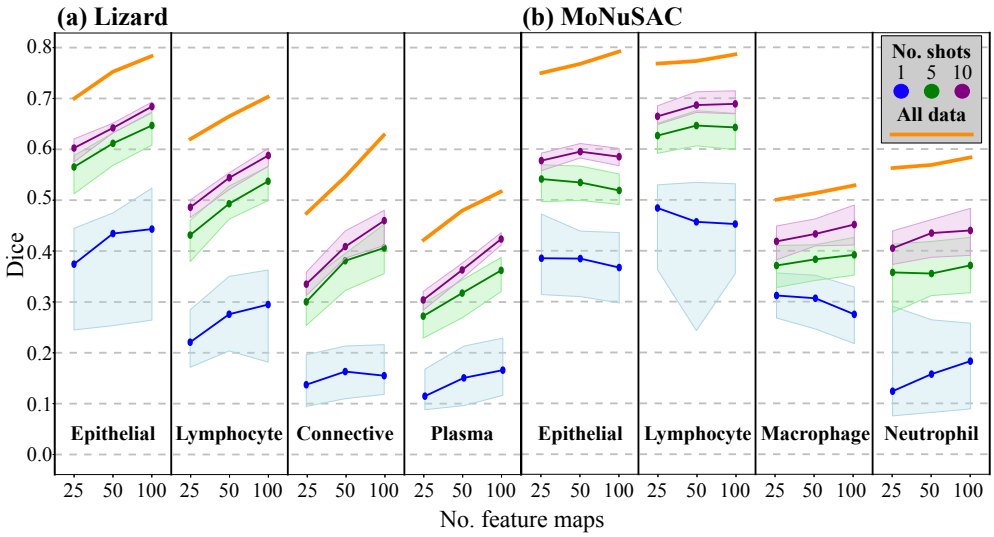


Figure 1.11: The quality of the segmentation against the number of shots and the number of features used. The orange line shows the average Dice score on the test set of the models trained on all labelled data. The points represent the median value across 50 experiments, while the shaded area is defined by the first and third quartiles.

In Chapter 4, we target the training process by proposing a new few-shot technique specifically tailored for cell segmentation. We leverage existing annotations for certain classes of cells to train feature extractor networks, which we then use to segment new cell classes using low amounts ($\leq$ 10) of annotations of the new class, called shots. While designing our few-shot algorithm, we consider requirements specific to cell segmentation such as the need for precise delineation between the cell instances and the relative similarity between the structures that are present in cell images. We study the performance of our algorithm on two data sets, each with four cell types, with the results shown in Figure 1.11. The graph shows the segmentation performance against the complexity of the feature extractors, with promising results for as little as 5 annotated images.

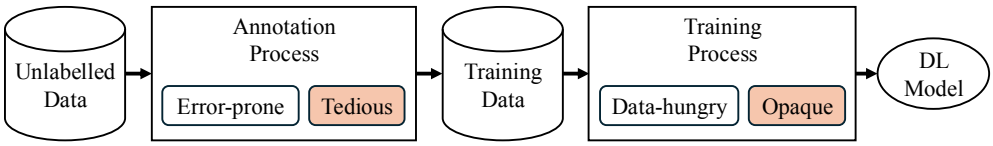


Figure 1.12: The final two research questions respectively tackle the tediousness of the annotation process and the opacity involved in training deep learning networks.

Research question 4A. *What insights into the learning process of different network architectures can be obtained from analysing activation maps in the context of archaeological site classification in Upper Mesopotamia?*

Research question 4B. *To what extent can activation maps be used as sources of annotation for site segmentation?*

In addition to the difficulty of creating annotated data sets, another challenge associated with deep learning in scientific domains is the relative reluctance with which these algorithms are perceived (emphasized in orange in the deep learning pipeline in

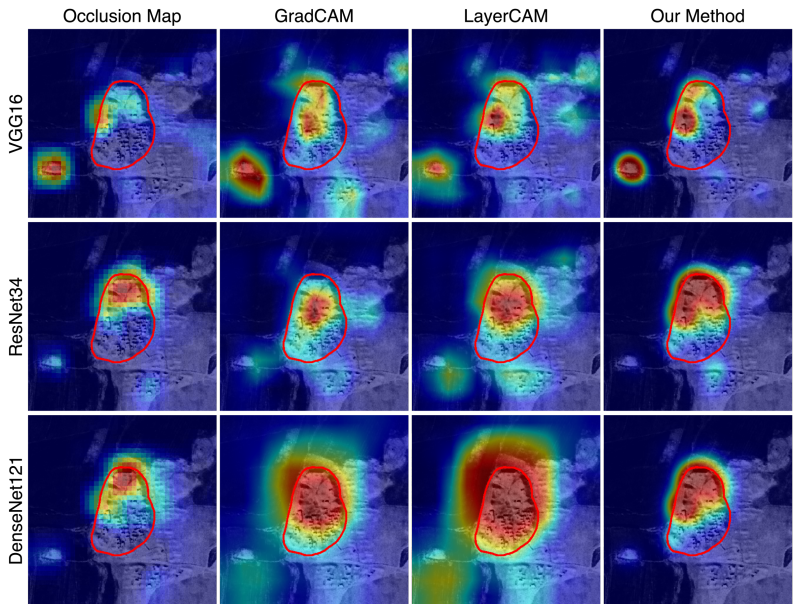


Figure 1.13: The activation maps of 3 CNNs (rows) given by 3 explainability techniques and our method (columns). The red line shows the expert delineation of the site. The site image is from [26].

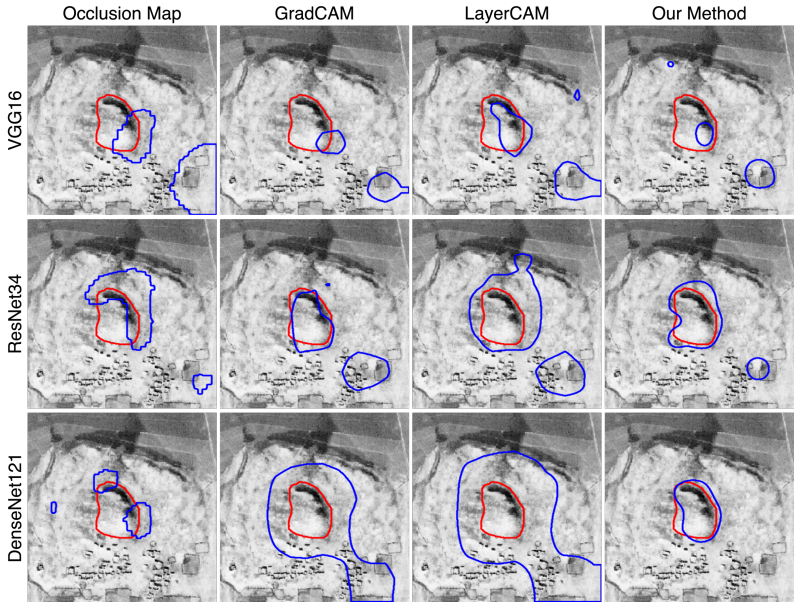


Figure 1.14: Site boundaries derived from the output of the explainability techniques (in blue) compared to the expert annotations (in red). The site image is from [26].

Figure 1.12). Due to their lack of traceability, deep learning predictions may not be trusted, particularly when the operator is unfamiliar with the neural network architecture and its operating principles. This reluctance is especially present in fields such as archaeology, which, as part of the non-exact sciences, relies heavily on interpretation and contextual understanding.

In Chapter 5, we tackle in tandem both problems associated with the usage of deep learning in archaeological research: the difficulty in explaining the results a CNN produces and the high cost of creating annotations for segmentation tasks. We explore how explainability techniques can enhance model interpretability and reduce annotation costs by applying these methods to three deep-learning architectures. We train the networks to classify whether an archaeological site is present in an image, a task for which the annotations are relatively cheap to produce. We then employ the activation maps, i.e., the output of the explainability techniques shown in Figure 1.13, both as sources of annotations (see Figure 1.14) for site segmentation and to analyze the visual cues that influence the networks' predictions. In addition, we develop a new method for creating activation maps specifically designed to produce accurate boundaries for this type of site.

Chapter 2

Annotation Errors in Cell Segmentation

2.1 Introduction

Image segmentation, i.e., the labelling of relevant features in images, has been an important topic for the computer vision community [62]. In recent years, the use of deep convolutional neural networks for image segmentation has become increasingly popular [3]. Although such algorithms are able to achieve similar performance to human annotators on certain tasks [51], they are heavily dependent on both the quantity and the quality of the training data. The importance of quality is especially prominent in the context of segmentation, where the annotation process is time-consuming and often requires domain-expert knowledge (e.g., in medical imaging). One important issue that arises is the high variability between expert annotators when segmenting anatomical structures from medical images [13, 94, 175]. For instance, the segmentation of multiple sclerosis poses difficulties to many experts since the lesion area can vary in size, shape or location [174], inducing high inter- and intra-observer variabil-

This chapter is based on:

> Ș. Vădineanu, D. M. Pelt, O. Dzyubachyk, and K.J. Batenburg. "An analysis of the impact of annotation errors on the accuracy of deep learning for cell segmentation". *International Conference on Medical Imaging with Deep Learning (pp. 1251-1267)*. PMLR (2022).

ity [22]. Also, there can be a considerable amount of disagreement between experts when defining the segmentation border of the optic nerve head in retinal images [49]. These annotation dissimilarities can mean that the manually annotated labels used for segmentation may deviate from the ground truth, which can negatively impact the accuracy of the supervised machine learning models.

In order to compensate for such inconsistencies, various label fusion techniques, e.g., STAPLE [165], VoteNet [47], have been proposed to extract an approximation of the ground truth from multi-expert annotations. However, such methods often require multiple opinions for the same data, a process that is costly and slow. In addition to the effort that the research community is putting into alleviating the label inconsistency issue, it is also important to study the actual impact that such label imperfections are causing to the segmentation algorithms. The benefits of such a study are twofold. Firstly, the engineers who use existing deep learning solutions when developing tools would learn whether they can reduce the expert time on annotations by admitting lower quality labels and still achieving the desired results. Secondly, the developers of deep learning techniques can be provided with insights indicating ways to design more robust algorithms with respect to annotation errors.

While the literature proposes multiple methods to mitigate the effect of annotation errors in image segmentation, there are few works evaluating the concrete implications of these errors. In particular, [179] develop a measurement of label quality in the context of semantic segmentation of synthetic urban street view scenes. They apply various levels of simplifications to the segmentation masks of the scene and use a modified version of FusionNet [120] and FCN16 [131] to generate the predictions. Their results emphasized the need for a large set of coarsely annotated images rather than strongly controlling the label quality. However, the study assumes the immediate availability of a large pool of unannotated images with an inexpensive coarse annotation process, which is often impossible to achieve in medical imaging, where even creating coarse labels requires a certain extent of expertise. [67] emulated three types of perturbations on a liver segmentation data set [17]. The errors included the application of random offsets, shifts and flips of pixel labels applied to the annotation images, while the evaluation was performed for UNet [125], SegNet [11] and FCN32 [131]. The selection of errors was further diversified by [156] with their work on an MRI brain tumour data set [68]. They made use of elastic transformations, random crops of the tumour area, constant shifts and random permutations between slices and their labels. Consequently, they observed the effects of the perturbations for multiple learning paradigms based on a UNet backbone. Both studies introduce errors present-

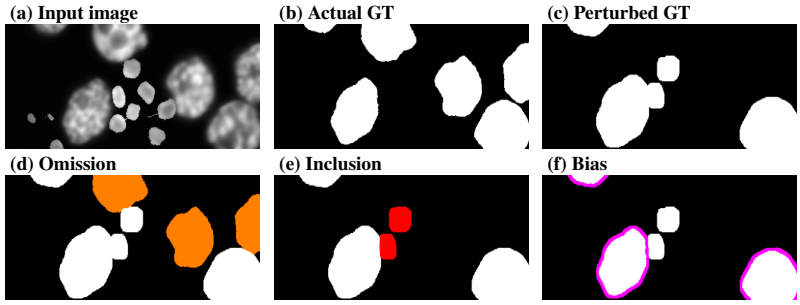


Figure 2.1: Example of our proposed perturbations. Figure (**a**) shows an input image where the HL60 cells are the target objects and the granulocytes form the background objects. The unmodified ground truth is shown in (**b**), and its perturbed version in (**c**). The errors are highlighted in (**d**) – omission/orange, (**e**) – inclusion/red and (**f**) – bias/purple.

ing plausible occurrence scenarios. However, each of them is performed on a single manually-annotated data set, whose labels can already be subjected to the errors the authors try to model.

In this chapter, we extend previous works by introducing three error-emulation techniques applied to three different data sets. So far, the current annotation error studies on biomedical images have been focusing on segmentation tasks of unitary objects (e.g., organs, tumours). Such objects are limiting most error emulation approaches to create perturbations only at the border of the object's label. We deviate from this approach by proposing an analysis of sparsely distributed objects in the context of cell segmentation. This enables us to not only induce errors at the border of the objects, but also to emulate errors concerning entire regions, such as the complete removal or addition of cells. In addition, we address the shortcomings of using manual annotations as ground truth by employing two perfectly-annotated synthetic data sets of HL60 and granulocytes [140] and validate our observations on manually-annotated microscopy images [153]. Moreover, we expand the current analysis by incorporating a network whose architecture diverges from the usual encoder-decoder paradigm.

2.2 Background and Methodology

Our analysis is focused on the segmentation task of 2-dimensional vector-valued (e.g., RGB) images, denoted as arrays of pixels $x \in \mathbb{R}^{N \times M \times C}$, where N, M, C represent the number of rows, columns and channels, respectively. The aim is to find a mapping from x to an output $y \in \mathbb{Z}^{N \times M}$ that subdivides the image into disjoint sets of pixels,

each set corresponding to a certain category. In our work, we address the problem of binary cell segmentation by separating only one class of objects from the background. Suppose the image x contains E cells. For each cell i we define the cell label l^i as the binary image in which the pixels belonging to that cell are set to one and all other pixels set to zero:

$$l^i_{nm} = \begin{cases} 1, & \text{if } x_{nm} \text{ belongs to cell } i \\ 0, & \text{otherwise} \end{cases} \qquad \forall\ 1 \leq n \leq N,\ 1 \leq m \leq M. \tag{2.1}$$

Given the set of all cell labels $\mathcal{L} = \{l^1, l^2, \ldots, l^E\}$, a target image for training can be constructed by:

$$y = \sum_{l \in \mathcal{L}} l \tag{2.2}$$

In order to approximate the desired mapping, we employ convolutional neural networks (CNNs) by passing the input image through a series of successive operations, called layers. The networks are given a set of input images $X = \{x_1, x_2, \ldots, x_{N_t}\}$ and the predicted output $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{N_t}\}$ is compared against the target output $Y = \{y_1, y_2, \ldots, y_{N_t}\}$ with the goal of minimizing a loss function.

Annotation errors. As the true output usually comes from manual annotation, it becomes subjected to human errors, which can hinder the training of CNNs. We model such inconsistencies and separate them into three categories (shown in Figure 2.1) as follows:

Omission Errors. Typical stained tissue scans can include tens or even hundreds of cells of different shapes and sizes [46]. When creating segmentation masks for such diverse and populated images, it is possible that an expert annotator may unintentionally ignore a certain proportion of the relevant cells. We call such absence of cell annotations omission errors and we develop a systematic method of altering the ground truth mask by removing a ratio of the present cells from the label set $\mathcal{L}$. An example of cell removal is showcased in Figure 2.1(d). We define $\mathcal{L}^{\mathrm{S}} \subseteq \mathcal{L}$ as a random subset of size $S \leq E$, where S is chosen to satisfy the omission rate $r_\omega = \frac{S}{E}$. The label after omission is comprised of the binary labels corresponding to the remaining cells $y^\omega = \sum_{l \in \mathcal{L} \setminus \mathcal{L}^S} l$.

Inclusion Errors. Another issue that can arise in tissue scans is the accidental annotation of cells belonging to the wrong category. In such cases, an annotator might sometimes include some fundamentally different cells due to their proximity or apparent resemblance to the correct cells. We incorporate this inclusion error into

our analysis with various amounts of severity, which correspond to the amount of "wrong" cells that we choose to include in the label set. One such case is presented in Figure 2.1(e). We define $\Lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^F\}$ as a set of binary labels for other objects within x and $\Lambda^S \subseteq \Lambda$ as a random subset of size $S \leq F$, where S is chosen to satisfy the inclusion rate $r_\phi = \frac{S}{F}$. The resulting subset is then added to the label set $\mathcal{L}$ before creating the final label $y^\phi = \sum_{l \in \mathcal{L} \cup \Lambda^S} l$.

Bias Errors. Another important factor that deserves attention is the ambiguity that is often present when delimiting the cell borders. Often, it is difficult for annotators to precisely distinguish the true outline of cells. This can lead to annotations that deviate from the gold standard (ground truth), inducing bias into the data. Such biases can manifest in the form of creating cell labels that excessively cover the actual cell surface, as can be observed in Figure 2.1(f). Moreover, the opposite can also happen, when the annotator "shrinks" the corresponding label relative to the true area of the cell. We consider both cases in our study and we also control the amount of bias we introduce by expanding and reducing the sizes of the cell labels that are present in our data sets. In order to model the annotation bias, we employ morphological operations [130]. Specifically, we simulate excessively covering cells by applying a dilation operation $\oplus$ to the target image y a number of q times, where q is randomly chosen between 1 and $q_{\max}$. Similarly, we simulate the shrinking of cells by applying an erosion operation $\ominus$ q times, where q is randomly chosen between 1 and $q_{\max}$.

2.3 Experiments

2.3.1 Experimental Setup

In this work, we considered three types of convolutional neural networks based on their wide adoption and distinctive characteristics. Our selected networks include:

- **UNet** [125] – encoder/decoder architecture, decoder with transposed convolutions, direct connections between the encoder and decoder;
- **SegNet** [11] – encoder/decoder architecture, decoder with unpooling, no connections between the encoder and decoder;
- **Mixed-scale dense network** (**MSD** [116]) – densely connected architecture, dilated convolutions.

We performed our experiments using PyTorch [114] implementations of our chosen network architectures, while keeping their structure, e.g., the number of layers, similar

to their original implementation. For our two-class segmentation problem, the true output will be a two-channel image, where a pixel on the first channel is 1 if it corresponds to a pixel of the background and 0 otherwise, while the reverse is true for the second channel. A soft-max activation is used on the output of the final layer, while all intermediate layers are paired with a ReLU function. We aim to minimize the Dice loss by using ADAM optimizer [82] while training the network for 20 epochs on the synthetically-generated data and for 50 on the manually-annotated images, the latter epoch count being larger due to the increased complexity of the images. After each epoch, the model is tested on a validation set selected as a separate portion of 30% from the training data and the model with the lowest validation score is kept. Our qualitative metric is the Sørensen–Dice coefficient, which we compute for the entire test set and average over 10 runs. Moreover, whenever a network reaches an untrainable state, i.e., it only segments the background, we discard the model and restart training with a different initialization. The performance comparisons were validated using Wilcoxon tests [124].

2.3.2 Synthetic Data

These experiments were conducted on simulated microscopy images of HL60 nuclei cells and granulocytes [140]. The images obtained from the Masaryk University Cell Image Collection[1] were generated by a virtual microscope [166]. An image-label sample pair for each data set is shown in Figure 2.2. The different sizes and position distributions of the two cell types make them good candidates for our analysis since the same generated perturbation can affect them differently. This will enable us to apply our observations to a broader variety of cells. For each category of cells, the data set consists of 30 volumes, each volume being separated into 129 slices of 565×807 16-bit pixels. We used 25 volumes for training, while 5 were kept for testing. We selected the slices that had a non-empty label, resulting in an average number of 84 slices per volume. For these data sets, we assume having one annotator per volume, thus, we emulate the errors once per volume.

Omission errors. We perform the omission for $r_\omega \in \{10\%, 20\%, 30\%, 50\%, 70\%\}$. The results presented in Figures 2.2(e,h) show that this category of errors has limited impact on the networks' performance when we consider moderate cases ($r_\omega \leq 30\%$). MSD and UNet show similar robust behaviour to moderate omissions, while SegNet presents a pronounced downward trend, with a 10% reduction in Dice score for 30%

[1] `https://cbia.fi.muni.cz/datasets`

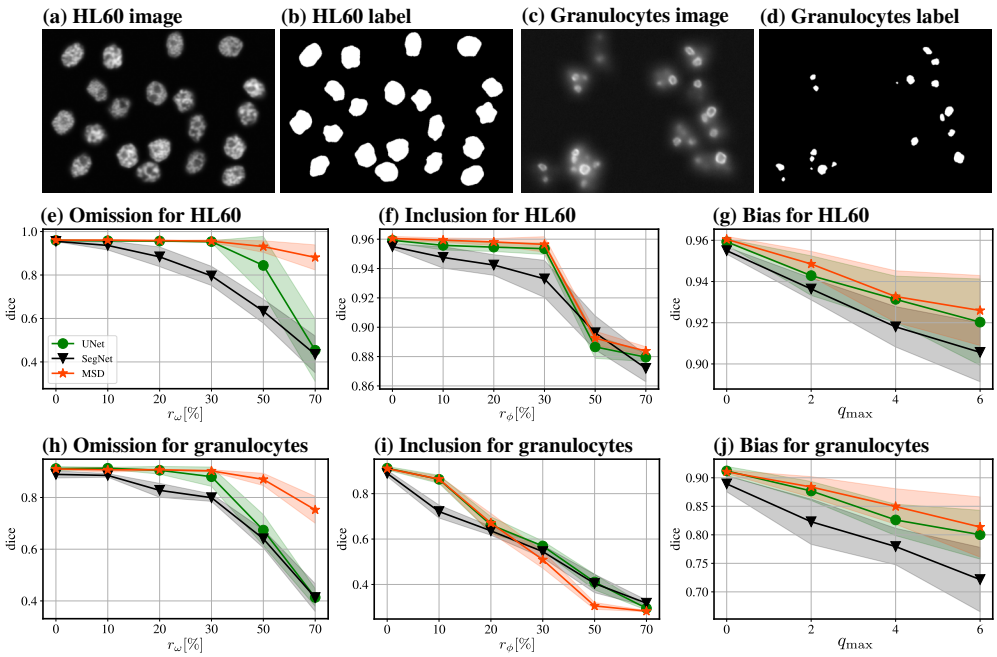


Figure 2.2: Example images and results for the synthetic data sets. Figures (**a**,**b**) and (**c**,**d**) show image/label pairs of simulated microscopy slices. (**e**–**j**) show the Dice score of trained networks on the test set as a function of perturbation severity, for HL60 cells (**e**–**g**) and granulocytes (**h**–**j**). Results are shown for: omission (**e**,**h**), inclusion (**f**,**i**), and bias errors (**g**,**j**). The shade around the curves corresponds to the standard deviation of the results.

omission, relative to no omission. For relatively large omissions ($r_\omega > 30\%$), MSD maintains a relatively low reduction in performance even for $r_\omega = 70\%$. However, this comes with the caveat that, for omissions above 30%, the training process of MSD occasionally collapses to an untrainable state. Both the training instability and the limited reduction in accuracy for large omission rates of MSD are a consequence of its design. The low number of parameters required by MSD might enable it to be less prone to overfitting on the wrongly labelled data, but also to become less stable when the label quality is substantially deteriorated.

Inclusion errors. In order to add inclusion perturbations, we merge the volumes of the HL60 nuclei cells and the granulocytes, while defining one data set to be the main one, with its labels being $\mathcal{L}$, while the other one becomes secondary, with its labels being Λ. In our experiments, we have $r_\phi \in \{10\%, 20\%, 30\%, 50\%, 70\%\}$. Figures 2.2(f, i) illustrate that the inclusion perturbation results in different behaviours for the models, depending on the data set it is applied to. In the case of HL60, SegNet

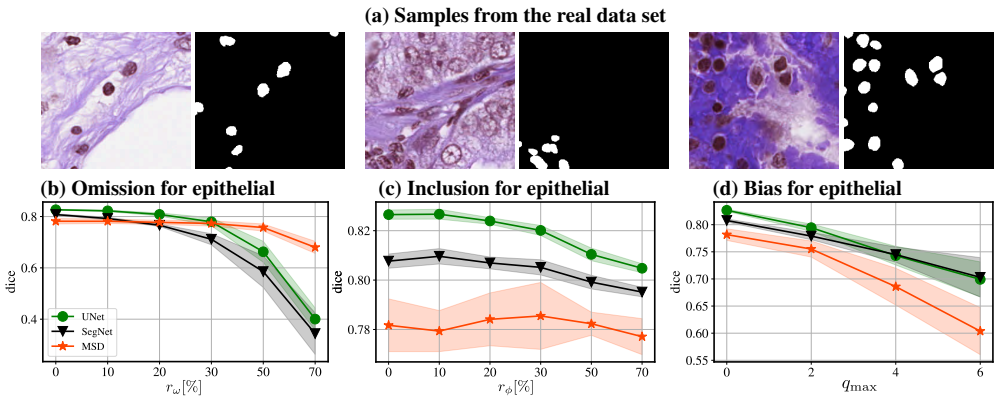


Figure 2.3: Example images and results for the manually-annotated data set. Figure (**a**) shows image/label pairs of stained-tissue images. Figures (**b**–**d**) show the Dice score of trained networks on the test set as a function of perturbation severity. Results are shown for: omission (**b**), inclusion (**c**), and bias errors (**d**). The shade around the curves corresponds to the standard deviation of the results.

presents a slow decreasing trend until $r_\phi = 30\%$, while UNet and MSD appear to be unaffected by the moderate inclusion. Also, since the granulocytes occupy a much smaller area in each slice than the HL60 cells, their addition into the latter's volume does not heavily impact the models even for 70% inclusion, resulting in a loss in performance of less than 1%. Moreover, Figure 2.2(i) shows that wrongly including large objects into the segmentation mask severely impacts all networks' capabilities, with an average Dice score drop of 23% for every 10% increase in r_ϕ.

Bias errors. For this type of perturbation, we performed our analysis by choosing $q_{\max}$ from $\{2, 4, 6\}$. Figures 2.2(g,j) show that introducing label bias through random morphological operations creates a descending trend for all network architectures. However, this trend presents different magnitudes depending on the data set. In the case of HL60 cells, we observe a decrease in Dice score of up to 5%, while for granulocytes the performance drops to 19%. This decline is a consequence of the much smaller footprint of the granulocytes' labels in relation to the background. Thus, mistakes in the outline of smaller cells are more costly than for their larger counterpart. In addition, we notice here that MSD and UNet perform similarly on the synthetic images, with SegNet lagging behind, by 6% and 10% on average for HL60 (Figure 2.2(g)) and granulocytes (Figure 2.2(j)), respectively.

2.3.3 Manually-Annotated Data

Following the observations drawn from the synthetic data we aim to extend them to a segmentation task of manually-annotated stained tissue images. We selected the data set belonging to MoNuSAC 2020 challenge [153], which contains H&E stained tissue images belonging to multiple organs. The data were gathered with the purpose of performing automatic cell segmentation, which can provide crucial information about the organ's health. This data set is comprised of 310 8-bit images of various sizes containing four types of cells: epithelial, lymphocytes, macrophages and neutrophils. Among these types, we selected the epithelial cells to be the target of our task, while considering the rest as background. The selection was motivated by the larger presence of the epithelial cells on a both per-image and per-data-set basis. Hence, we are left with 96 images for training and 37 for testing. Moreover, due to the varying size of the images, an extra preprocessing step was performed. The step involved separating each image into 256×256 patches using a sliding window technique while allowing for an overlap of 64 pixels between patches. A few samples of the selected patches are shown in Figure 2.3(a). Also, given the variability in size, quality and provenience of the data, we assume an individual annotator for every single image. Thus, we will apply our perturbation framework to each image separately.

The omission and bias-inducing processes are performed similarly to the synthetic data. For inclusion, we choose the main type of cells to be epithelial, while the lymphocytes form the secondary category. We chose lymphocytes since their pairing with epithelial cells is the most prevalent in the data set. We show the experimental results in Figures 2.3(b–d). In the case of omission, one notable difference from the synthetic data is the slight performance gap between MSD and the UNet/SegNet pair for $r_\omega < 30\%$. Nonetheless, this gap decreases the more error we allow, showing MSD to plateau at 77% Dice score until we remove 50% of the cell labels, while the other networks are severely affected (18% and 26% reduction for UNet and SegNet, respectively). When it comes to inclusion, the segmentation performance, similarly to the HL60 cells, appears to be rather unaffected by wrongly labelled additional cells until 30% inclusion. Moreover, the larger rates ($\geq$50%) inflict a more modest loss in the Dice score compared to the HL60 volumes due to the poorer fit the models have on real data. Since their learned parameters may not be a perfect fit for the data, the models can allow small perturbations of their input without suffering large losses. The bias on epithelial cells shows UNet and SegNet to develop an increasing gap from MSD, which reaches a 22% reduction in Dice score for $q_{\max} = 6$. Here, MSD appears to suffer

from the lack of complexity since this data set presents a more complex background with high variability between images, impeding, thus, a very good distinction of the correct cells. This tendency is further exacerbated by the perturbations applied to the cells' masks.

2.4 Conclusion

Understanding the consequences of labelling errors is of great importance for the field of biomedical image segmentation. Our study provided insights into meaningful issues that can be present in the annotation process for cell segmentation. We emulated three different labelling errors (omission, inclusion and bias) for two perfectly-labelled synthetic data sets and one manually-annotated data set and observed their impact on the results of three networks. We found that wrongly including large objects into the segmentation labels drastically decreases the quality of the predictions, while smaller objects are filtered out more easily when moderately included ($r_\phi \leq 30\%$). We also observed that, even in low amount, the presence of bias deteriorates the predictions for all cell types, especially for relatively smaller cells such as granulocytes and epithelial cells. Finally, we observed that moderate omissions ($r_\omega \leq 30\%$) present a negligible impact to both MSD an UNet, with the latter slightly outperforming the former on the manually-annotated data set. However, for larger omissions, MSD still retains a competitive Dice score. This robustness to omissions can be exploited in settings where the expert annotator would be required to label just a portion of the present cells, significantly reducing the annotation costs. Also, MSD could be used to pre-process training labels for more complex, but noise-sensitive, learning algorithms.

Chapter 3

Upgrading Low-Quality Annotations

3.1 Introduction

Deep-learning algorithms have been providing effective solutions for many tasks, contributing to the advancement of domains such as speech recognition [77] and computer vision [157]. One important computer vision task that is being tackled with such algorithms is image segmentation [106], a process that labels each pixel into categories, e.g., background and various cell types. However, deep-learning models require large quantities of annotated data for training. In addition, the provided annotations should also be of high quality. Specifically, the annotations should be accurate by providing information that reflects the reality within the input, and be complete, meaning that they provide all the information required for the given task, e.g., all pixels from an image have an associated label in a segmentation task.

For many biomedical imaging tasks, including cell imaging [61], the annotations are created manually by domain experts. Due to the limited availability of experts,

This chapter is based on:

> Ș. Vădineanu, D. M. Pelt, O. Dzyubachyk, and K.J. Batenburg. "Reducing Manual Annotation Costs for Cell Segmentation by Upgrading Low-Quality Annotations". *Journal of Imaging, 10(7), 172.* MDPI (2024).

the annotation process is often tedious [176], limiting the capacity for annotating the large quantities of data required by deep-learning algorithms. As a result, the general adoption of deep learning for such specialized domains may be considerably hindered. An annotation process with fewer quality constraints could significantly reduce the burden on expert annotators, enabling them to produce annotated images within a shorter time frame. For instance, when creating segmentation masks, the boundary of every cell in the image has to be carefully delineated. By providing coarser delineations, only annotating a subset of all cells, or relying on automatic but inaccurate segmentation tools based on classical image processing, a much faster annotation process can be achieved. However, training directly on low-quality annotations harms the performance of cell segmentation deep-learning algorithms [158]. Thus, it becomes apparent that a solution that leverages inaccurate annotations to expand costly training data sets can greatly benefit the adoption of deep learning for cell segmentation.

Learning from imperfect or missing labels due to annotation constraints is a long-standing issue associated with machine-learning tasks. In the case of cell segmentation, obtaining large amounts of labelled data requires time-consuming efforts by experts with specialized knowledge of the task. One field concerned with this problem is weakly-supervised learning, where the aim is to train deep-learning algorithms to produce complete segmentation masks by only providing the models with partial annotations. Such techniques usually vary in the amount of information that is present in the annotations, which can include bounding boxes [121], rough sketches of shape contours [21], geometrical shape descriptors in the form of centre points and lines [102], or partially-annotated segmentation areas [117]. Despite their promising results, these techniques are generally tailored towards a single type of inconsistency, which can limit their applicability.

Directly accounting for labelling errors, implicit consistency correction methods compensate for inaccuracies in the annotated input during the training process by, for instance, reducing the influence of gradients coming from segmentation areas of lower confidence [105], by using a teacher–student architecture [69] to change the label of less confident areas in the annotation mask [177], or by using adversarial training to only annotate high-confidence areas of unlabelled data [109]. On the other hand, explicit consistency correction solutions provide fine adjustments to the output of trained deep-learning models [34, 10, 29, 161]. Similarly to weakly supervised techniques, these methods lack a broad applicability and their utilization depends rigidly on custom architectures. When it comes to improving the provided labels, Yang et al. [172] developed a solution for iteratively adjusting the manual annotations of retinal

vessels by employing generative adversarial networks. Their framework, however, only produces small adjustments, relies on a relatively large amount of high-quality annotations, and may suffer from the challenges associated with generative models, e.g., mode collapse and convergence failure [32].

Also concerned with annotation scarcity, few-shot segmentation aims to segment new query images by leveraging information from relatively few support images with a limited amount of annotations. However, these approaches generally require additional training tasks with a large set of semantic classes [56, 55] whose annotations can be costly to obtain. The need for manual annotations can also be avoided by employing general foundation models such as the Segment Anything Model (SAM) [83], or cell-specific models such as Cellpose [137]. However, although the applicability of such models is not confined to a single image modality or cell type, they do not generalize well to images outside their vast training pool. For instance, the SAM is not accurate when the targets have weak boundaries [98], which can be the case with cell images [4], whereas Cellpose is sensitive to variations in the texture of the objects [137]. This may make these general solutions less suitable than techniques trained for a specific cell type.

In summary, although there are many methods designed for improving deep-learning segmentation with incorrect or incomplete labels, these solutions generally tackle specific types of inconsistencies, e.g., boundary uncertainty, require custom architectures or training schemes, or have considerable annotation requirements for additional training tasks. In this chapter, we present a method designed to be applied to a wide set of inconsistencies, with low data requirements and a flexible training scheme allowing for a straightforward integration in other pipelines. We propose a framework for effectively obtaining large amounts of high-quality training data with limited required human annotation time for the task of cell segmentation. Our approach is based on manually annotating a small training set of high quality, which we then enlarge with a much larger set with low-quality annotations (possibly produced with considerably less human effort). In order to leverage the low-quality annotations, we train a convolutional neural network to learn the mapping for upgrading a low-quality annotation to a high-quality one by presenting it with both high-quality annotations as well as low-quality versions of these annotations. We create multiple types of erroneous annotations by perturbing the high-quality annotations with a function that approximates potential errors resulting from a low-quality annotation process. Moreover, we show that this perturbation function does not need to exactly replicate the annotation errors present in the low-quality annotations in order for a good mapping to be trained. The

training process requires pairs of perturbed annotations with their corresponding images as input for the upgrade network with the unperturbed, high-quality annotations as targets. We apply the learned mapping to the large low-quality set to enhance its annotations. Finally, we combine the initial small set of well-annotated data together with the larger set with upgraded annotations and use them for training accurate deep-learning models for the task of cell segmentation. By separating the inconsistency correction step, i.e., the upgrading of annotations, from the segmentation step, we enable our framework to tackle a wide array of inconsistencies and we facilitate its integration into other segmentation pipelines.

3.2 Materials and Methods

3.2.1 Data Sets

Synthetic Data

We opted to use synthetic data to study most aspects of our method since their ground truth annotations do not suffer from the inconsistencies a human annotator may induce. Thus, we can be confident that such external factors do not influence the outcomes of our experiments. Also, to isolate the effect of a particular type of inconsistency in the low-quality set, we apply perturbation functions (see Section 3.2.2) throughout the experimentation with synthetic data. We employ three data sets [140], which consist of microscopy images of HL60 nuclei cells, granulocytes, and both cell types, respectively, produced by a virtual microscope [166] extracted from the Masaryk University Cell Image Collection (`https://cbia.fi.muni.cz/datasets`) (accessed on 15.06.2023). Each data set consists of 30 volumes of 129 slices, each containing 565×807 16-bit pixels. We filtered the volumes by eliminating the slices with empty labels, which resulted in differently-sized volumes, averaging 84 slices per volume. In addition, 25 volumes were used for training, while 5 volumes were kept only for testing. In Figure 3.1, we show sample slices and their corresponding high-quality annotations of the synthetic data sets. Since they are organized in volumes, we want to avoid selecting high-quality annotations of adjacent slices since such samples show little variation in their input and may be less informative when training the upgrade network than more distant slices. Consequently, we sample by subdividing a volume into a number of sections equal to the number of slices we want to select. We then select the middle slice of each section, thus ensuring an equidistant separation between slices. Additionally, when we sample from multiple volumes, we similarly partition

each volume, but we select every next slice from a section belonging to a different volume in a circular manner. For instance, when taking a total of 5 slices from 5 volumes, the first slice will be selected from the centre of section 1 from volume 1, the second from section 2 of volume 2 and so on.

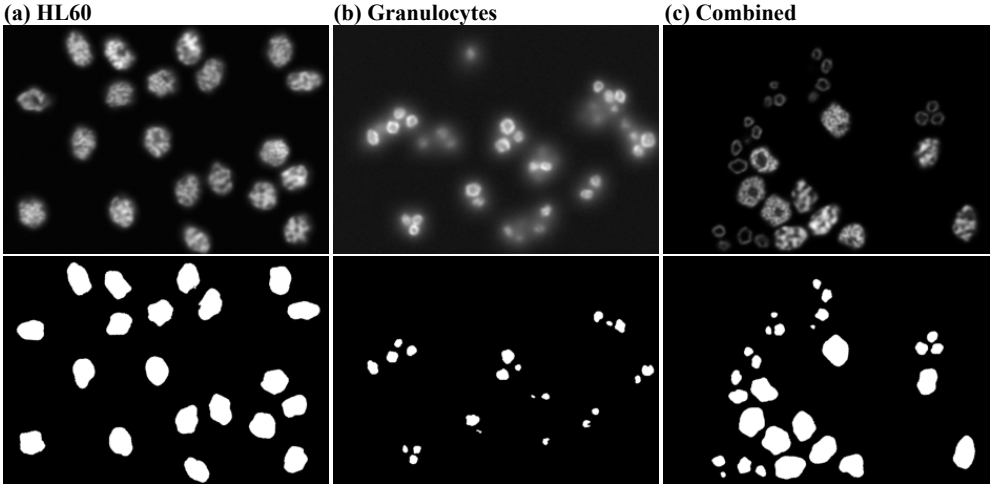


Figure 3.1: Sample slices and their corresponding high-quality annotations for the synthetic data sets we considered for analysis. The slices are produced with a virtual microscope [166].

Real Data

We also employ two manually-annotated data sets: the EPFL Hippocampus data set [95] and a large-scale data set for colonic nuclear segmentation called Lizard [59]. The EPFL data set is comprised of a training and a testing volume, each containing 165 slices of 768 $\times$ 1024 8-bit grayscale pixels. This set of images, obtained using focused ion beam scanning electron microscopy, is commonly used for benchmarking mitochondria segmentation algorithms, whose monitoring can provide, for instance, insights into the development of neurodegenerative diseases [84]. The Lizard data set contains histology RGB images of colon tissue of varying sizes with instance labels for each cell. Among the six cell types annotated in this data set, we selected the most prevalent category, i.e., epithelial cells, as our target and the remaining cells as background objects. This choice allows us to test our method on the largest number of samples, which ensures that we obtain the most statistically significant results. We split the images into 500 $\times$ 500 patches, with 100 pixels overlapping between patches and removed patches that did not contain epithelial cells. We partitioned the resulting set into 1209 training and 288 testing patches. In this case, we assume the

corresponding provided ground-truth annotations to be of high quality. For each data set, we select a small subset of samples for which we keep the high-quality annotations while perturbing the annotations of the remaining samples to generate the low-quality set. This perturbation step is performed only once per annotated image.

3.2.2 Method

In Figure 3.2, we illustrate an overview of our method. We consider a high-quality annotation process that produces labels in a slow and costly manner and a low-quality annotation process, yielding labels faster and cheaper. Within a given time frame, the processes would generate a small data set with high-quality labels and a larger lower-quality set. We apply perturbations to the well-annotated labels and we use the perturbed labels together with their corresponding images as input to train an upgrade model. We employ the upgrade model to enhance the labels of the larger data set, which we use in conjunction with the well-annotated samples to train the final segmentation model.

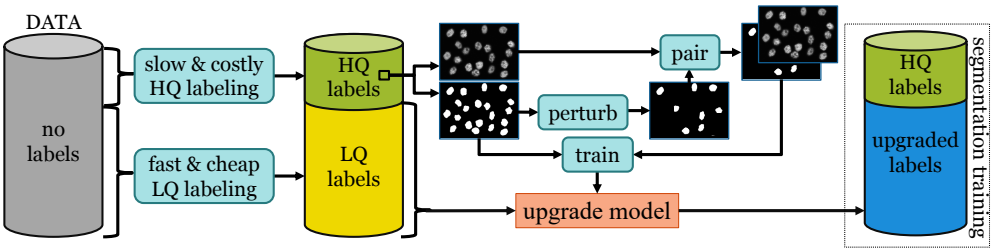


Figure 3.2: Workflow. We train the upgrade model on a small set with high-quality labels. We apply the trained model to upgrade the low-quality labels of a larger set. We enlarge the initial high-quality set with the upgraded labels and we use the combined set for segmentation training.

Background

We apply our framework to the segmentation task of 2-dimensional vector-valued (e.g., RGB, grayscale) images. In this chapter, we define an image as a matrix of pixels $x \in \mathbb{R}^{N \times M \times C}$, where N, M, and C represent the number of rows, columns, and channels, respectively. The goal of segmentation is to create a mapping from a given input x to the target $y \in \mathbb{Z}^{N \times M}$ in order to provide a separation between the different entities within that image. Essentially, a label is attributed to each pixel according to the entity that it belongs to. When using deep learning for image segmentation, this mapping is

approximated using convolutional neural networks (CNNs), $f_\delta : \mathbb{R}^{N\times M\times C} \to \mathbb{R}^{N\times M}$, which require a set of image-target pairs $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{N_t}, y_{N_t})\}$, to train their parameters, δ. The process of training neural networks usually involves successive predictions based on the input x and adjusting the parameters such that the loss between the predictions and the labels is minimized. In order to achieve the desired results, the network requires well-annotated training samples. We describe the annotation process that produced high-quality labels as the output of the high-quality annotator,

$$A^{HQ} : \mathbb{R}^{N\times M\times C} \to \mathcal{A}^{HQ}, \tag{3.1}$$

that receives an input image x and produces a label that belongs to the set of high-quality annotations, $\mathcal{A}^{HQ}$, i.e., it is both complete and correct. Such an annotation can be the result of a consensus between multiple experts or can require a slow and careful delineation of the shape of each element in x by a single expert. Additionally, we define the set of well-annotated images, $X^{HQ} = \{(x, A^{HQ}(x))\}$, needed to train the network parameters,

$$\widehat{\delta} = \underset{\delta}{\operatorname{argmin}} \sum_{(x,y)\in X^{HQ}} L(f_\delta(x), y), \tag{3.2}$$

where L is a loss function. Due to their large parameter count, these models are generally prone to overfitting and therefore require large quantities of well-annotated samples.

Perturbation-Removal Framework

Since producing a sufficient number of high-quality annotations may prove unfeasible for cell segmentation, the required annotations may be supplied via a less rigorous annotation process. A low-quality annotation process would, for instance, result from an individual expert who quickly produces the annotation, without spending additional time on finer shape details or on removing ambiguities. Also, for setups that require consensus, the label can come from a single expert, a person in training, or a non-expert, thus reducing the annotation costs. Alternatively, the low-quality annotations can even be the product of traditional segmentation techniques (e.g., thresholding, graph cut [14], Otsu [111]) or machine-learning-based algorithms, removing the need for a human annotator in this stage of the process. For instance, one easily-applicable strategy to produce low-quality annotations is to simply train a segmentation network

on the few available high-quality samples and then use its predictions on the remaining unannotated samples as low-quality annotations. We define the low-quality annotator

$$A^{LQ} : \mathbb{R}^{N \times M \times C} \rightarrow \mathcal{A}^{LQ}, \tag{3.3}$$

as a function that produces labels that are either incorrect or incomplete or both, thus, being included in the set of low-quality annotations, $\mathcal{A}^{LQ}$.

Training solely with low-quality annotations generally leads to inaccurate results [158]. Thus, we propose a solution to enhance the quality of a larger set of low-quality annotations, which we utilize to enlarge an initially small set of high-quality annotations. Our framework requires a small number of well-annotated images, X^{HQ}, together with a substantially larger set of images and their low-quality annotations, $X^{LQ} = \{(x, A^{LQ}(x))\}$, with $|X^{HQ}| < |X^{LQ}|$. We aim to enhance $A^{LQ}(x)$ to $\mathcal{A}^{HQ}$ by finding the upgrade function

$$U : (\mathbb{R}^{N \times M \times C}, \mathcal{A}^{LQ}) \rightarrow \mathcal{A}^{HQ}, \tag{3.4}$$

which translates an annotation of the input image created by the low-quality annotator to an annotation belonging to the space of high-quality annotations. In order to create both high- and low-quality versions of annotations, we utilize a perturbation function that aims to approximate the unknown mapping from a high-quality annotation to a low-quality one. We handcraft function

$$P : \mathcal{A}^{HQ} \rightarrow \mathcal{A}^{LQ}, \tag{3.5}$$

which applies perturbations to a high-quality annotation to create an annotated image that approximates a faster, but lower-quality, annotation process. The choice for such a function can vary by task and data set, with implementations that can include heuristics or even learning the perturbations from the data. In our work, we assume that we can approximate the perturbation function, P, by implementing a custom stochastic version of it. Additionally, we assume that the function U that maps the low-quality label to a high-quality one is a learnable function. We employ the high-quality set to generate many $(x, P(A^{HQ}(x)))$ pairs. Given the stochastic nature of our chosen perturbation function, we can generate multiple perturbed versions of the same high-quality annotation; thus, we only require a small number of $(x, A^{HQ}(x))$ pairs. We utilize the generated pairs to train an upgrade network, u_θ, parametrized

by θ, which approximates U by finding

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{(x, A^{HQ}(x)) \in X^{HQ}} L(u_\theta(x, P(A^{HQ}(x))), A^{HQ}(x)), \tag{3.6}$$

where L is a loss function. After training u_θ, we apply it to our lower-quality set. In this way, we enhance the low-quality annotations, which results in the pairs $(x, u_\theta(x, A^{LQ}(x)))$ of input images and upgraded annotations. Finally, we use both the enhanced $(x, u_\theta(x, A^{LQ}(x)))$ pairs and the initial high-quality $(x, A^{HQ}(x))$ pairs as training samples for our final segmentation task. Therefore, our segmentation CNN f_δ will be obtained as

$$\begin{aligned} \widehat{\delta} = \underset{\delta}{\operatorname{argmin}}(& \sum_{(x,y) \in X^{HQ}} L(f_\delta(x), y) + \\ & \sum_{(x,y) \in X^{LQ}} L(f_\delta(x), u_{\widehat{\theta}}(x, y))). \end{aligned} \tag{3.7}$$

Algorithm 1 shows the pseudocode of a segmentation pipeline that makes use of our upgrade network. The requirements of our framework are (1) a small set with high-quality annotations, (2) a larger set with low-quality annotations, and (3) a perturbation function. The objective of this pipeline is to obtain the parameters δ of a well-trained segmentation network. We initially train the upgrade network u_θ only on the high-quality data X^{HQ}, whose labels we perturb with the previously selected perturbation function, P. We aim here to obtain predictions from input images and perturbed labels that match the high-quality annotations as closely as possible. After estimating the parameters of u_θ, we apply it to X^{LQ}, whose images and resulting upgraded annotations we employ, in conjunction with X^{HQ}, to estimate the parameters δ of a segmentation network.

Producing Low-Quality Annotations

We designed our method for the task of binary cell segmentation, where the object of interest is a single type of cell. In order to apply our perturbation function, we require the instance label of every cell in the image. Therefore, considering E cells in image x, we define $\mathcal{L} \subset \mathbb{Z}$ as the set of all cell instance labels, with $|\mathcal{L}| = E$. Our label then becomes

Algorithm 1 Upgrade Framework

Require: X^{HQ}, X^{LQ}, P **return** δ
(1) Train the upgrade network u_θ:
for $(x, y) \in X^{HQ}$ **do**
Perturb y: $P(y)$
Predict upgraded label: $u_\theta(x, P(y))$
Compute loss: $L(u_\theta(x, P(y)), y)$
end for
Estimate $\widehat{\theta}$ according to Equation (3.6)

(2) Upgrade low-quality set and expand segmentation training data:
for $(x, y) \in X^{LQ}$ **do**
Upgrade low-quality label: $u_\theta(x, y)$
end for
Estimate $\widehat{\delta}$ according to Equation (3.7)

$$y_{nm} = \begin{cases} i, & \text{if } x_{nm} \text{ belongs to cell } i \in \mathcal{L}, \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$
$$1 \leq n \leq N,\ 1 \leq m \leq M.$$

We apply three types of perturbations (omission, inclusion, and bias), introduced in [158], which are designed to reflect the incompleteness and inaccuracy of the cell segmentation masks resulting from an annotation process with fewer resources. For instance, a much shorter annotation time can be spent by using segmentation masks that only contain a proportion of the total cells present in the image. Moreover, allowing for inconsistencies in cell recognition in the form of inclusions can also reduce the time an annotator spends choosing which cells to include in the segmentation mask. Finally, by eliminating the need to provide correct cell border delineations, we can expect a boost in the annotation speed.

Omission Perturbation. We randomly select a subset of $S \leq E$ of cell instance labels $\mathcal{L}^{\mathcal{S}} \subseteq \mathcal{L}$, whose size is chosen such that it satisfies the *omission rate* $r_\omega = \frac{S}{E}$. Our perturbation function, therefore, becomes

$$P(y)_{nm} = \begin{cases} 0, & \text{if } x_{nm} \text{ belongs to cell } i \in \mathcal{L}^S, \\ y_{nm}, & \text{otherwise} \end{cases} \quad 1 \leq n \leq N,\ 1 \leq m \leq M. \tag{3.9}$$

Inclusion Perturbation. Given an image x and $\Lambda \subset \mathbb{Z}$, a set of instance labels of other objects belonging to x ($\mathcal{L} \cap \Lambda = \emptyset$), we perform inclusion by randomly selecting a subset $\Lambda^S \subseteq \Lambda$ of the objects, whose size $S \leq F$ satisfies the *inclusion rate* $r_\phi = \frac{S}{F}$. Hence, we apply the perturbation as

$$P(y)_{nm} = \begin{cases} j, & \text{if } x_{nm} \text{ belongs to shape } j \in \Lambda^S, \\ y_{nm}, & \text{otherwise} \end{cases} \quad 1 \leq n \leq N,\ 1 \leq m \leq M. \tag{3.10}$$

Bias Perturbation. We model the inconsistency in border delineation by performing morphological operations [130] on the cell labels. We employ dilation operations, D, to enlarge the cell area and erosion operations, E, to shrink the cell area. The operation is randomly chosen and the impact of the operation is controlled by factor q that controls the number of iterations, with a 3×3 all-ones matrix as the fixed structural element, for which we perform the chosen operation. This bias severity constant, randomly picked between 1 and q_{max}, indicates the largest allowed number of iterations. As a result, the perturbation is formed either as

$$P(y) = E^q(y) \text{ or } P(y) = D^q(y). \tag{3.11}$$

where E^q and D^q denote q iterations of erosion and dilation, respectively.

Given the relatively ill-defined distinction between low-quality and high-quality annotations, we will further consider as low-quality annotations only the ones affected by large degrees of perturbations, i.e., 70% omission, 70% inclusion, a bias of 6, or a combination of perturbations. Thus, we only consider as low-quality the annotations that significantly diverge from the gold standard. In Figure 3.3, we illustrate an example of an annotation where all three perturbation types are present and highlighted. Alternatively, we investigate the case where the low-quality annotator A^{LQ} would not imply any human effort. This can happen when A^{LQ} are produced by a segmentation

network trained on the small number of samples in the high-quality set $\mathcal{A}^{HQ}$. In this case, the generation of low-quality annotations is disentangled from the perturbations that we apply when training the upgrade network.

3.2.3 Experimental Setup

We designed our experimental setup around a PyTorch [114] implementation of UNet [125]. UNet features an encoder–decoder architecture with skip connections between the encoding layers and the decoding layers of the same spatial resolution. We employed 4 convolutional blocks in the encoder and 4 in the decoder, with a block containing 2 convolutional and 2 batch normalisation layers. We treat both the segmentation and upgrade tasks as binary pixel-wise classification tasks. Thus, the output of the network in both cases is a two-channel image with the first channel's pixels being 0 if they belong to the foreground and 1 if they belong to the background, with the opposite holding true for the second channel. All activations between layers are ReLU functions, with the exception of the last layer, where the output is processed by a soft-max function. We train the network until there is no improvement in the validation score for 10 consecutive epochs, at which point we only keep the model with the highest score. Our loss function is the Dice loss, and we update the network's parameters according to ADAM optimization algorithm [82], with a learning rate of 10^{-5} and a batch size of 4. We partition our data into training and testing with an additional 80/20 split of the training data into training and validation. Finally, we present our results by reporting the Sørensen–Dice coefficient computed over the entire test set and averaged over 5 runs. We validated our comparisons by using the Wilcoxon non-parametric test [124].

3.3 Results

We performed a series of experiments to analyze various aspects of our proposed framework. In Section 3.3.1, we use the synthetic data sets with objective ground truth to measure the quality gain of upgraded annotations under various sets of assumptions. On the same data sets, we also evaluate the benefits of expanding the segmentation training data with upgraded annotations in terms of segmentation performance and annotation cost (Section 3.3.2). Furthermore, in Section 3.3.3, we validate our previous observations on real manually-annotated data. Lastly, we show, in Section 3.3.4, a case study of an application where our solution can be integrated to improve the prediction

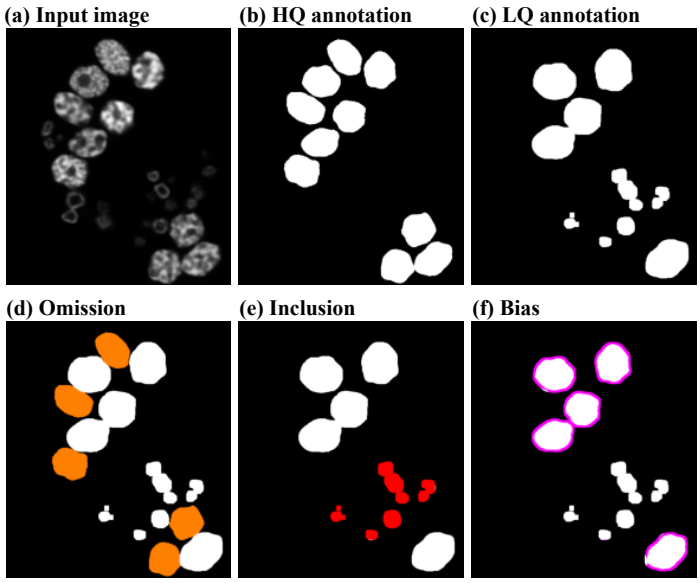


Figure 3.3: An example of the perturbations applied to the high-quality annotations. Figure (**a**) presents an input image from the combined data set, where the HL60 cells are the target cells and the granulocytes are the included cells. The high-quality annotation corresponding to the input is shown in (**b**). The low-quality version of the annotation shown in (**c**) is affected by 50% omission, 50% inclusion, and a bias of 6. The omission perturbation is represented by the orange omitted cells in Figure (**d**), inclusion by the red shapes in (**e**), and bias by the magenta contours in (**f**).

quality of a segmentation network trained with insufficient samples.

3.3.1 Analysis of the Upgrade Network

To assess the optimal training set size for the upgrade network u_θ, we created various training sets by varying both the total number of annotated slices and the number of volumes from which the annotated slices were selected. The models were trained to upgrade annotations affected by 70% omission, 70% inclusion, and a bias of 6, respectively. The results presented in Figure 3.4 show that the upgrade network requires just 5 well-annotated slices to improve the quality of the annotations, regardless of the applied perturbation. We also notice that the resulting quality of the upgraded annotations plateaus quickly to Dice values > 0.9. We report the optimal number of training slices for different perturbations together with the corresponding Dice score of the upgraded annotations in Table 3.1.

So far, we assumed that we can perfectly model the errors affecting the low-quality

3.3. Results

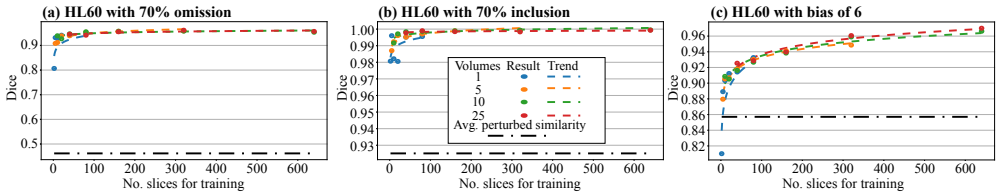


Figure 3.4: The Dice similarity with the ground truth test set of the upgraded annotations as a function of the total number of slices used for training and the number of volumes from which the slices were selected for the HL60 cells (represented using a different colour). The coloured dots represent the experimental results, while the coloured dashed lines are showing the general trend of the results. The straight dash-dotted line represents the average Dice similarity with the ground truth test set of the perturbed annotations before the upgrade.

Table 3.1: The Dice similarity with the ground truth test set of the annotations affected by perturbation and the upgraded annotations, as well as of the predictions produced by segmentation networks trained only on the high-quality data, on the high-quality data together with the data with upgraded annotations, and results of using thresholding as baseline. In each row, the largest value is highlighted in bold. The training setup indicates the data set on which the upgrade network was trained, as well as the total number of slices used for training and the number of volumes from which the slices were selected. The cell types marked with an asterisk come from the combined synthetic data set.

	Training setup for upgrade network			Quality of training annotations		Quality of segmentation network				
						Training data				
Perturbation	Data	Vols.	Slices	LQ	Upg.	HQ	HQ + upg.	HQ + LQ	LQ only	Thrs.
70% omission	HL60	10	10	0.462	**0.939**	0.823	**0.929**	0.311	0.311	0.887
	gran.	10	80	0.495	**0.92**	0.892	**0.894**	0.41	0.414	0.732
70% inclusion	HL60*	10	10	0.925	**0.992**	0.913	**0.962**	0.891	0.89	0.892
	gran.*	10	10	0.381	**0.98**	0.856	**0.898**	0.364	0.353	0.214
bias 6	HL60	10	10	0.857	**0.909**	0.823	0.923	0.931	**0.933**	0.887
	gran.	10	40	0.675	**0.865**	0.868	**0.877**	0.827	0.81	0.732
30% om. 30% inc. bias 4	HL60*	10	10	0.71	**0.929**	0.913	**0.934**	0.739	0.745	0.892
	gran.*	10	10	0.54	**0.86**	**0.856**	0.854	0.505	0.5	0.214

annotations with our perturbation functions. However, in practice, it might be difficult to exactly match the type and severity of the perturbations present in the data. To account for that, we relax this assumption by allowing a mismatch between the error generated by the perturbation functions and the errors in X^{LQ}. In Table 3.2, we report the effect of such mismatch on the performance of the upgrade network when the annotations of X^{LQ} contain 30% omission, 30% inclusion, and a bias severity of 4, respectively. We observe that, even when not reaching the highest Dice scores, the upgraded annotations show high Dice scores when u_θ is trained on the highest perturbation level. This implies that varying the presence of a large proportion of the cell masks can be more beneficial for training u_θ than aiming to exactly match the

Table 3.2: The Dice similarity with the ground truth test set of the upgraded network trained on various degrees of perturbations. The perturbations present in the low-quality set are 30% omission, 30% inclusion, and a bias severity of 4, respectively. For each perturbation type, the highest score is highlighted in bold.

	Training perturbation for upgrade network								
	Omission			Inclusion			Bias		
	20%	30%	50%	20%	30%	50%	2	4	6
HL60	0.955	**0.972**	0.952	0.973	0.972	**0.986**	0.915	0.918	**0.926**
gran.	0.838	0.86	**0.93**	**0.984**	0.98	0.981	0.821	0.837	**0.884**

amount of error present in the X^{LQ}.

In addition to the perturbation function, another essential requirement of our solution is the presence of a high-quality set of annotations for training u_θ. Since we use synthetic data, the quality of this set is ideal, which, however, is not expected from manual annotations for many reasons, including inter-observer variability [13] or limited available resources. We model these inaccuracies by introducing moderate amounts of perturbations into the high-quality set. Figure 3.5 illustrates that the upgrade networks trained on the larger HL60 cells are robust to imperfect HQ annotations, whereas the ones trained on granulocytes are more sensitive due to the comparatively smaller footprint of the cells. Thus, the same amount of perturbation affects the quality of the granulocytes annotations more drastically than that of HL60 cells. Despite allowing for a moderate amount of omission and inclusion perturbations, the networks trained on granulocytes show a sharp drop in performance for bias since this type of perturbation introduces the greatest variation in shape relative to cell size among the two data sets.

We compare our solutions with works tackling the issue of training biomedical image segmentation models with imperfect or incomplete annotations. We selected

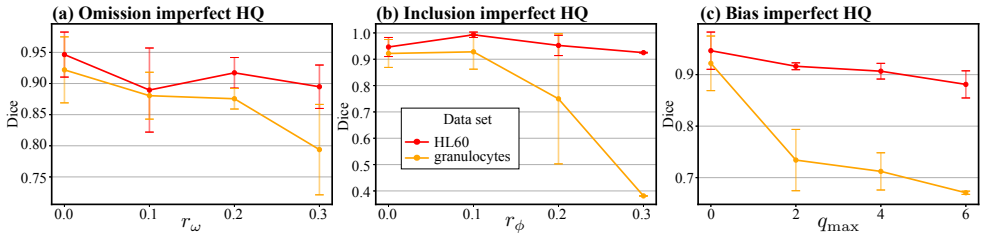


Figure 3.5: The Dice similarity with the ground truth test set of the upgraded annotations as a function of the perturbation level present in the high-quality training set of the upgrade network. The vertical bars correspond to the standard deviation of the results.

Table 3.3: The Dice similarity with the ground truth test set of the predictions produced by three different segmentation networks designed for training with incomplete/noisy annotations. The models were trained on 10 volumes of the combined set of HL60 and granulocytes with the HL60 cells as the target cells. The upgrade network of our method was trained using 20 well-annotated slices, which were also included in the training set of the other two methods. For each perturbation, the highest score is highlighted in bold.

	Perturbation		
Method	70% omission	70% inclusion	bias 6
Partial Labelling [117]	0.906	0.859	0.803
Confident Learning [177]	0.381	0.888	**0.947**
Ours	**0.923**	**0.962**	0.916

techniques that employ full-size segmentation masks for training and that apply corrections to these masks to either fill incomplete areas or remove incorrect ones. Also, although we compare the selected methods for all our perturbation types, it is important to note that Partial Labeling [117] was designed for setups closer to omission than the other perturbations, whereas Confident Learning [177] tackles uncertain areas at the border of the masked areas resembling more our bias perturbation. In Table 3.3, we observe that our method generates comparable results with Partial Labeling for omission and Confident Learning for bias perturbation. However, among all three perturbation types, our framework performs consistently better than the other solutions, showing wider applicability to different types of inconsistency.

3.3.2 Segmentation Improvements

In Section 3.3.1, we investigated the capability of the upgrade network to improve the quality of annotations affected by errors. In this section, we are analysing whether adding the upgraded annotations to the training set results in improved segmentation performance and reduced overall annotation costs. In Table 3.1, we report different scenarios under which X^{HQ} and X^{LQ} can be used to train networks for segmentation. Given an initial data set with low-quality annotations, we can use it directly as a training set for segmentation (LQ only column in Table 3.1). We can also spend additional resources on improving the quality of a small number of annotations and utilize them in conjunction with the low-quality set (column HQ + LQ) or we can employ the high-quality set alone for training (column HQ). Finally, we can use our framework for upgrading the low-quality annotations and, together with X^{HQ}, forming a larger training set of improved quality for the segmentation network (column HQ + upgraded). In order to ensure that the synthetic data cannot be easily segmented based

on the pixel intensity levels, we use as baseline a simple thresholding solution in which the input images are segmented by selecting a threshold via grid search with a step of 1% of the maximum pixel intensity. For each data set, we select a single threshold that yields the highest Dice score on the training set. The low baseline results in Table 3.1 reflect the complexity of the simulated data sets. Our results show that, for most cases where u_θ improved the quality of annotations, the addition of samples with upgraded annotations translated into a higher segmentation performance of the final segmentation network on the test data.

From Table 3.1, we observed that adding the upgraded annotations to the training set results in better segmentation. However, this performance gain resulted from upgrading a large number of low-quality annotations, which may also prove difficult to produce in practice. To account for this, we perform an experiment analysing the trade-off between annotation cost and performance. For a fixed number of slices, we select 10% of them to have high-quality annotations, while the rest have low-quality annotations. We apply our framework to this set of slices and compare against segmentation networks trained with low-quality annotations, i.e., 0% high-quality slices, and against segmentation networks trained on high-quality slices only, i.e., 100% high-quality slices. We define the annotation cost as the *equivalent number of low-quality annotations* that would be produced with the same effort as a given annotation. For instance, for a low-quality annotation, the equivalent number of low-quality annotations is 1, while for a high-quality annotation, this number will differ depending on the particularities of the task, such as the data sets or the experience of the annotators, as is the case with works comparing annotation costs in the literature [133, 36]. For illustration purposes, we consider the equivalent number of low-quality annotations for a high-quality annotation to be 5. We observe in Figure 3.6 that, except for bias perturbation, the segmentation networks trained with our framework are the most cost-effective option for reaching the highest Dice scores. When it comes to bias, the variation in cell size induced in the training set with low-quality annotations forces the network to learn an "average" cell size that matches more closely the ground truth in the test set. However, in cases where the bias is more systematic, we expect a considerable drop in performance for the networks trained only with low-quality labels.

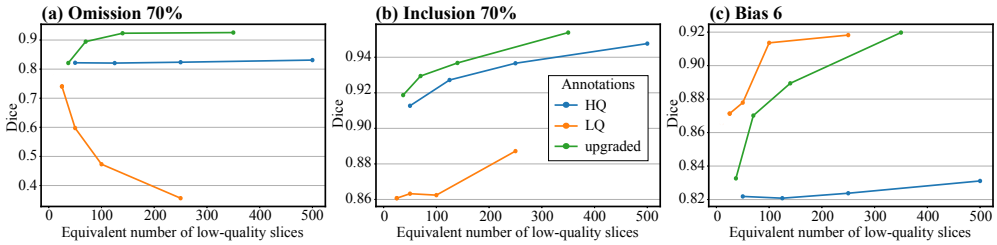


Figure 3.6: The Dice similarity with the ground truth test set of the segmentation networks as a function of annotation cost. The results in Figure (**a**) correspond to the HL60 data set where the low-quality annotations suffered from 70% omission, the results in Figure (**b**) correspond to the combined data set with HL60 cells as targets and 70% inclusion in the low-quality annotations, and the results in Figure (**c**) correspond to the HL60 data set with a bias of 6 in the low-quality annotations.

3.3.3 Enhancing Manual Annotations

In Section 3.3.1, we showed that the upgrade network is able to improve low-quality annotations of synthetic images under various circumstances. Here, we expand our analysis by validating our observations on real cell images. We integrate the two described real data sets in a scenario emulating the process experts may undertake to enhance the quality of their annotations. Our goal is to assess whether the quality gains reported in Table 3.1 can be similarly reproduced on real manually-annotated data. We consider a setup where the constraints on the annotation process are accurately captured by the perturbation functions used during the training of the upgrade network. With omission, we model an expert that deliberately ignores most cells in an image, focusing only on 30% of them. Inclusion allows for the presence of other structures that, for instance, can result from using networks trained on other cell data sets, or from foundation models. Bias would allow the annotator to either focus on the "core" of the cell, as shown in Figures 3.7m,o, or on the wider cell area without rigorously delineating the boundaries. Figure 3.7 shows the results of the upgrade network trained on 24% of the training samples of EPFL, and on 20% of Lizard's, respectively. We notice, both qualitatively and quantitatively, that our solution can successfully upgrade annotations affected by high perturbation levels, requiring a relatively low number of high-quality annotations for real, more complex data sets. Also, the large quality increase for omission and bias highlights the potential of our framework to expand the size of cell data sets with relatively low effort for producing the low-quality annotations.

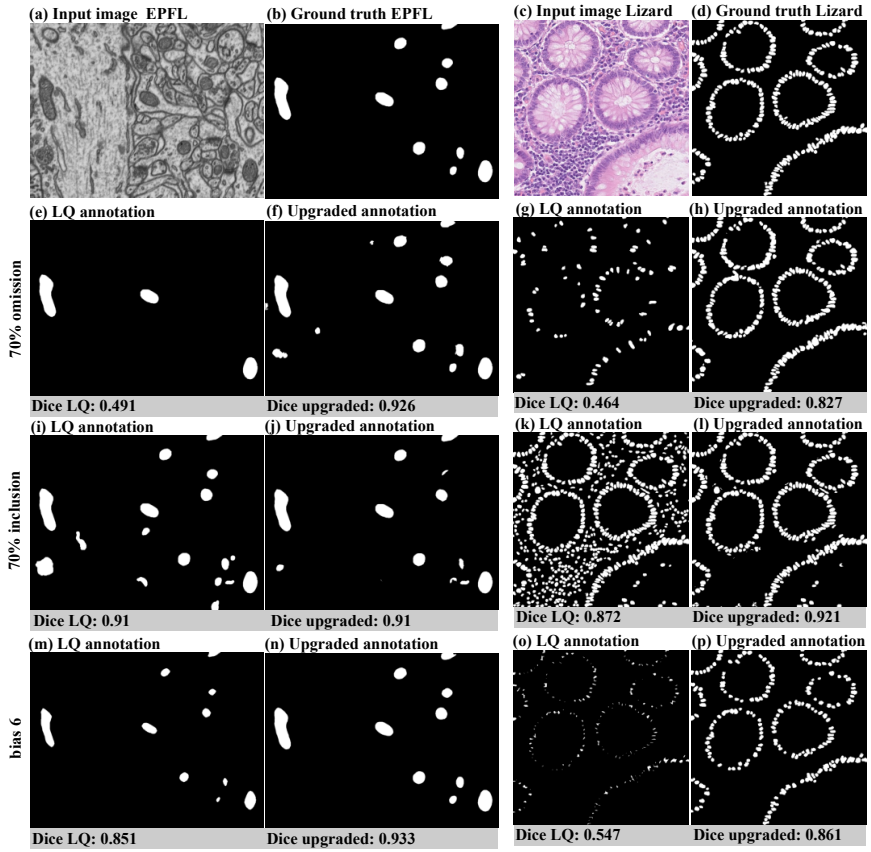


Figure 3.7: An example of perturbations applied to the real data sets paired with upgraded annotations. Figures (**a**–**d**) show the input image paired with its corresponding ground truth for EPFL and Lizard. Figures (**e**–**h**), (**i**–**l**), and (**m**–**p**) present the perturbed-upgraded annotation pairs for 70% omission, 70% inclusion, and a bias of 6, respectively. The results below the images represent the Dice similarity between the ground truth, the low-quality annotations, and the upgraded annotations, respectively. Both metrics were computed on the entire test set.

3.3.4 Case Study: Upgrading Low-Quality Predictions

We showcase here an example where the upgrade network can be applied in a scenario requiring no manual annotation cost for producing the low-quality annotations. In this case, X^{HQ} can be employed to train a segmentation network whose predictions can then be further used as the cheap annotations of X^{LQ}. We consider the predictions of a segmentation network trained with 10 well-annotated samples of Lizard data set in a setup similar to [161]. We use the same X^{HQ} for training our upgrade network.

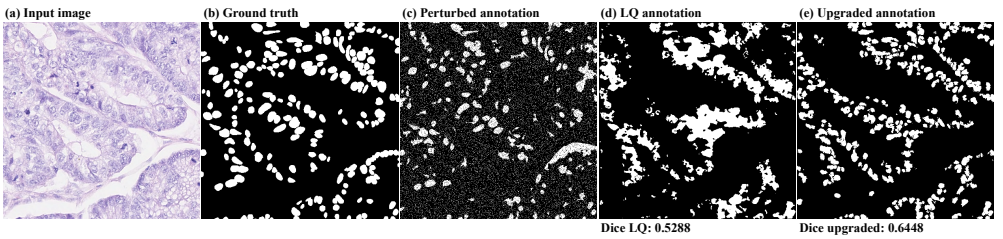


Figure 3.8: An example of an application where our framework can upgrade the predictions of a segmentation network trained on insufficient data. Figures (**a**,**b**) show the Lizard input image together with the corresponding ground truth. Figure (**c**) is an example of a perturbed annotation used during the training of the upgrade network. Figure (**d**) shows a low-quality annotation produced by the segmentation network with its upgraded version presented in Figure (**e**). The results below the images represent the Dice similarity between the ground truth, the low-quality annotations, and the upgraded annotations, respectively. Both metrics were computed on the entire test set.

We opted for a set of perturbations that would guide u_θ to compensate for prediction inaccuracies that we visually assessed. At each training iteration, we perform a 50% omission, followed by the inclusion of 10% of the segments extracted by Felzenszwalb's algorithm [54] to emulate missing or mispredicted structures. We add salt and pepper noise with a 10% probability to mimic the observed gaps in the segmented area as well as the small clusters of false positive pixels that can be noticed in Figure 3.8d. Finally, the resulting label is subjected to bias perturbation with a bias of 6 to guide the upgrade network towards better delineation of cell boundaries. The results, shown in Figure 3.8, demonstrate the potential of our method to refine the predictions of an undertrained segmentation network. We achieve a 22% improvement in the quality of the predictions without requiring additional supervision. Moreover, by visually inspecting the results, we notice that u_θ achieves good separation between the individual shapes, a property not captured by the Dice score metric. These delineated shapes can then be used, for instance, to facilitate a further instance segmentation step.

3.4 Discussion

Our results reported in Table 3.1 indicate that, with as few as 10 well-annotated images, we can improve low-quality annotations to a level comparable with the gold standard. In addition, as can be seen in Figure 3.4, the performance of the upgrade network relative to the size of the high-quality data set follows a logarithmic trend. Therefore, continuously increasing the size of X^{HQ} will not generate meaningful im-

provements. By knowing the logarithmic trend of the performance of u_θ, the end user of our framework would benefit from being able to decide more easily when enough high-quality data has been gathered and annotated, since, once u_θ performs well for a certain size of X^{HQ}, little improvement can be expected when the size of the training is increased. Furthermore, we showed that the upgrade network produced positive results for all considered cell data sets. The only requirements are a small high-quality set of annotations, a separate larger set of low-quality annotations, and a perturbation function that can map a high-quality annotation to multiple lower-quality versions of it, resembling the quality within the low-quality set. Since our requirements are independent of the data set, we expect our method to also work on other image modalities where our assumptions are met. This also applies to data collected in the three-dimensional regime, such as tomography. In this case, our framework can be applied on each individual slice separately.

We observed that using both the upgraded annotations of the low-quality set together with the small well-annotated set generally results in higher segmentation scores. Moreover, we noticed that the highest Dice scores are obtained when the upgrade model is both trained with and applied to annotations perturbed with 70% omission, 70% inclusion, or a bias of 6. We also saw in Figure 3.6 that our framework can be a cost-effective solution to increase the performance of segmentation networks when the annotation time is a constraint. Moreover, by comparing with other works targeting the enhancement of imperfect annotations, we showed that our upgrade network can handle a wider variety of perturbations than existing techniques. Thus, our solution is well-suited for being embedded into an annotation process with limited resources, rather than for fine-tuning, where there is a wide gap between the cost of producing a low-quality annotation and the cost of producing a high-quality one. For instance, for automatically-produced annotations by a non-learning algorithm, the only costly requirement would be to manually enhance a small proportion of them, on which the upgrade network can be trained. Moreover, as shown in Figure 3.8, our solution is flexible enough to be used for upgrading predictions of a network trained with insufficient data. These upgraded predictions can then be used to enlarge the existing data set or be further adjusted by experts, reducing the overall annotation time.

We also noticed the benefit of training for high perturbation levels, i.e., 70% omission, 70% inclusion, and a bias of 6, when we tested the robustness of our solution with respect to discrepancies between the perturbation levels used to train the upgrade network and the perturbation levels in the low-quality set. In Table 3.2, we saw that, generally, when we train for the highest perturbation level we reach comparable, or higher, performance than when training on the same perturbation applied to generate the low-quality set. Since, in practice, the annotation inaccuracies can have a systematic, i.e., annotator-specific, component and a random component, it may prove impossible to exactly model these inaccuracies through perturbations. Thus, the robustness to discrepancies in perturbation levels shown by our framework can indicate its potential applicability in practical scenarios. We additionally showed that our framework is robust to reductions in the quality of X^{HQ}. Figure 3.5 shows that we can expect a relatively small drop in performance when we moderately reduce the quality of the well-annotated set. This observation may imply that the annotation process of X^{HQ} can become less costly, e.g., requiring fewer experts per high-quality annotation, while still being able to produce annotations to train a well-performing upgrade network. However, the less information is present in an annotation, e.g., small cell areas, the more sensitive the framework becomes to inconsistencies.

Given that we focused solely on cell segmentation, we are unable to conclude with certainty whether or not our framework is applicable to other image segmentation applications where the goal would diverge from the cell segmentation setup, for instance by requiring the segmentation of a single contiguous target object. However, considering that our framework does not demand a specific type of annotation, as long as sufficient realistic low-quality versions of the high-quality annotations can be created with enough variety between them, we expect the upgrade network to still be applicable. Despite this, further experimentation is required to ensure that our requirements are met by other segmentation applications. Another limitation presented by our work is the lack of integration of the third dimension for volumetric data sets. This can be tackled in the future by, for instance, employing an architecture with 3D convolutions as the upgrade network. Finally, throughout our experimentation, we upgraded only annotations suffering from high levels of inconsistencies, while ignoring the fine-tuning of less severe cases. We expect our upgrade network to not perform similarly well on such cases, given that the small errors would not allow for much variation in the generation of the low-quality versions of the annotations. This would then impede the network from learning a generalizable mapping from a low-quality annotation to a high-quality one.

3.5 Conclusions

We presented our framework for enlarging training data sets with limited human annotation costs by only requiring a small set of data with high-quality annotations and a larger set with low-quality annotations that would require little or no human annotation effort. We utilize a small high-quality data set whose annotation quality is reduced for providing it as input to an upgrade network that learns the mapping from a low-quality annotation to a high-quality one. We then use the upgrade network to enhance the annotation quality of the larger low-quality set.

We observed that our solution is applicable to at least three types of annotation inconsistencies (omission, inclusion, and bias), that it is robust to changes in the annotation quality of the training set, and that it can have wider applicability than existing works. We showed that our work can be applied to enhance the low-quality predictions of a network trained on an insufficient number of samples. Finally, we showed that the networks trained on data sets enlarged by our method present higher segmentation scores than only training on high-quality data.

Chapter 4

Few-Shot Cell Segmentation

4.1 Introduction

Recently, deep learning (DL) has become an integral part of many imaging tasks, showing accurate results for problems such as image segmentation [106], a process that labels every pixel of an image into categories. Despite their potential, DL solutions are less applicable in scenarios where the annotated data are scarce [16], such as medical or biological image segmentation. These settings require trained experts to produce the annotations needed to train DL algorithms. Few-shot learning (FSL) techniques present potential solutions for these data-scarce domains by exploiting supervised information from a data-rich source task to adapt to the target task by only utilizing a limited number of labelled samples of the target task [163]. Despite the apparent suitability for cell segmentation, there is a lack of research targeting few-shot segmentation of new structures in cell data sets when other labelled structures are available. Moreover, the particularities of cell imaging make existing few-shot medical image segmentation approaches [160, 127, 138] unsuitable for cell segmentation. Thus, there is a need for a few-shot segmentation method targeted towards cell segmentation.

In FSL, we assume the availability of a relatively large amount of annotated samples

This chapter is based on:

> Ș. Vădineanu, D. M. Pelt, O. Dzyubachyk, and K.J. Batenburg. "From Feature Maps to Few-Shot Cell Segmentation". *International Workshop on Medical Optical Imaging and Virtual Microscopy Image Analysis*. Springer Nature (2025).

for a source task as a *training set*. For a different target task, only a few annotated samples are available, called the *support set*. The challenge is to effectively derive unique representations for the target task from the support set. Subsequently, these representations are leveraged to predict unlabelled samples, known as the *query set*. Given its broad definition, tackling the FSL problem can include domain adaptation [81, 39], image augmentation [27], or visual prompting [79]. Here, we focus on semantic segmentation, where the goal is to segment structures with limited annotations within a data set, leveraging sufficient annotations for other classes of structures. Such a scenario may prove especially suitable for cell segmentation, since, besides requiring domain expert annotators, the number of structures to be annotated within an image is large and the process is tedious due to the varying cell size. Additionally, adapting from one cell type to another may be possible with the limited amount of annotations involved in FSL due to morphological similarities among certain cells.

Despite the promising applicability of FSL techniques to cell segmentation, there is a limited amount of research targeting few-shot segmentation of new classes in cell data sets. Segmenting new classes with FSL is, however, more widely attempted in medical imaging. In this case, one technique that many works rely on is attention-guided segmentation [127, 55, 170], where the activations generated from the support images are used to weigh the activations of the query images. Another popular category of works uses prototype learning [160, 141, 112], where prototype vectors learned from the support set are compared against the features extracted from the query images to generate predictions. Although these approaches perform well in organ segmentation, where the structures are relatively large, morphologically dissimilar, and located in relatively fixed positions, they are not entirely suitable for cell segmentation, where structures do not necessarily fit into the aforementioned pattern. For instance, one difference from organ segmentation lies in the varying cell positions within tissues. This affects attention-based FSL methods since guiding the segmentation of the query based on the attention provided by the support requires alignment between the target structures from these images. This alignment issue is also acknowledged in [122], which motivates the authors to employ the prototype learning paradigm. Prototype learning solutions compare prototype vectors against the feature maps generated in the last layers of an encoder, which results in low-resolution predictions. This can hinder the segmentation of cell microscopy images, which generally contain clusters of cells, since the lack of resolution would not allow for the delineation the individual cells within the clusters. Besides the methods designed for few-shot medical image segmentation, there are many developed for natural images [122, 173, 90]. However,

their applicability has not been extensively explored for (bio)medical imaging.

In this chapter, we propose a novel few-shot segmentation solution designed for cell segmentation. We train mixed-scale dense (MSD) networks [116] as feature extractors on the training set and then we use the support set to learn a linear combination of the extracted features that can be applied to segment a new class of entities. We account for limitations of previous works, such as the low resolution of the prototype-learning predictions, by producing features of the same spatial dimensions as the input image. Moreover, unlike attention-guided methods, we do not require similar positioning between support and query structures since we disentangle the adaptation step on the support from the query prediction. Also, since we only learn a low number of weights for the linear combination, the adaptation step can be performed rapidly, enabling easier prototyping.

4.2 Background and Methodology

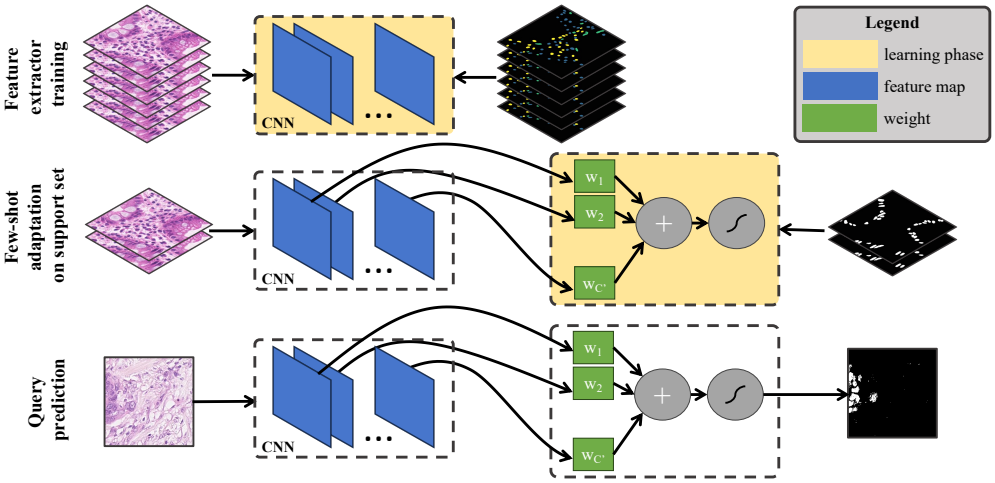


Figure 4.1: Workflow. We train feature extractors on the known classes. Consequently, we extract feature maps from the support set to learn a recombination for predicting the new class. Lastly, we apply the learned recombination to the query images.

We consider the case of segmentation of 2D vector-valued images, e.g., RGB or grayscale, where the aim is to learn a mapping from a matrix of pixels with L rows, M columns, and C channels $x \in \mathbb{R}^{L \times M \times C}$ to the target $y \in \mathbb{Z}^{L \times M}$, where each pixel of y has a value reflecting the entity it belongs to. One method of approximating

this is with convolutional neural networks (CNNs) $f_\delta : \mathbb{R}^{L\times M\times C} \to \mathbb{R}^{L\times M\times N_c}$, where N_c is the number of classes and whose parameters δ are learned from a training set $X_{\text{tr}} = \{(x_1, y_1), (x_2, y_2), ..., (x_{N_{\text{tr}}}, y_{N_{\text{tr}}})\}$, with $N_{\text{tr}} = |X_{\text{tr}}|$.

Few-shot image segmentation requires an initial training set X_{tr}, with the set of classes C_{tr}, which is used to train the model's parameters. The model is employed to predict a new set of classes $C_{\text{te}} : C_{\text{te}} \cap C_{\text{tr}} = \emptyset$ by only relying on K annotated samples (shots) for each of the new N classes (ways), where $N = |C_{\text{te}}|$, making the segmentation task an N-way K-shot problem. For the remainder of this chapter, we will consider 1-way segmentation problems, i.e., the binary segmentation of the new class. The K annotated shots comprise the *support set* $X_s = \{(x_s, y_s)\}$ from which the model distils knowledge about the new class to produce a segmentation of the unannotated *query set* $X_q = \{x_q\}$.

For our method, summarized in Figure 4.1, we consider feature extractor CNNs capable of generating feature maps at the same spatial resolution as the input image. We train the feature extractors on the classes known in the training set. Specifically, in this work, we train a binary segmentation model for each of these known classes. Consequently, we use the feature maps of the images from the support set generated by the trained feature extractors to learn a set of weights for recombining the maps to predict the new class. Finally, we employ the feature extractors and the learned weights to predict the query images. In this work, we choose MSD networks [116] as feature extractors. MSD bypasses the need for downscaling and upscaling the feature maps for capturing features at different scales by replacing standard convolutional kernels with dilated convolutions [70]. Since each feature map has the same spatial dimensions as the input image, this network can localize well the individual cells, generating activation areas that correspond to the actual position and shape of the cells within the image. Preserving the spatial dimensions of the feature maps also enables the network to densely connect its layers, allowing MSD to produce accurate results with relatively few trainable parameters. The low parameter count implies that MSD is less prone to overfitting [158], making it a well-suited feature extractor for data-scarce domains, such as medical image or cell segmentation.

We decompose the feature extractor network f_δ into a feature maps generator g_ϕ and a predictor o_ϵ. Therefore, we have $f_\delta = o_\epsilon \circ g_\phi$, where $g_\phi : \mathbb{R}^{L\times M\times C} \to \mathbb{R}^{L\times M\times C'}$ uses the parameters ϕ to generate C' feature maps from the input image and $o_\epsilon : \mathbb{R}^{L\times M\times C'} \to \mathbb{R}^{L\times M\times N_c}$, parametrized by ϵ, outputs the prediction from the feature

maps, with $\phi \cup \epsilon = \delta$. We begin by training the feature extractor on the training set:

$$\widehat{\phi}, \widehat{\epsilon} = \underset{\phi,\epsilon}{\operatorname{argmin}} \sum_{(x,y)\in X_{\mathrm{tr}}} L(o_\epsilon(g_\phi(x)), y), \tag{4.1}$$

where L is a loss function. Consequently, we employ the feature maps generator to learn the weights $W \in \mathbb{R}^{C'}$ and intercept $b \in \mathbb{R}$ of a perceptron in the few-shot adaptation step:

$$\widehat{b}, \widehat{W} = \underset{b,W}{\operatorname{argmin}} \sum_{(x,y)\in X_{\mathrm{s}}} L(\sigma(b + g_{\widehat{\phi}}(x) \cdot W), y) + \lambda \|W\|_2, \tag{4.2}$$

where $\sigma : \mathbb{R}^{L\times M} \to \mathbb{R}^{L\times M}$ is an element-wise activation function. The $\|W\|_2$ regularization term is included because we noticed its benefit in 1-shot cases, where overfitting can become more likely. Equation 4.2 enables us to create a new linear combination of the feature maps, suitable for the new class in the support set. Finally, we apply the weights to predict the query images as $\widehat{y} = \sigma(\widehat{b} + g_{\widehat{\phi}}(x) \cdot \widehat{W})\ \forall x \in X_q$.

When utilizing a cross-entropy loss, Equation 4.2 becomes a logistic regression task [108] for which highly efficient implementations are available [53]. For other loss functions, e.g., Dice loss, we use a second-order optimizer, which has several advantages compared to first-order approaches (e.g., faster convergence and better robustness to hyperparameter settings [171]). Second-order optimizers are typically not suitable for deep learning due to their high computational costs when optimizing a large number of parameters. However, the number of weights of our perceptron is relatively low, making second-order optimization a viable choice. Since second-order optimization methods perform best when the initial guess of the parameters is close to the optimum [12], we use logistic regression to provide this initial guess.

4.3 Experiments

4.3.1 Experimental Setup

We implemented our experiments in PyTorch [114]. For training the feature extractors, we partition our data into training and validation with an 80/20 ratio and stop the training when the validation score does not improve for more than 10 consecutive epochs. We use the Dice loss function and ADAM [82] optimizer. For obtaining the perceptron's weights W and intercept b, we employ BFGS [169] with Dice loss as the objective function. We choose the regularization parameter in Equation 4.2 by visually

Method	ABD-MRI*	ABD-CT*	Lizard†	MoNuSAC†
PANet	46.75	28.95	10.08	27.77
SE-Net	47.45	39.242	10.16	23.12
GCN-DE	67.3	61.73	19.04	21.79
SSL-ALPNet	70.12	65.05	6.01	18.59
BAM	-	-	5.4	9.77
Ours	-	-	48.76	48.27

Table 4.1: The average Dice score [%] on the test set of state-of-the-art medical and natural image few-shot segmentation models. $*$: Results taken from [170]. †: Results generated by following the open-source implementation of the methods.

assessing the predictions of several random selections of support and query images. The optimization stops when the gradient norm is lower than 10^{-5}. To report the results, we use the Dice coefficient on a separate test set, using 5 instances of trained feature extractors with 10 randomly sampled support sets (50 results) per experiment.

4.3.2 Data

We chose Lizard [59] and MoNuSAC [153] segmentation data sets, containing, respectively, 291 (191 train, 100 test) and 410 (310 train, 100 test) 8-bit RGB H&E stained tissue images of various sizes. From Lizard, we keep epithelial (E), connective (C), lymphocyte (L), and plasma (P) classes, whereas from MoNuSAC we use epithelial (E), lymphocyte (L), macrophage (M), and neutrophil (N). For both data sets, we separate each image into multiple 256×256 patches via a sliding window technique with a stride of 64 pixels. Additionally, to show the performance of the FSL methods designed for medical image segmentation, we use ABD-CT from [89] (30 3D CT scans with 1755 slices) and ABD-MRI from [80] (20 3D MRI scans with 492 slices). From both data sets, we report the results on four classes: liver, spleen, left kidney, and right kidney.

4.3.3 Results

Existing Works on Cell Segmentation. We used open-source implementations, provided by their respective authors, of SE-Net [127] and PANet [160], two methods that constitute the seminal works in medical imaging for attention-guided few-shot segmentation, and for prototype learning, respectively, as well as of GCN-DE [138] and SSL-ALPNet [112], two well-performing derivations of SE-Net and PANet, respectively. Also, we explore the results of BAM [90], a recent method with state-of-the-art

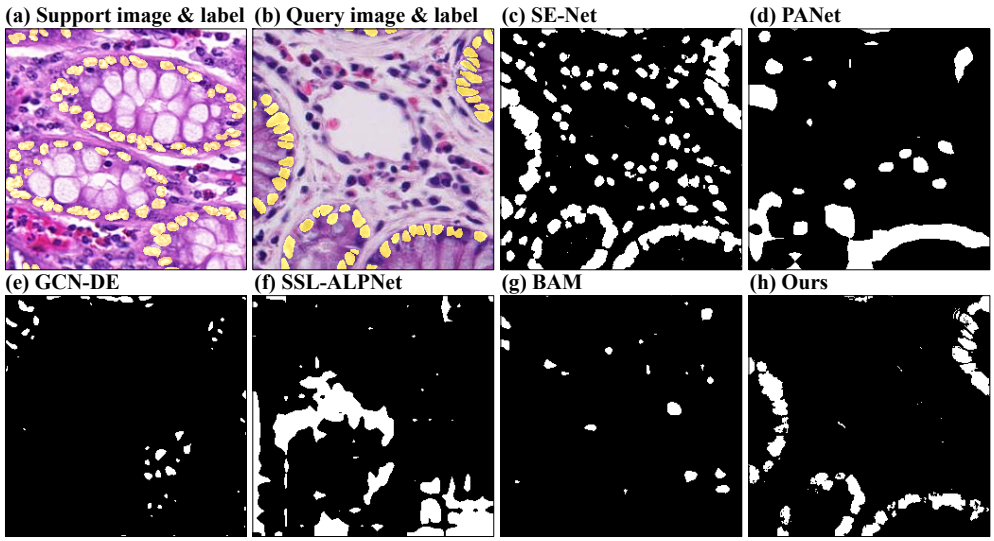


Figure 4.2: Visual comparison of predictions of epithelial cells from Lizard by methods designed for natural and/or medical image segmentation, and our method.

results for natural image segmentation. We applied these techniques on the cell data sets whose results we report in Table 4.1. The results on ABD-MRI and ABD-CT, taken from [170], correspond to a one-shot setting, while for Lizard and MoNuSAC we used five shots in the support set for PANet, BAM, GCN-DE and our method, while for SE-Net and SSL-ALPNet, we employed one support image since these methods do not allow pairing a query image with multiple support images. Since BAM has not been applied to medical imaging, we do not show results for it on ABD-MRI and ABD-CT. We notice that although these methods show good results for the task they were designed for, i.e., few-shot organ segmentation, their performances do not translate to few-shot cell segmentation. In this context, they achieve considerably lower scores compared to our method. Figure 4.2 shows a qualitative comparison between the aforementioned methods and our solution on the Lizard data set where the unknown class is the epithelial cell type. The other cell classes were used during training. For the methods allowing multiple shots, we used five. For the others, we used one shot. In Figure 4.2, we only show the support image common to all methods. We observe that SE-Net, GCN-DE, and BAM show difficulties in adapting to the new cell type. SE-Net segments most cell-like structures within the query image, whereas the predictions of GCN-DE and BAM contain structures belonging to the cell types from the training set. The prototype-based solutions, i.e., PANet and SSL-ALPNet, show

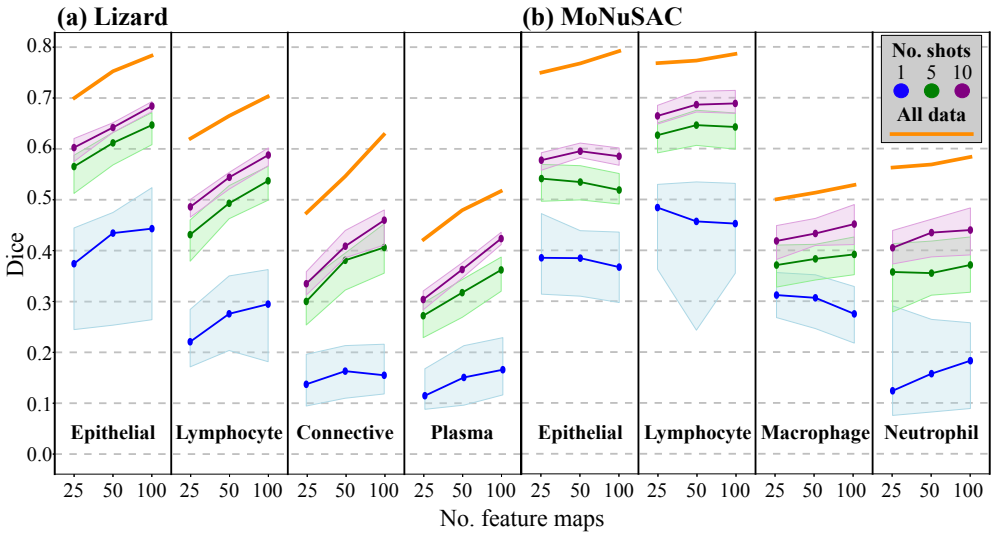


Figure 4.3: Figures (**a, b**) show the Dice score of our method for different numbers of shots and feature maps. The orange line shows the average Dice score on the test set of the models trained on all labelled data. The data points reflect the median Dice score on the test set, while the shaded area is defined by the first and the third quartiles.

comparatively better adaptability. PANet includes a region with the epithelial cells in its predictions, with the same area also covered by SSL-ALPNet. However, both methods present relatively large areas of false positive segmentation. In this example, our method shows the best coverage of the target cells, while introducing the least amount of false positives.

Performance Analysis. For this experiment, we trained feature extractors of 25, 50, and 100 layers, on three classes, while selecting 1, 5, and 10 images whose labels we include in the support set. We trained the feature extractors as individual binary segmentation networks, each covering one of the classes; four binary segmentation networks in total. For every target class, we include the feature extractors trained on the other classes within the few-shot adaptation step. We present the results generated by the feature extractor trained on the target class as an upper-bound baseline. The results reported in Figure 4.3 show that offering the perceptron more features for recombination, i.e., training deeper feature extractors, generally results in improved Dice score, especially for higher numbers of shots. This behaviour is expected, since a richer set of features means more flexibility in choosing a more general combination separating the new cell type from the other structures in the image. Also, we notice that, due to the random nature of the few-shot allocation, adapting the perceptron

Optimization	Lizard				MoNuSAC			
	E	L	C	P	E	L	M	N
Logistic [%]	61.37	50.26	34.57	32.48	49.1	63.63	37.6	37.54
BFGS [%]	**64.66**	**53.89**	**40.27**	**36.21**	**51.86**	**64.45**	**39.18**	**37.57**

Table 4.2: Median Dice score results on the test with weights trained by logistic regression and further refined by BFGS.

on only one shot is detrimental to its performance because of the risk of picking a less representative or less informative support image, such as an image with a small amount of annotated cells. This issue is reflected in the large blue-shaded areas corresponding to 1-shot results, showing high variability in the performance of our method. However, when utilizing 5 shots, we notice more robust results that are less affected by the chosen support set. Also, since we learn the perceptron's weights using a second-order optimizer, the adaptation step is performed quickly, averaging 9 seconds on a Nvidia RTX 3070 GPU, allowing eventual flaws in choosing the support set to be quickly detected and corrected in practice.

Few-Shot Adaptation Choice. We assessed the benefit of utilizing the weights obtained via logistic regression as a starting point for a second-order optimization algorithm with Dice loss. We conducted the experiments by randomly selecting 5 support images of the target class on which the perceptron was trained via logistic regression. Consequently, we employed BFGS to optimize the resulting weights with the Dice loss as the optimization function. Table 4.2 illustrates that the additional optimization step is beneficial for the performance of our method with an average Dice score gain of 10.14% for Lizard and 2.8% for MoNuSAC.

4.4 Conclusion

Cell segmentation is an important annotation-scarce task that can benefit from few-shot learning, but for which existing methods are unsuitable. Here, we present a novel few-shot segmentation method designed to account for the particularities of cell segmentation, such as the varying position of the target structures and their proximity to each other. To achieve this, we utilize the high-resolution feature maps generated by MSD networks [116], trained on the known classes, as input to a perceptron, which we adapt to the few shots of the new class. We showed that our method can be successfully applied to cell images, requiring as little as five annotated images in the support set for producing Dice scores less than 20% lower than of models trained on several

hundred annotated images. In the future, we aim to improve the reliability of our solution by exploring other types of feature extractors, incorporating additional regularization techniques, or using ensemble methods. Moreover, to better contextualize our results, we intend to provide additional comparisons with popular fully-supervised cell segmentation methods such as UNet [125] and Hover-Net [60].

Besides being used as a standalone cell segmentation tool, our solution can also be embedded into an active learning setup where the quick adaptation step would enable the user to immediately choose an appropriate support set where its predictions can constitute the base for a further refinement step, e.g. as in [159]. In both cases, our method can significantly reduce the amount of training annotations necessary for costly segmentation tasks. For instance, within a semi-automated annotation tool, our solution can produce initial suggestions of annotations, which can then quickly be corrected by experts, while a fully-supervised model trains in the background on the rectified annotations such as in [61].

Chapter 5

Explainability and Annotations with Activation Maps

5.1 Introduction

Automating archaeological feature detection and classification on remotely sensed imagery is increasingly becoming possible. Until recently, the reliability of object-based solutions, i.e., the partitioning of remote-sensing images into categories [37], suffered from the sensitivity of algorithms to image variations, e.g., in contrast, or brightness, or from the heterogeneity of archaeological objects as algorithms expect homogenous entities [88]. While these methods were once considered improvements over pixel-based classification (as early examples, see [31, 33]), object-based methods [40] could not be fully used for automatically detecting the relevant archaeological features of remote-sensing images.

However, a new approach is forming thanks to the fast-emerging deep learning

This chapter is based on:

> Ș. Vădineanu, T. Kalayci, D. M. Pelt, and K.J. Batenburg. "Convolutional Neural Networks and Their Activations: An Exploratory Case Study on Mounded Settlements". *Journal of Computer Applications in Archaeology, 7(1).* ubiquity press (2024).

paradigm that intends to bypass earlier obstacles by learning definitory patterns of the target objects directly from the data. Due to recent advancements in hardware technology and the availability of abundant data, machine learning, especially deep learning, algorithms have seen widespread and rapid adoption in many domains, including archaeology. Deep learning algorithms particularly achieve state-of-the-art performance via convolutional neural networks (CNNs) for many image processing tasks, such as image classification.

In archaeology, Bayesian regularization and Levenberg–Marquardt algorithms have been compared for predicting metrics of Neolithic laminar artefacts [146]. Similarly, machine learning algorithms have been employed to cluster cultural and technological groups within archaeological datasets [147]. Deep learning approaches have also proven successful in detecting and segmenting archaeological structures from LiDAR data [63] and in semi-automatically mapping archaeological topography using airborne laser scanning data [145]. CNNs have facilitated the detection of "princely" tombs [25], and have revealed shell-ring building practices by Archaic Native Americans [38]. Additionally, deep learning-based automated analysis has been applied to archaeo-geophysical images, enhancing the interpretation of geophysical survey data [87].

Nevertheless, the new paradigm already signals it is not devoid of problems such as the requirement of large quantities of high-quality annotated data, high computational costs, and the opacity of the CNNs' decision process. In this chapter, we highlight two of these key issues that might benefit from further research: the annotation cost and the explainability of network architectures. As a constructive approach to address these issues, we present ways to link annotation and explainability problems through visualizations, supported by exploratory statistics.

The annotation problem is particularly relevant to archaeology. While deep learning algorithms are highly effective for numerous imaging tasks, their training demands substantial annotated data. The challenge lies in generating annotations, especially in specialized domains such as archaeological satellite imagery, where annotations are often created by trained experts with limited availability [15, 73]. Annotated data scarcity becomes particularly problematic for labour-intensive tasks, such as segmentation, which requires classifying the pixels within an image. Such constraints can impede the practicality of deep learning applications. Therefore, addressing the challenge of annotated data scarcity can play an important role in the further adoption of deep learning in archaeological research.

We also observe that achieving high accuracy is the main concern in the schol-

arship. When provided with sufficient training data, recent deep learning models generally produce highly accurate detection and classification results regardless of the architecture. Yet, the influence of architectural choices over which image features contribute towards a prediction receives less attention. This perceived opacity of neural networks' decision-making process may contribute to some research fields approaching their use with caution. Therefore, besides alleviating the burden of extensive manual annotation, visualizing what the most relevant image areas are for a given prediction can build trust among practitioners. Moreover, such insights can assist the experts in developing less biased workflows [100, 142], rectifying mis-annotated samples or discovering new patterns in the images.

To address the two key issues outlined above, we utilize explainability techniques, i.e., methods producing visual interpretations of a CNN's output in relation to its input, whose results we refer to as activation maps. Particularly, we focus on the explainability techniques producing activation maps reflecting the contribution of each individual input pixel towards a CNN prediction. To address the annotation scarcity and the perceived opacity of deep learning, we employ the resulting activation maps as sources of both cheap annotations and insights into the patterns found by CNNs. We address the annotation task by proposing an automated annotation pipeline for generating segmentation masks of archaeological sites from the activation maps extracted with explainability techniques. We apply these techniques to trained classification CNNs, whose training annotations are relatively cheap to produce compared to segmentation masks. We also explore to what extent we can extract meaningful visual insights from the features deemed relevant by different types of CNN architectures. We compare the activation maps extracted from multiple network architectures and study which parts of an archaeological feature contribute the most to the network's predictions. Additionally, we verify whether the highlighted features can signal the presence of mis-annotated images or overfitting. Our integrated workflow helps us to explore the annotation and explainability issues in tandem.

In our workflow, we employ Occlusion Maps [57], LayerCAM [73], and Guided GradCAM [129] as explainability methods. To combat the lack of spatial resolution associated with existing techniques, we also propose an extension to Guided GradCAM. As a case study, we map the extent of ancient settlement mounds within CORONA satellite images in the Upper Khabur Basin of Upper Mesopotamia. We apply these four explainability techniques to three widely used CNN architectures: VGG [132], ResNet [66], and DenseNet [71]. Finally, we explore activation maps to localize ancient settlement mounds using CNNs trained for binary image classification by employing

only image-level annotations.

Our aims are twofold: (i) providing an analysis of the visual cues that contribute to CNNs' predictions of sites from remote sensing images of the Upper Khabur Basin and (ii) using visual cues from activation maps as sources for segmentation annotations. To achieve these goals, we utilize existing explainability techniques and we also propose a new method for extracting activations that better match the expert interpretation of a site than existing works.

5.2 Background

5.2.1 The Study Area

The Upper Khabur Basin is located within the larger gently undulating plain of Upper Mesopotamia that stretches east-west between the massive Anti-Taurus Mountains in the north and the short mountain range called Jebel Sinjar in the south [42]. The Abd-al Aziz mountain ridge rising across Sinjar also bounds the study area. The primary contributor of the hydrological system is the Euphrates River. Running down from the northwest of Lake Van at an approximate altitude of 3,500 meters, the river significantly drops its gradient as it further moves into the Upper Mesopotamian plain, in modern-day Syria. The Khabur Basin (Figure 5.1) takes its name from the Khabur River, the largest tributary to the Euphrates.

In the Upper Khabur, several wadis (Aweij, Khanzir, Jaghjagh, Jarrah, Kuneizir, and Rumeilan) run in north-south direction eventually draining to Wadi-el-Radd [42, p. 173]. Wadi is an Arabic term denoting a valley-like morphological feature that is dry except during periods of rainfall. Even if they were temporal and usually short-lived, flowing water contributed to the geography and life. Therefore, they "played an important role for human societies within this area and many archaeological sites—often tells (settlement mounds)—are located along them." [41, p. 337] (Figure 5.2).

5.2.2 Settlement Mounds

The long-term accumulation of everyday-life cultural material through centuries results in a particular site type, called a settlement mound [126]. These are signature settlements in southwest Asia, getting the names of tell in Arabic, tepe or chogha in Farsi, or höyük in Turkish [101]. Yet, it is important also to note that other regions in the world also host mounded settlements, including Greece [43] and Hungary [113]. Depending on their (post-)depositional processes, density and duration of occupation,

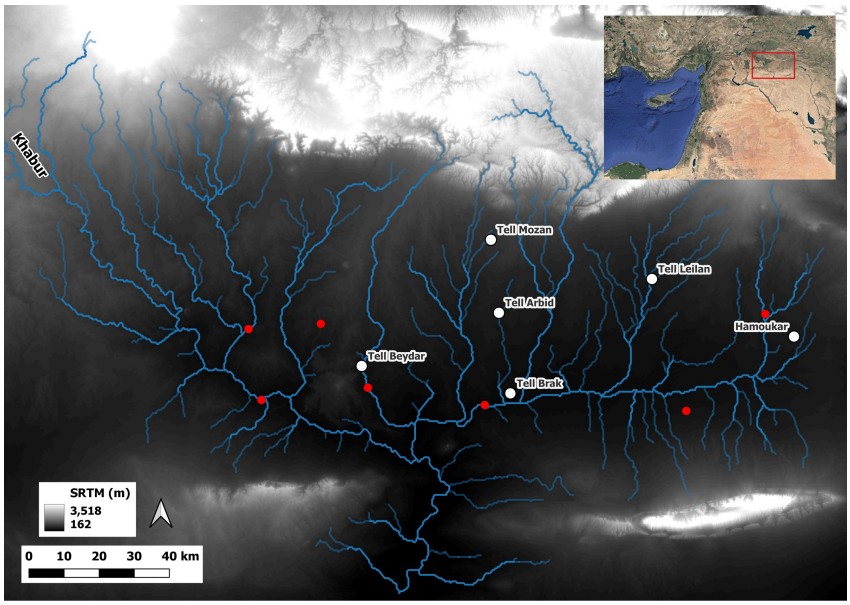


Figure 5.1: We investigated mounded sites in the Upper Khabur Basin of Upper Mesopotamia. One can see the 100-kilometre-long Sinjar mountain range in the lower-right corner. The Abd-al Aziz mountain ridge is across the river from Sinjar. Some key settlements in the area are shown in white. Red dots indicate the locations of other settlements discussed in the text.

local geological and geographic conditions, and many other factors mounds exhibit considerable differences. These differences, however, bear the potential for morphological analysis [149, 23]. Mound morphology is almost always variable, but it is possible to identify some broad trends also in our study area. Using the results of Tell Beydar Survey [150] and Tell Hamoukar Survey [148] one may summarize site morphologies, but only briefly and only with great generalization: due to less intense occupation, smaller/lower mounds were formed primarily during early prehistoric times. Rapid nucleation during the second half of the Early Bronze Age (mid-second millennium BCE) resulted in taller and more prominent mounds. From the Late Bronze Age onwards, including the Iron Age, less intensive occupation was attached to the now-abandoned Early Bronze Age mounds. This new phase of nucleation added further to morphological complexities. Lower-density occupation in later periods [97] must have contributed less to the formation of mounds.

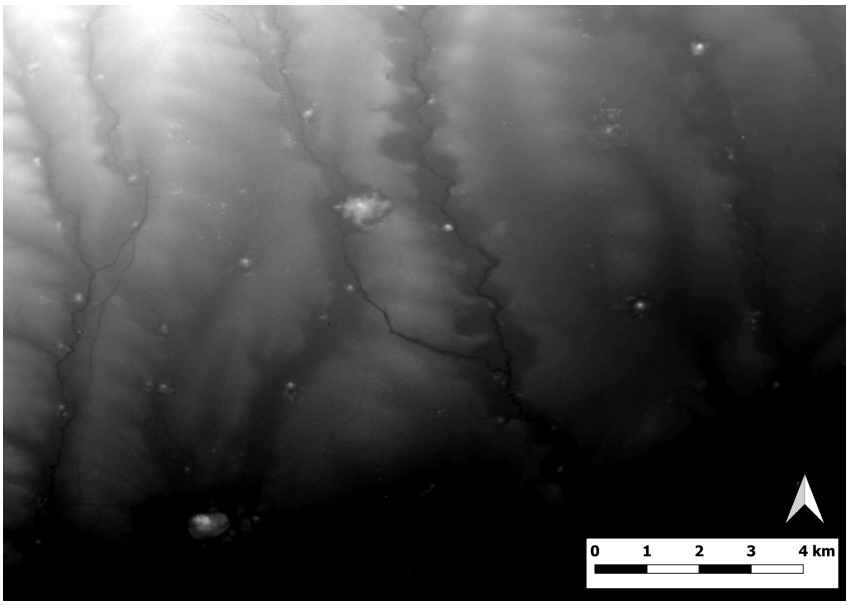


Figure 5.2: "Spots" on TanDEM interferometric SAR imagery indicate settlement mounds. They are located mainly along north-south running wadis. The large spot/site at the centre of the image is Tell Sharisi (36.891° N, 41.365° E). In the bottom left, we see another mound, Tell Farfara (36.825° N, 41.334° E).

5.2.3 Corona Satellite Imagery

In the study area, tell-sites can be as high as 40 meters and can attain sizes of more than 100 hectares [168]. Due to their considerable sizes and relatively defined site extents, but also thanks to the moderately flat topography of the study area, ancient settlement mounds are visible on remote sensing data, but notably on historical CORONA spy-satellite imagery [119, 24]. In return, it is possible to conduct desktop surveys with visual interpretation [23] and with additional products, such as digital elevation models [103].

The state-of-the-art sensors resolve the ground in great detail and provide data from non-visible portions of the spectrum. CORONA as a historical dataset, but especially the Key Hole KH 4B series (1967–1972) contributes to landscape archaeology in other different ways. At the very least, CORONA predates the negative impact of modern irrigation systems, great dam projects [149, p. 12] and urban sprawl [167, p. 228] on material culture (Figure 5.3).

In particular, the high-resolution of KH-4B (ca. 1.85 meters at Nadir) provides an extensive coverage, mainly due to the panoramic scan. Thanks to multiple CORONA

Figure 5.3: State-of-the-art sensors, such as WorldView-2, can resolve the ground in great detail, on the right. CORONA provides historical evidence of land-use land-cover changes, on the left. In this particular example, one can assess the impact of modern buildings on Tell Beydar. Image resolutions are comparable despite the age of spy-satellite imagery.

KH-4B missions, archaeological landscapes can be investigated in time series and the most optimal scenery can be selected for further research. Recent studies highlight the potential of Hexagon [58, 64] and U2 imagery [65]. Yet they are still not widely available for wide-scale analysis. Therefore, ortho-corrected CORONA is still a viable source for exploring diverse archaeological landscapes across the globe.

5.3 Deep Learning and Activation Maps

5.3.1 Deep Learning for Image Classification

Convolutional neural networks (CNNs) explore patterns in input images through the use of units organized as filters. These filters, forming a convolutional layer, generate intermediary images known as feature maps [92], which essentially represent the prominence of specific features within the image. For example, a filter might emphasize vertical edges, while another filter could identify horizontal edges, textures, and so on. These resulting feature maps then become the input for the subsequent set of filters in the following convolutional layer.

Among the problems tackled with CNNs, we focus on image classification due to its relatively cheap annotation process and widespread relevance. The classification CNN typically comprises two main components: a convolutional part, functioning as a feature extractor, and a fully-connected part, serving as the classifier. In the convolutional part, the learned parameters correspond to the filters within the convolutional layers, while the fully-connected part of the architecture utilizes its learned weights to categorize the features extracted by the convolutional layers. The categorization is performed by reweighing and combining the feature maps in order to produce a set of class probabilities out of which the predicted class is chosen.

Different CNN architectures employ distinct strategies to produce accurate classifications, varying in aspects such as the number of layers, the filter size, and the connectivity between layers. Despite the variety in architectural choices, many CNNs perform similarly well across different tasks and data sets [74]. Moreover, although different architectures may perform similarly on a given task, their inner decision process can vastly differ, thus influencing their explainability and utility as detection tools, therefore, making the selection of suitable architectures a non-trivial problem.

Among the popular well-performing CNNs for the task of image classification, we focus on VGG [132], ResNet [66], and DenseNet [71], listed in the order of their development. All three networks showed particularly good results for the classification of natural images on the ImageNet data set [44], with each network claiming improvements over its predecessor. All three network architectures are still widely used today. Their extensive adoption, architectural differences, and the distinctiveness of remotely sensed imagery from natural images make a comparison between these networks worth exploring. Moreover, such comparison intrinsically contributes to explainability studies by assessing the suitability of the different architectures as visualization tools of relevant patterns within remote sensing archaeological data.

VGG was among the first solutions aiming to improve the classification performance of CNNs by increasing the depth, i.e., the number of layers, of the architecture (Figure 5.4a). This was achieved by reducing the size of the convolution kernels to 3x3, substantially decreasing the number of parameters per layer. VGG consists of several layers where the information is processed sequentially, using the feature maps from the previous layer as input to the next. The feature extractor architecture consists of blocks of 3x3 convolution layers followed by max-pooling layers that reduce the size of the feature maps in half by selecting from every non-overlapping group of 2x2 pixels, the pixel with the highest value. After the final max pooling layer, the resulting feature maps are spatially flattened to sets of 1-dimensional vectors, which

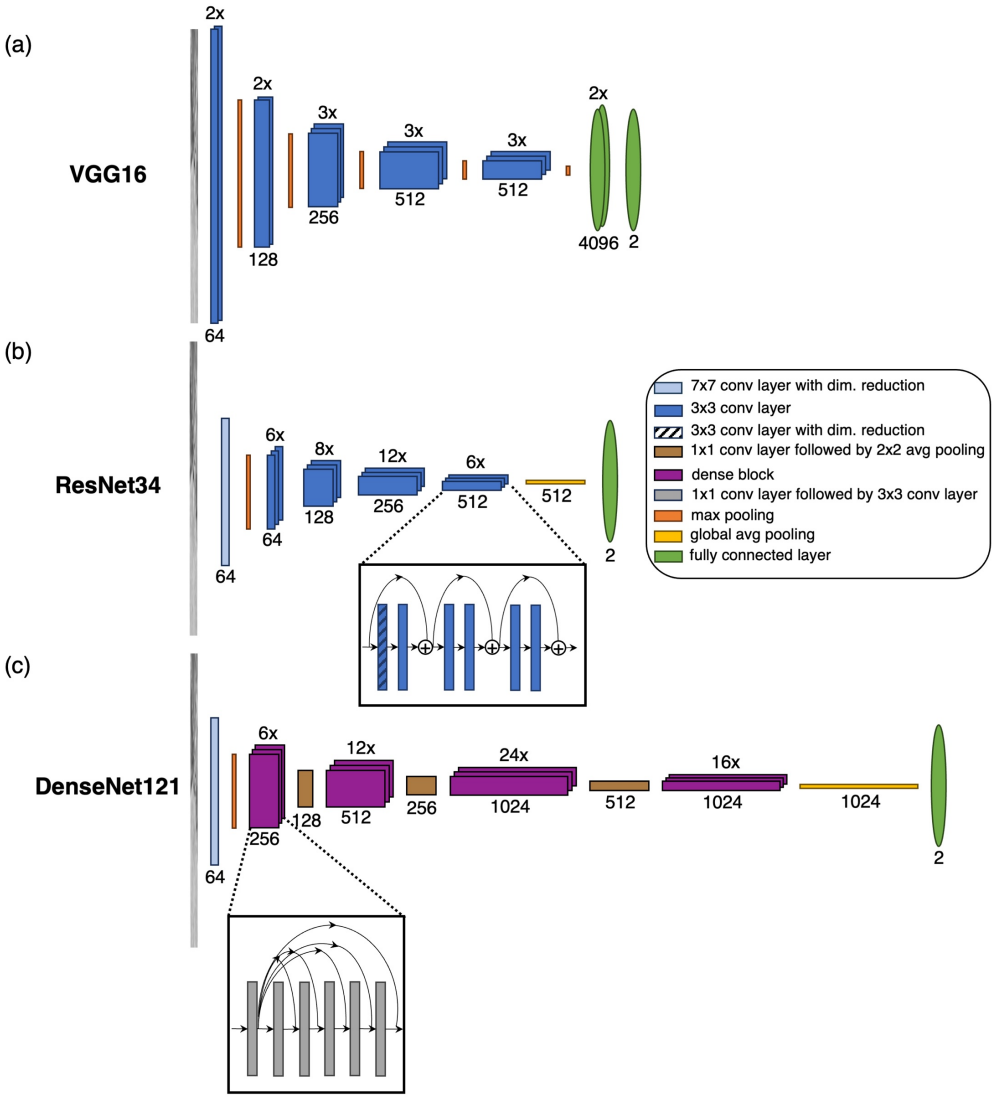


Figure 5.4: Schematic representations of VGG16 (a), ResNet34 (b), and DenseNet121 (c). The values above a block of layers correspond to the number of layers within a block. The values below the blocks refer to the number of filters each layer has.

are passed to the classifier.

ResNet is a CNN architecture that focuses on training a larger number of layers than VGG (Figure 5.4b). It achieves this by creating alternative paths that allow for the output of a layer to skip being processed by the immediately following layers.

The unprocessed output is then added to the result of the sequential path. In this way, the network focuses on learning the additions (residuals) that need to be applied to the input such that the relevant features are extracted for classification. This approach allows increasing the number of convolutional layers, enabling the network to learn more complex features. Apart from the alternative paths, ResNet differs from VGG by how it reduces the spatial dimensions of its feature maps since it replaces max-pooling layers with strided convolutions, i.e., instead of sliding the convolutional filter on the image/feature map with a step of one pixel, the step size is increased according to the stride value. After the last set of convolutions, the resulting feature maps are aggregated into a set of 1-dimensional vectors by passing the final feature maps through a global average pooling layer, which reduces a map to the average of the values across all its spatial dimensions. Averaging instead of flattening the final feature maps has the benefit of making the architecture independent of the image size, since the size of the input to the fully connected layer is dictated by the size of the channel dimension of the last convolutional layer, rather than the spatial shape of the feature map after passing the image through the convolutional and pooling layers.

DenseNet also provides a solution for training a large number of layers by developing connections that pass all the feature maps that were created by previous layers to all subsequent layers (Figure 5.4c). The architecture is composed of dense blocks, which aggregate the feature maps from all their convolutional layers, and transition blocks which provide both spatial as well as channel-wise dimensionality reduction. Similar to ResNet, the network makes use of global average pooling to make the transition from the feature extractor to classifier, while relying on average pooling instead of strided convolution for the dimensionality reduction of the feature maps.

There is already a significant number of archaeological case studies using these specific network architectures. However, limited literature is available that considers the rationale behind choosing one architecture type over another. For instance, Albrecht et al. [6] use a VGG to classify archaeological features on LiDAR data. They report they chose VGG because "this approach is accurate and flexible for the archeologist's needs" [6, p. 18]. Similarly, Somrak et al. [136] aim to detect archaeological features on Airborne Laser Scanning (ALS) data using VGG. They used this type of architecture mainly because "[t]here have been previous uses of the VGG network" [136, p. 7]. Verschoof-van der Vaart et al. [154, p. 7] provide more specific reasoning for their choice of the VGG architecture as it "performs better than most shallower networks and needs significantly less memory than some deeper networks, while yielding comparable results.". Patrucco and Setragno [115, p. 19] decided to deploy DenseNet since

"[t]his network allows using fewer channels for each layer, thus having fewer training parameters and a smaller network". Trier, Cowley and Waldeland [144] identified the problem early on. They deploy ResNet18, but also state that "the development of 'general purpose' archaeological CNNs is desirable if the discipline as a whole is to make better use of the methodology." [144, p. 168].

It is also common to use multiple networks and compare results. When multiple networks are used, the major aim is to compare accuracies, leaving little room for advancing research on explainability. Abellán et al. [1, p. 4] use "six architectures to test the accuracy in classifying tooth marks". In another study, researchers worked with seven deep learning models and their choice for the network was based on the ranking of these models [48]. Bonhage et al. [18] further solidify the accuracy problem by asking "what level of accuracy would be required from automated systems to be acceptable for a specific purpose.".

Overall, it appears that when scholars work with a single architecture, there is relatively more discussion on the reasons behind choosing that network. Nevertheless, the rationale behind their choice tends to remain implicit, restricting interpretability. Comparative approaches focus mainly on the accuracy of the results these networks can produce and make limited contribution to our understanding of how different architectures can be exploited to retrieve more information about the data itself. Using explainability techniques can benefit the scholarship as they contribute to understanding whether the image features deemed as relevant by the networks have intuitive explanations.

5.3.2 Explainability Techniques

Despite their proven capabilities in increasingly difficult tasks, one major challenge that the current CNNs are facing is a lack of interpretability of their predictions. Consequently, the applicability and reliability of these solutions can be distrusted. In response to this, multiple techniques have been developed to explain the decision process undertaken by CNNs before generating a prediction. In the context of image classification, where the prediction takes the form of class probabilities, these explainability techniques have the added benefit of providing localization information of the most relevant image sections that influenced the prediction of the CNN.

We selected three such techniques, namely Occlusion Maps (OM) [57], Gradient-Weighted Class Activation Maps (GradCAM) [129], and LayerCAM [73]. In our work, we also propose a localization technique based on Guided GradCAM [129]. All the

selected methods have the benefit of being independent of the type of CNN being used, offering good flexibility for experimenting with multiple neural network architectures. Additionally, all techniques produce easily-interpretable output under the form of activation maps, i.e., images of the same shape as the input image whose pixel values reflect the contribution of the input image's pixels towards the prediction of the network.

The working principle behind OM is that covering relevant sections within an image should drastically impact the classification result of the CNN, while covering background areas should influence the results less. Therefore, in order to find these relevant sections, a window is slid on top of the image with all the pixels within the window area being occluded (their values are set to 0). For every window position, the occluded image is set as input to the trained neural network and the difference in classification probability between the non-occluded and the occluded image is registered. After a complete pass throughout the image, the result is a 2-dimensional array of probability differences, where the highest differences denote the location of the relevant image sections.

A more invasive approach is proposed by GradCAM, which relies on processing the feature maps given by the last convolutional layer of a CNN. In general, after the training process, the initial layers of the CNN "learn" to recognize low-level features, such as edges, while the final layers recognize high-level features, e.g., the archaeological mound itself. Considering this, by analysing the output of the last layer before classification, the resulting feature maps should highlight the position of the most relevant features for the classification task. However, the information within the feature maps must be aggregated into one activation map that reflects the contribution of an image feature to the network's prediction. Therefore, the feature maps are weighed by their gradient with respect to the prediction result and their sum produces the final activation map.

A related technique is employed by LayerCAM, where the activation maps can similarly be extracted and weighed them by their gradients. However, as opposed to GradCAM, which performs this process only for the last convolutional layer, LayerCAM produces an intermediary activation map for every convolutional block within the network's architecture. The resulting intermediary maps are linearly combined to produce the final activation map, with higher weights assigned to the intermediary maps extracted from the later blocks of the network.

One common disadvantage that the three aforementioned techniques share is that their activation maps come at the cost of spatial resolution. Since OM aggregates

results for covering an entire area within an image and since it is not computationally feasible to slide the window every pixel, the resulting activation map is of a lower resolution than the input image. We can make similar observations both for GradCAM and LayerCAM which rely on the feature maps generated by the last convolutional layer of the CNN. Due to the image downscaling within the CNN, these feature maps have far lower spatial resolution than the input image, making the resulting activation maps also suffer from this lack of resolution.

Guided GradCAM proposes a solution for inferring high-resolution activations in the form of the individual contribution of each pixel towards the prediction. The image is first passed through the CNN, and then the resulting feature maps are passed backwards from the last layers towards the first ones. This process generates activation images containing clusters of pixels whose high values signify the presence of relevant features. However, these high-valued pixels are sparsely distributed, which makes delimiting the relevant features difficult. To ensure contiguous activations, we develop an addition to this method which we detail in Section 5.4.2.

Besides adding to the interpretability of the model's decision-making process, the activation maps can be thresholded to automatically create pseudo-annotations for more labour-intensive tasks such as the segmentation of the site area, i.e., the separation of the site from the surrounding area. The availability of cheap annotations can thus facilitate more experimentation with existing data sets and the development of more complex tools whose training would require prohibitive amounts of expert annotations. For instance, the generation of an image-level annotation for a classification task requires far less effort compared to creating a segmentation mask for the same image since the image-level label can be attributed after a relatively quick visual inspection, whereas a segmentation annotation involves the careful delineation of the site boundary. Thus, for existing classification data sets the generation of pseudo-annotations for segmentation would allow training segmentation algorithms with little intervention from domain experts.

5.4 Methodology

5.4.1 Data Preparation

For the study, we acquired CORONA KH-4B data from the CORONA Atlas & Referencing System [26]. Images from DS1105-1025 (November 1968) and DS1102-1025 (December 1967) cover the entirety of the Khabur Basin. For the initial desktop

survey, first, we orthorectified CORONA imagery and mosaicked them to generate a seamless coverage of the Khabur Basin. Second, we visually confirmed the location of 300 settlement mounds on CORONA. We also randomly picked 300 points to explore 'no-site' landscapes, and visually confirm areas that did not contain a settlement mound (Figure 5.5). Next, a custom-built script visited 'Site' and 'No-Site' locations and clipped a square chunk (1000 pixels x 1000 pixels) around each target. Image chunks were contrast stretched between 0 and 255 to exploit 8-bit data depth fully.

In the following step, we augmented data through rotation, swirling, and clipping. First, we rotated each chunk in cardinal directions to make four scenes available from

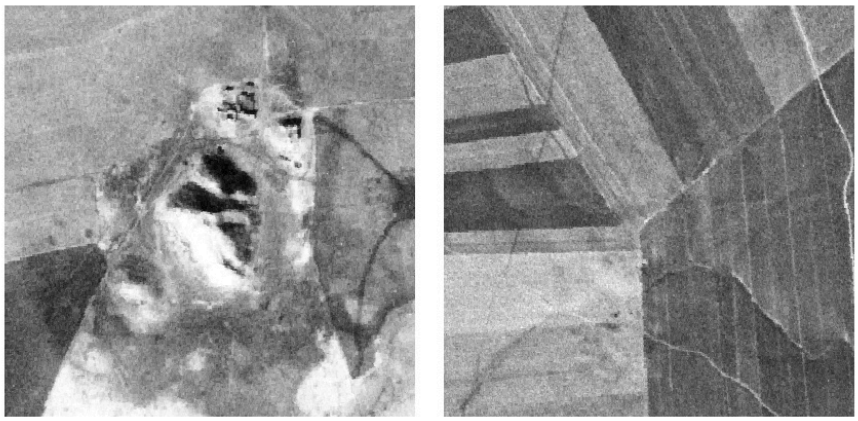


Figure 5.5: The binary classification scheme in this analysis. A CORONA image chunk with a site (left) and an image chunk with no indication of a mounded settlement (right).

the same area. Second, using the scikit-image Python package [152], we swirled all rotated images with a radius of 400 and with the parameter randomly determined from a uniform distribution with lower boundary of -2 and upper boundary of +2. These parameters ensure the pseudo-target generation mainly swirls the original site while keeping the background as intact as possible. With swirling, we aimed to mimic the relatively circular nature of sites; mounded settlements tend to have more circular footprints than rectangular site types. In the end, eight image chunks (four rotated and four swirled) with 1000 x 1000-pixel dimensions are further clipped into smaller pieces with 400 x 400-pixel dimensions. The clipping strategy involved "moving" the sites in four corners as well as keeping them at the centre of a scene. In doing so, the aim was to represent different parts of the immediate surroundings of the sites in additional images (Figure 5.6). This final clipping operation generated 40 images per

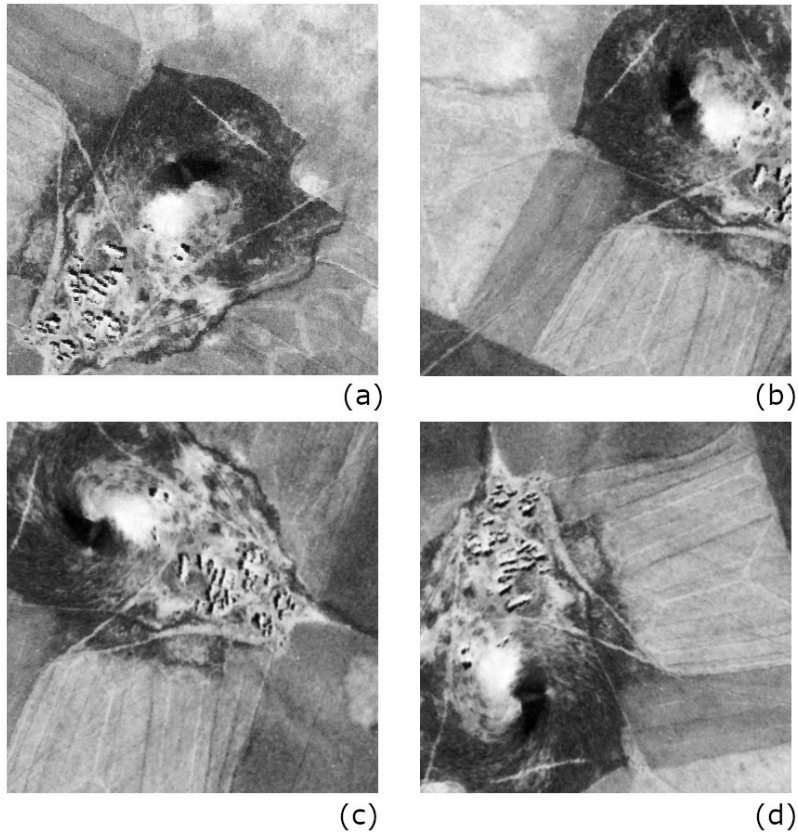


Figure 5.6: Four (out of 40) samples from the augmentation process are presented here. (a) the initial clip of a mound as documented on CORONA imagery, (b) clipping and rotation moves the site to the upper right corner while revealing a different background context, (c) rotated as in sub-figure b, but also moved to top left corner and swirled, (d) a different set of rotation, clipping, and swirling.

site. Therefore, we were able to gather 12,000 (40x300) image sets (binary code: 1) for 'sites'; and for 'no-sites' (binary code: 0). In total, 24,000 images were available for training.

5.4.2 Proposed Pipeline

For this work, summarized in Figure 5.7, we aim to utilize activation maps to derive cheap annotations that can be used for a site segmentation task as well as to formulate interpretations of the relevant areas within the images that are triggering the prediction of a network. We begin by training classification CNNs on image-level

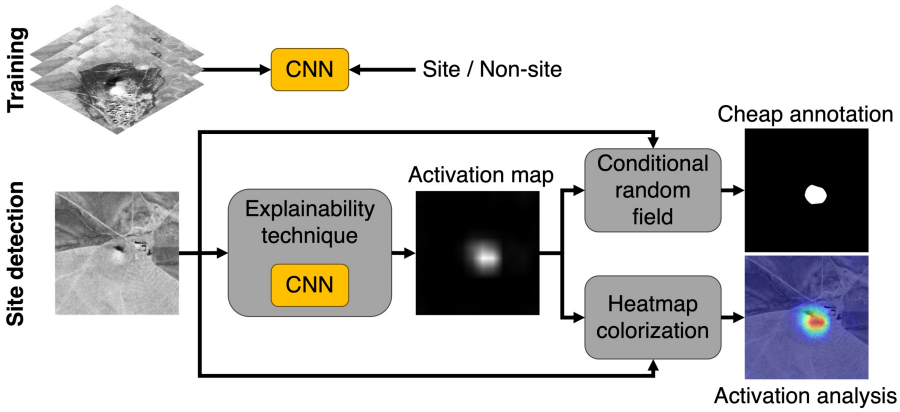


Figure 5.7: Workflow. We train CNNs for classifying whether a site is present or not in an image. We embed the trained CNN into explainability frameworks which we can use to both produce a segmentation mask of the site or to analyse the important image features highlighted by the network.

binary annotations, where a positive annotation signifies the presence of a site, and a negative annotation denotes the absence of the site from the input image. We utilize PyTorch [114] implementations of the three network architectures [99]. We treat the site detection as a binary classification task where the input to the network is a single-channel grayscale image and the output is a 2-valued vector, with the first value indicating the probability that no site is present in the image, while the second value indicates the opposite probability. We split our data into training and validation with an 80/20 ratio. For every type of architecture, we train 5 networks with different initialisations and a different random split of the data. During training, if we observe no improvement in the validation score for 10 consecutive epochs, we stop the process. We use the binary cross-entropy as the loss function, and we update the parameters with ADAM optimization algorithm [82].

Generating Activation Maps

After training, we include the trained networks in the explainability tools which produce an activation map. To get more stable activation maps, we average the results from multiple initializations of the same CNN architecture. Also, for each initialization, we create a different random split between the training and validation images. We utilized Captum [85], a model interpretability library for PyTorch models to generate Occlusion Maps and to perform Guided GradCAM, whereas for GradCAM and

LayerCAM we developed our implementations based on the original papers. In all cases, the results given by the explainability methods take the form of images, where a pixel value denotes the probability that the corresponding pixel from the input image belongs to a relevant region for the classification task.

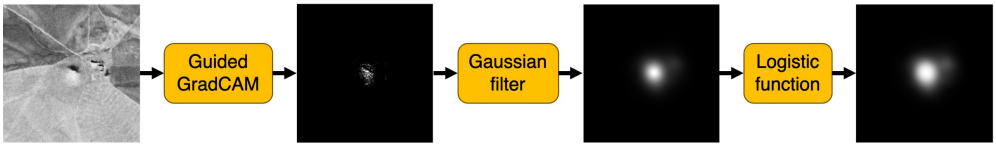


Figure 5.8: We propose an extension to the Guided GradCAM in order to tackle the reduced spatial resolution problem. Adding a Gaussian filter and a logistic function enhances image features that are comparable to annotations.

In addition, to address the lack of resolution of the activations generated by Occlusion Maps, GradCAM, and LayerCAM and the sparsity of Guided GradCAM activations, we propose an extension of the latter that aims to provide smoothness in activation areas, while maintaining the resolution of Guided GradCAM, which we present in Figure 5.8. We apply a Gaussian filter to smooth the pixel values of the activation image, therefore creating continuous activation areas. This, however, comes with the caveat of widening the gap in value between high-activation and low-activation areas, which can lead to a pessimistic estimation of the relevant image features. We compensate for this by passing the filtered activation image through a logistic function which creates a nonlinear rescaling of the pixel values such that previously low-activation areas would receive higher values. For visualizing the relevant features, we translate the activation map into a heatmap which we then overlap on top of the input image (see Figure 5.7). This results in a more straightforward analysis of the image features.

Here, we make observations based on visual interpretation. The aim is to build qualitative knowledge for how three different network architectures (VGG, ResNet, DenseNet) 'learn' what a settlement mound is, highlighted by four different activation techniques (Occlusion Map, GradCAM, LayerCAM, and our method based on Guided GradCAM). Our workflow includes selecting representative examples from the overall data set and exploring the activation maps on remotely sensed data.

From Activation Map to Segmentation Mask

After obtaining the activation maps, we process them to obtain segmentation masks. In order to do this, we translate the smooth probability landscapes provided by the activation map into hard area borders by utilizing conditional random fields (CRF)

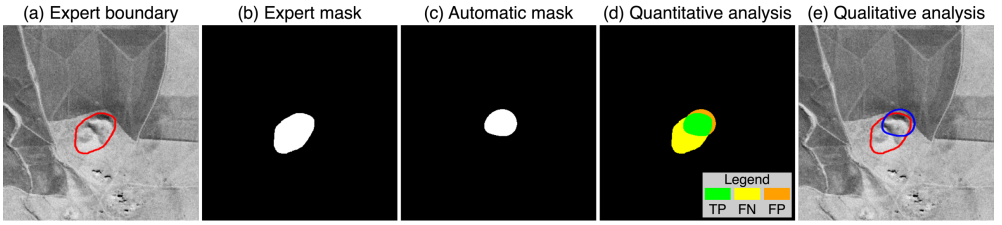


Figure 5.9: Example of an expert annotation (a), the segmentation mask derived from it (b), the automatically derived segmentation mask (c), the intersection of the expert and automatic masks (d), with green, yellow, and orange pixels representing true positives, false negatives, and false positives, respectively, and the intersection of expert and automatic boundaries (e).

[139]. The CRF considers both pixel probabilities from the activation map and the similarity between neighbouring pixels from the input image and outputs a binary image where the foreground corresponds to areas within the image occupied by relevant features and the background covers the rest of the image. An example of a segmentation mask generated from an activation map is shown in Figure 5.7.

We analyse the suitability of the segmentation masks produced by CRF both qualitatively and quantitatively by comparing them with site delineations provided by a domain expert. The human annotation process included drawing mound boundaries as they appear to the expert on CORONA images (Figure 5.9a). While site delineation is a subjective process, mound formation produced footprints easier to trace than many other site types and morphologies. City walls around some of these settlements also helped the annotation. To produce a quantitative analysis, we first binarize the human-annotated image such that the pixels within the boundary are assigned the value of 1, while the rest are assigned 0, creating a mask ready for further analysis (Figure 5.9b).

We then compute the Dice similarity score [45] between the binarized human annotation and the masks produced by the conditional random field (CRF) (Figure 5.9c) to assess the suitability of the automatically generated masks as annotations for segmentation. The equation describing the Dice score is presented in

$$\text{Dice} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{5.1}$$

where TP, FP, FN, refers to the number of true positive, false positive, and, respectively, false negative pixels between the binarized ground truth and the prediction (example in Figure 5.9d). For the qualitative comparison, we use the output of the

CRF to generate a boundary which we overlay, together with the human-generated one, on top of the input image (Figure 5.9e).

5.5 Results

5.5.1 Classification Performance

Since our primary aim is to explore activation maps in relation to future annotations, we only briefly mention overall model performances. We evaluated the performance by computing the precision and the recall. These metrics are computed based on the true positives (TP), false positives (FP), and false negatives (FN), calculated in the context of binary image classification, i.e., they count image-level class labels, rather than pixels. The precision reflects the proportion of relevant samples that the classification model is able to find, i.e., the proportion of correctly predicted sites among all site predictions (precision = TP / (TP + FP)). The recall, on the other hand, shows the ability of the model to find all relevant samples in the data set, i.e., the proportion of correctly predicted sites among all images with sites (recall = TP / (TP + FN)).

Network	Precision	Recall
VGG16	0.9996	0.9962
ResNet34	0.9994	0.9983
DenseNet121	0.9994	0.9976

Table 5.1: Validation set performance of the different architectures.

In Table 5.1, we report the classification results on the validation set of the three networks. It appears that all networks learned a good fit for the data, being able to correctly classify ('site' or 'no-site') for almost all the images. All networks present similar precision, with ResNet34 showing a larger recall than the other two —albeit only very slightly. It appears that a simple augmentation technique could generate powerful classifier models with similar performances in the study area. Nevertheless, the models are trained for a very specific site type within a particular geography. Therefore, these models' generalizability is an open question; transfer learning is beyond the scope of this chapter. On the other hand, trained networks may equally perform in areas with similar relatively flat morphologies hosting settlements with mound morphologies, such as Neolithic Thessaly [7, 118].

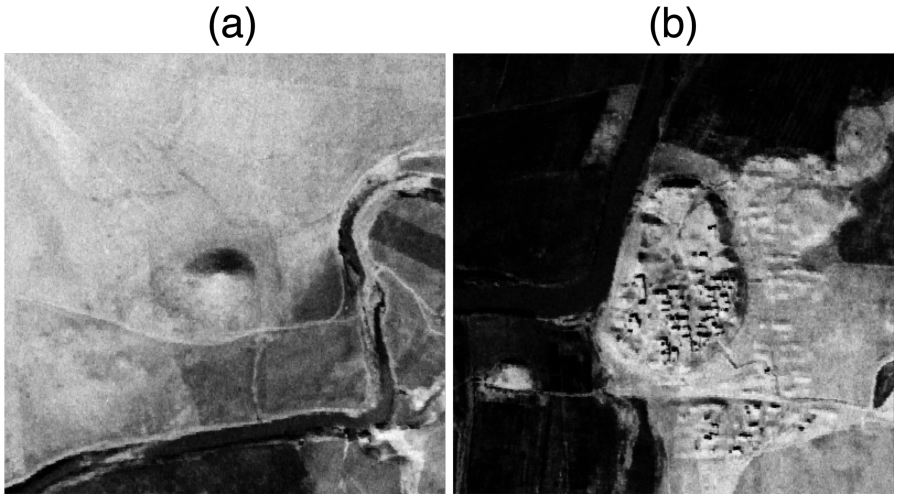


Figure 5.10: Two sites with simple (a) and complex (b) morphologies.

5.5.2 Analysis of Activation Maps

To assess the interpretability of the selected network architectures, we perform a visual analysis of the activation maps produced for sites with both simple (Figure 5.10a) and complex (Figure 5.10b) morphologies.

We begin by analysing the activations of the simple morphology (Figure 5.11). The Occlusion Map and our method fit well to the human interpretation of a site boundary for all three networks. GradCAM and LayerCAM activations exceed the site boundaries, especially for DenseNet. DenseNet produces wider activation areas, owing to its approach of aggregating information from multiple layers, thus being activated by a wider set of image features than the other two architectures. GradCAM is of particular interest since the highly activated area appears to have no immediate connections with the shape or the shadow, the two prime indications of a mound for the annotator.

Studying a more complex morphology reveals that activations can be discontinuous. In the current example, the site is dotted with modern structures, potentially adding complexity to network training. All three architectures are activated more in the north (Figure 5.12). Incidentally, this portion is cluttered less by later human occupation. It is also possible that the shadow generated more contrast against the background for the high-level features, resulting in a northerly activation. Finally, we note that only VGG is successful in identifying the smaller mound at the lower-left corner. Conforming with

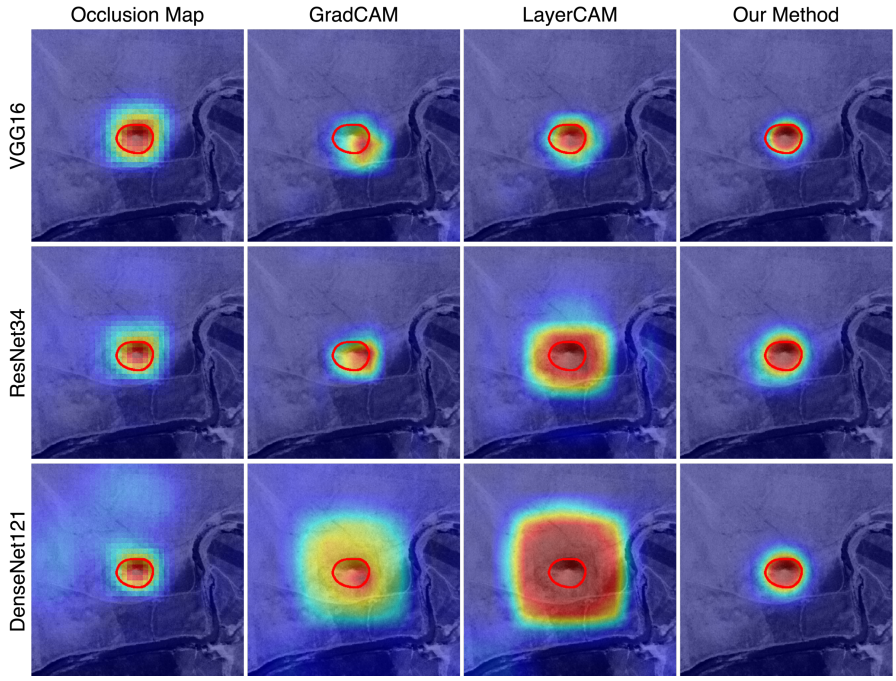


Figure 5.11: Activations of a single conical mound (36.832° N, 40.229° E). The red line corresponds to the expert annotation.

the previous example, the Occlusion Map and our method provide the activations that match more closely the human intuition for this smaller and circular feature.

The activations of both types of sites show that, across all network architectures, the predictions were influenced by actual archaeological features within the images. For instance, in Figure 5.11 all activations are centred on the small conical mound, whereas, in Figure 5.12, parts of the elevated area of the site are highlighted by all explainability techniques.

5.5.3 Activations as Sources of Annotations

For this experiment, we use the conditional random field (CRF) to process the activation maps into site segmentation masks. For ease of comparison with the expert annotations, we represent these masks as boundaries applied on top of the input image. Besides visual inspection, we also numerically assess the quality of these automatically generated masks by measuring their Dice similarity score relative to the expert annotations. We report two examples, for simple and complex morphologies. The expert

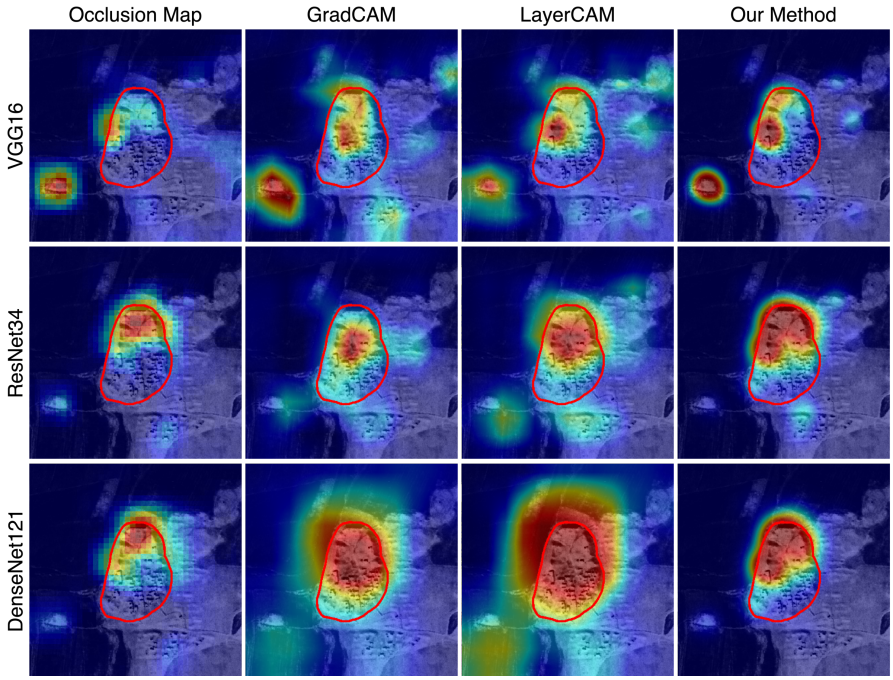


Figure 5.12: A feature of interest with a mound morphology (36.652° N, 40.270° E). Only the north end is clear of built environment, coinciding with most of the activations. The mound is surrounded by now dried out catchment of a branch of the Euphrates River system. This soil appears very dark on CORONA imagery. The red line corresponds to the expert annotation.

annotation (red polygon) and the results of activation mapping processed by CRF (blue polygon) are overlaid on CORONA imagery.

For a simple conical morphology, but with a more elongated extension, the network architectures variably estimate the site boundaries. We notice that the boundary generated with our method shows the highest overlap with the human annotation for all three networks, but especially for VGG and DenseNet (Figure 5.13).

The example is more telling when we study a more complex morphology and background (Figure 5.14). Adding to the complexity is how the site is represented on CORONA imagery. Image boundaries cut some parts of the site as it does not fit into the predetermined image chunk. Jakoby [72] discusses if Tell Mosti with a 'cup-and-saucer' shape exhibits morphological characteristics of a Kranzhügel type [135]. To bypass the site representation problem, the human annotation only considered the 'cup' as the 'site'. Once again, our method is able to determine the extents of the

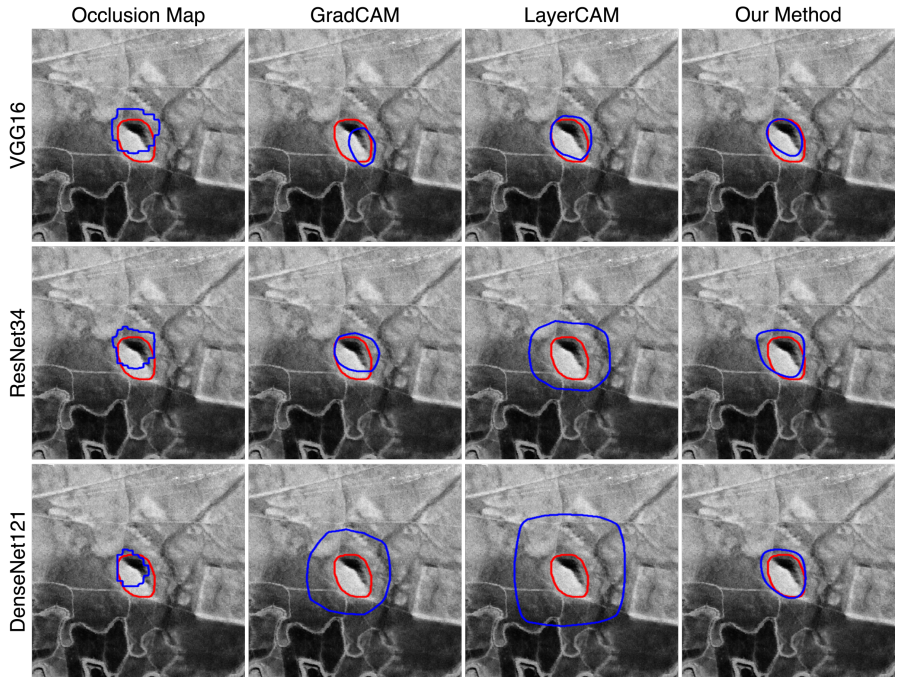


Figure 5.13: A conical mound (36.639° N, 40.977° E) and its model predictions. Note how they all miss the rectangular site just next to the mounded settlement. The red line corresponds to the expert annotation, while the blue line is the predicted boundary.

site, but only for DenseNet and ResNet. These networks are clearly archaeology agnostic, but still conforming with the visible boundaries of the 'cup'. We discuss the image-cutting site boundaries in the next section as we evaluate biases in the training dataset.

Network	Occlusion Maps	GradCAM	LayerCAM	Our Method
VGG16	0.4483	0.4083	**0.6043**	0.5928
ResNet34	0.4826	0.5149	0.4041	**0.5954**
DenseNet121	0.4514	0.3537	0.213	**0.6185**
Mean Values	0.46	0.43	0.41	**0.60**
Variances	0.0004	0.0067	0.0383	**0.0002**

Table 5.2: Dice similarity between the predicted and annotated site area over the entire data set. The bold values represent the highest score achieved per network.

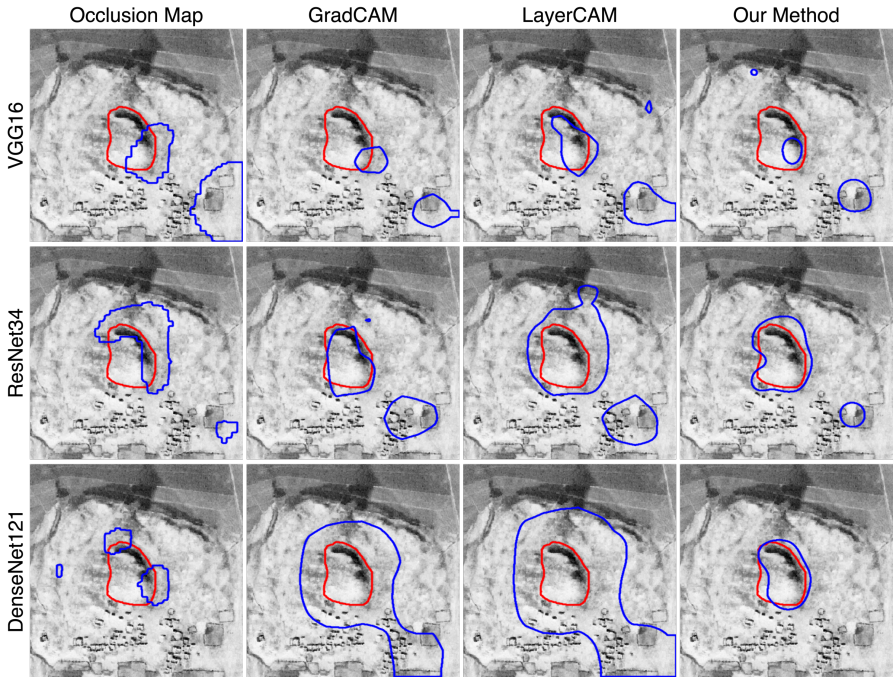


Figure 5.14: Tell Mosti (36.624° N, 41.615° E) exhibits a more complex case. Please note the actual site is larger than the digitized/annotated "crown". The image extent cuts the site due to its size. The red line corresponds to the expert annotation, while the blue line is the predicted boundary.

Finally, we provide a quantitative analysis to show an overview of the quality of the generated masks over the entire data set. We report the Dice similarity scores between the predicted boundary and the annotation in Table 5.2. We observe that although all three networks perform similarly for the classification task, their ability to delineate the boundary of an archaeological site differs. The variations in performance possibly stem from architectural differences between networks, as well as from the type of explainability method we employed. One notable exception is given by our method, which shows the least amount of variation between Dice scores across networks. Also, the masks generated from processing the activations of our method produce the highest individual score for DenseNet and better average performance across all architectures than the masks generated from the other explainability techniques. Thus, our extension to the Guided GradCAM appears to be a robust annotation generator for the given training data set collected from this particular geography.

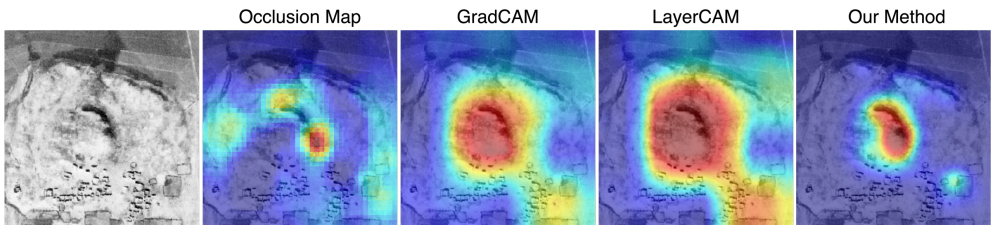


Figure 5.15: Different activation maps for DenseNet. Please note how our extended Guided GradCAM method identifies some parts of the cup of the site matching the human annotation. The slightly activated area in the right left is much more pronounced in GradCAM and LayerCAM. The red line corresponds to the expert annotation.

5.6 Discussion

Here, we present the activation maps of multiple sites with varying morphologies, and we discuss the potential of the activations as sources for cheap annotation. We also derive interpretations of these activations to understand how different CNN architectures learned to distinguish archaeological sites.

We start our discussion using the previous example from Tell Mosti. As we discussed above, the human annotation only included the 'cup' of the site, so there is a clear mismatch between human annotation and model estimation for the most part. It is only that our proposed addition to the Guided GradCAM estimates an area close to human interpretation, but GradCAM and LayerCAM reveal a high-activation area in the lower-right corner of the image (Figure 5.15). To investigate, we explored a high-resolution digital elevation model of Tell Mosti (Figure 5.16). Overall, higher elevations roughly overlap with the results of activation maps. In this particular case, we observe the benefit of analysing the activation maps since they indicated the south-eastern extension, which the expert missed since it is not immediately visible on the CORONA imagery.

To showcase how activation maps may relate to non-circular site morphologies, we selected Tell Jamilo (Figure 5.17a) and Tell Hadi (Figure 5.17b) which present comparable morphologies. The orientations differ but their tangled morphologies are similar. The activation maps of both sites show that VGG predictions are more strongly triggered by round features, a characteristic that many mounds within our data set share. We observe a similar pattern for VGG in the activation map of Figure 5.12. This reliance on round features can be due to the loss of context information after each pooling operation is performed from one group of layers to the next. On the other side, ResNet and DenseNet still retain context information even in the deeper layers

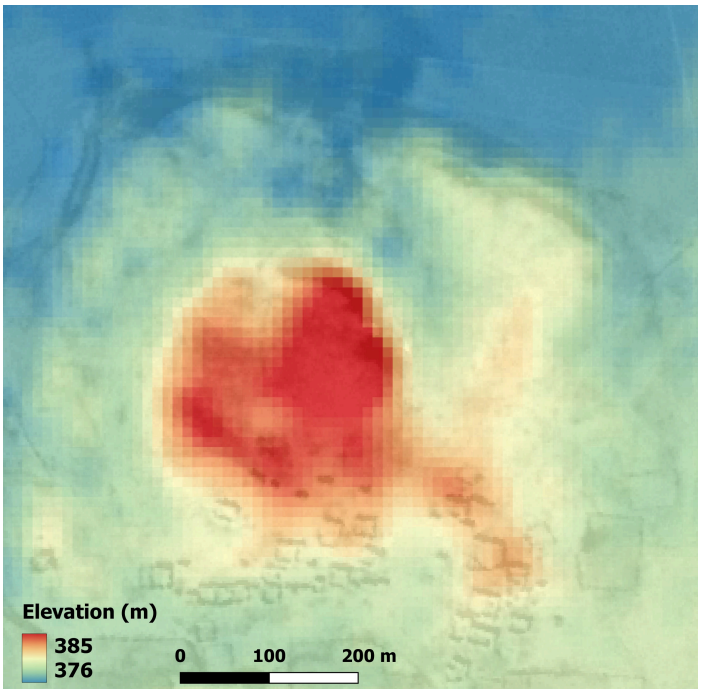


Figure 5.16: The digital elevation model of Tell Mosti highlights the elevated core of the site. Manual annotation considers only the highest eastern blob but misses the entire core.

by using skip and, respectively, dense connections in their architectures, making their activations a promising source for generating segmentation masks.

Furthermore, we explore single conical and complex morphologies in the same image chunk. Figure 5.18 contains two sites in the same image frame, the larger more complex one being identified as Tell al-Shur [128]. Because the sites are close to each other, small CORONA image chunks covered both. We initially identified them as different sites, so the script created one image case for each site with greatly overlapping backgrounds.

In the first instance (Figure 5.18a), the larger site with more complex morphology is at the centre and the smaller conical site is slightly to the right. We observe that DenseNet is able to highlight both sites with GradCAM and LayerCAM, matching the human interpretation. Also, our method is particularly convincing as an annotation source since it activates both sites at the same time with relatively good coverage of the archaeological features without including much of the surrounding landscape. The same couple produces different activations when the central focus is shifted (Fig-

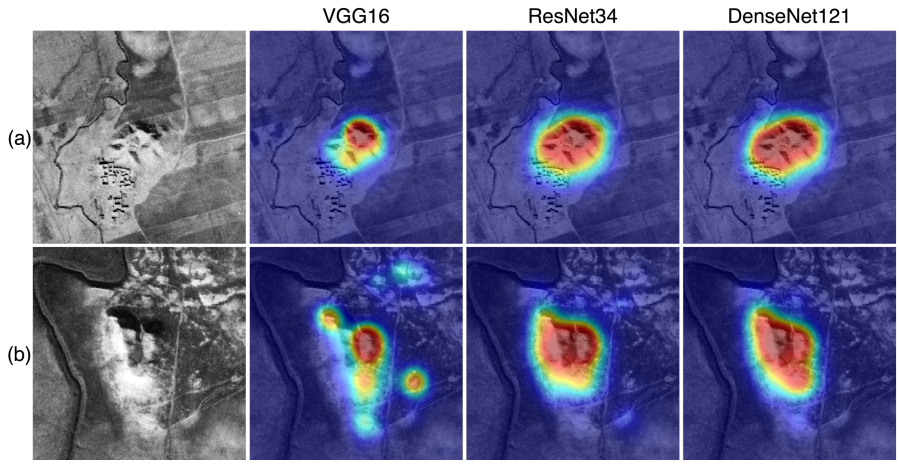


Figure 5.17: Activations generated with our method for Tell Jamilo (36.683° N,40.607° E) (a) and Tell Hadi. (36.870° N,41.865° E) (b).

ure 5.18b). As in the previous case, the circular site is represented almost in entirety. However, it appears as Tell al-Shur lost significance in the activations. It is possible that all three networks learned that a mound should be circular, and our swirling augmentation further emphasized circularity. It may be also possible that models were influenced from the location of Tell al-Shur within the image. In this case, the site is located at the edge of the image, suggesting that the contribution of a site to the prediction in a multi-site image is dependent on the site's position within that image. This is expected since the networks are trained for classification, which does not incentivize the activation of all archaeological features present in an image, but rather of the strongest visual cue, which, in this case, is the circular small mound. When both sites are fully included in the image but shifted upwards from the centre (Figure 5.18c), we notice similar activation patterns as in Figure 5.18a. This mainly shows the invariance to image shifts, a general characteristic of CNNs due to their usage of pooling layers. The difference from the activations in Figure 18b shows that this invariance still requires the relevant image features to be entirely present in the image.

Throughout our analysis of activation maps, we noticed that LayerCAM, through its aggregation of activations from multiple layers, produces the widest coverage of the archaeological features, which proves especially useful when multiple sites are present in an image. GradCAM, by focusing only on the final layer, trims this wider context which results in more focused activations, but at the cost of ignoring some

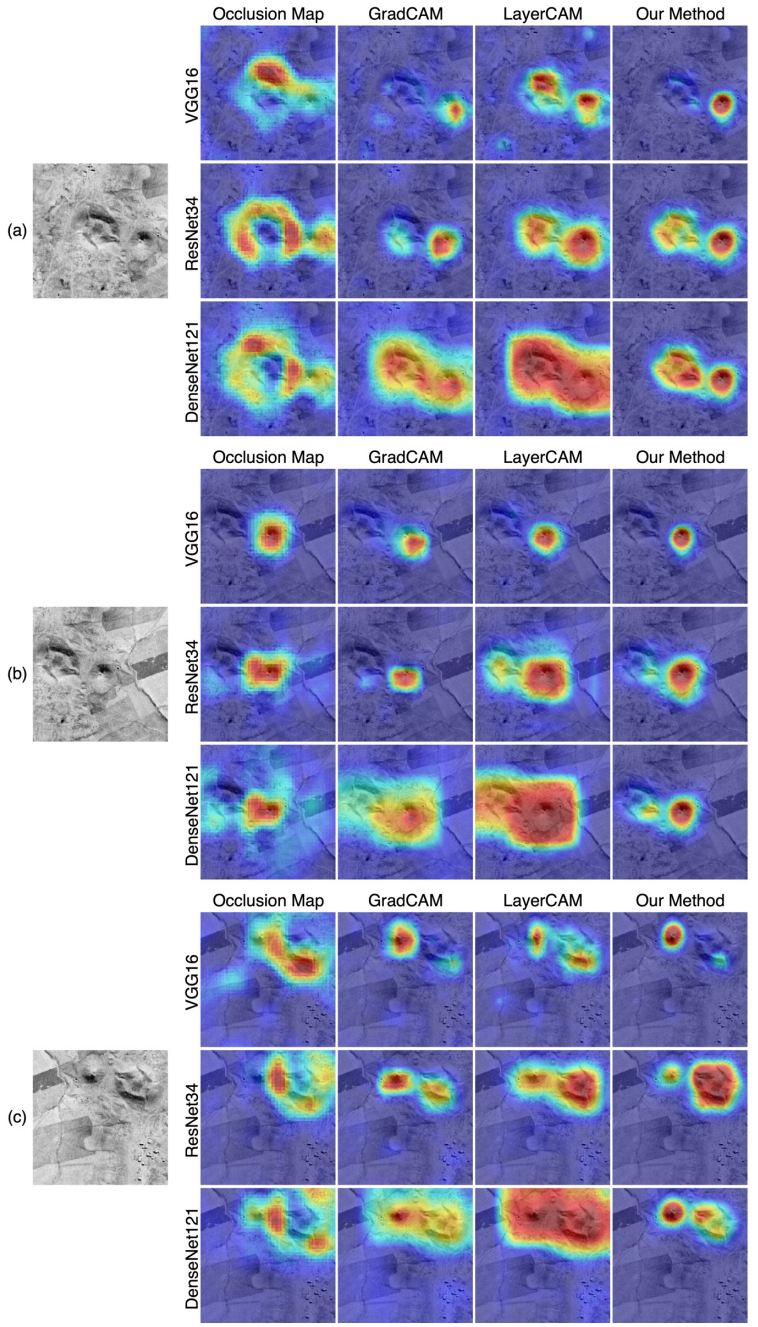


Figure 5.18: Different activations of Tell al-Shur (36.845° N, 40.458° E) with a complex morphology and a nearby conical site. Different augmentations of the same region are activated in different ways.

archaeological features, as is the case for ResNet in Figure 5.18. The Occlusion Maps mainly highlight differences in elevation, by signalling shaded areas as discriminatory features. This technique, however, suffers from a high computational cost, as it requires a new network prediction for every position of the occlusion window, its output is of low resolution, and it also requires choosing a window size, since small occlusion windows may not significantly change the prediction score. Our method generally produces activation maps that include even less of the surrounding landscape than GradCAM which enable a clearer observation of the relevant archaeological features within the images. Moreover, given their sharp boundaries, the activation maps generated with our method can become a strong basis for producing annotations for segmentation as also indicated by the results in Table 5.2.

When it comes to the network architectures, we notice that VGG seems to rely less on context, i.e., image features characterizing the whole site, focusing more on general features, such as roundness. On the other hand, ResNet and DenseNet appear to base their predictions on increasingly more contextual information due to their connections that forward information from previous layers to the following ones, while VGG lacks this characteristic. This wider coverage of the archaeological site by ResNet and DenseNet activations can mean that these two architectures may show better adaptability than VGG to changes in site morphology when, for instance, the geographical area changes.

5.7 Conclusion

Exploring how different networks are activated for mounded settlements proved to be a fruitful exercise. The study generated voluminous data, and we followed a particular path in interpreting experiment results. Therefore, the topic is open, and many other inferences can be made. Our aim has not been to develop a "best-practice guide" with detailed accuracy statistics and thresholds. Inferences we made in this chapter depended upon our CORONA-specific training dataset with a specific site morphology.

The results we reported here are not benchmarks for any network or an activation method. The settlement mound has a particular morphology uniquely contextualized in Upper Mesopotamia. Therefore, our interpretations are specific to the training dataset, and we try to avoid making broad statements. However, experimenting with network architectures using different activation techniques appears to be a fruitful exercise and the workflow may be generalizable.

Our work, while only emphasizing coarse associations between settlement mor-

phology and periodization, opens the door to more detailed and systematic analyses through the application of deep learning. The widespread presence of mounds suggests an opportunity to extend computer-assisted morphological analysis, with our study serving as a step in that direction. Additionally, our approach finds utility in a detection mechanism, where users can observe highlighted regions as potential archaeological sites within large geographical areas.

Furthermore, we showcase the potential of using activation maps as the basis for producing cheap annotations, which, with the incorporation of corrections, either through user intervention in an active learning setup [123] or automatic adjustments [159], can contribute to refining predictions and improving the overall accuracy of algorithms for site delineations. For this particular region, we observed that DenseNet in conjunction with our modified version of Guided GradCAM produces the most accurate site annotations. Moreover, DenseNet's usage of wide contextual information may indicate good robustness to potential changes in the site's morphology and in its surrounding landscape.

Finally, despite our initial focus on settlement mounds and exclusion of periodization concerns in our preliminary experiments, our method holds promise as a potential deep learning-based expert helper system for assisting desktop surveys. Additionally, the integration of Digital Elevation Models (DEM) into the workflow could amplify our method's potential for morphological classification, presenting a versatile tool for archaeological analyses. Also, to better assess the applicability of this study, we aim to expand it by including a more diverse set of site morphologies.

Chapter 6

Conclusion and Outlook

The main goal of the research presented in this thesis was to develop solutions that facilitate applying deep learning algorithms under annotation constraints, with applications for cell imaging and archaeological remote sensing. Throughout this research, we targeted the challenges present for the human operator in both the annotation and the training processes within the deep learning pipeline. To do so, we relaxed the quality requirements placed on the expert annotators, we proposed annotation-efficient learning paradigms, and we introduced explainability case studies. Here, we present a summary of the main contributions of this thesis, acknowledge its limitations, and propose future research directions.

6.1 Contributions and Limitations

In Chapter 2 we modelled three types of inconsistencies that can occur when creating annotations for cell segmentation. These inconsistencies can be considered both annotator-related errors or deliberate relaxations of the annotation process to allow for creating larger quantities of annotations within a fixed time budget. We considered the effect of the omission of a certain proportion of the target cells, the inclusion of objects other than the target cells, and the effect of inconsistent cell boundaries under the form of exaggerated or reduced boundary delineations (called bias). We performed gradual reductions in the annotation quality and we tested their effect on the training of three architecturally-dissimilar segmentation networks. Our results indicated that the networks were least affected by omissions, with inclusion and bias producing more severe degradation of the performance, especially when the cells have small foot-

prints. These findings may allow human operators to optimize quality control efforts by focusing on the most impactful error types, thereby enhancing the robustness of models even when annotation resources are constrained. This contribution directly addresses one of the annotation process bottlenecks by enabling more strategic quality trade-offs, thus supporting the deployment of robust segmentation models with fewer high-quality annotated samples. One inherent limitation of our study is the scope in which it was performed. We considered three CNN architectures, three data sets, and three types of annotation errors, which can limit the broadness of our conclusions.

Based on the findings from Chapter 2, we proposed in Chapter 3 a method that can enhance the quality of annotations suffering from various types of inconsistencies. Our main contribution is designing a learning pipeline for cell segmentation in which a small data set with high-quality annotations is leveraged to train a CNN to upgrade a low-quality annotation to a high-quality one. We achieve this by perturbing the high-quality annotations and tasking the CNN with retrieving their initial quality. We then use this upgrade CNN to enhance the quality of a larger set with low-quality annotations. We showed that by combining the initial small high-quality set with the larger set with upgraded annotations, we can train better-performing cell segmentation CNNs than on the high-quality set alone. This approach presents a practical solution for scaling annotated data in a cost-effective manner by reducing the need for extensive expert annotations. By enhancing lower-quality annotations through an automated upgrading process, our method increases the usable dataset size without a proportional increase in human effort, contributing to the applicability of deep learning in scientific imaging under constrained annotation budgets. This contribution aligns with the thesis's aim to develop annotation-efficient solutions that support the training of deep learning networks in data-limited domains. The main limitation of our approach is the necessity of engineering perturbations that replicate the errors expected in the low-quality data set. Although we showed that the match between perturbations and errors does not have to be perfect, designing them is still an additional cost in the development of the model.

The main contribution of Chapter 4 is creating a few-shot technique particularly suited for cell segmentation. This method fits into the general few-shot learning pipeline while accounting for requirements specific to cell segmentation. We leverage the high-resolution feature maps produced by MSD networks trained on the known cell classes, which we then linearly recombine to adapt to the new class of cells. We demonstrated that the few-shot learning paradigm can be effectively applied to cell images, with our method surpassing other techniques designed for natural images or

medical image segmentation. Here, we targeted the training process of the deep learning pipeline and proposed a solution that reduces the reliance of the human operator on large quantities of annotations. The few-shot learning technique we developed reduced the number of labelled samples required for training new classes in the context of cell segmentation, enabling future experimentation with data sets previously unsuitable for learning setups. One important limitation of our method is its one-shot performance. When trained with a single image, the results vary based on how representative that image is for the rest of the data set. This variation, however, becomes significantly lower when using 5 or 10 shots. Another limitation is the requirement to have sufficient annotations for some of the cell types within a given data set in order to train the MSD networks. This limits the applicability of our work to scenarios in which one wants to segment new structures from an already annotated data set.

Chapter 5 contributes with an annotation process for archaeological site segmentation starting from image-level annotations. We use binary annotations for image classification, i.e., whether a site is present within an image, to train classification models from which we employ explainability techniques to extract activation maps that we further process to obtain site boundaries. In addition to producing cheap annotations for segmentation, we leverage the resulting maps to perform an analysis of the learned features by three CNNs, which can contribute to a better understanding of the CNNs operation and, consequently, to the wider adoption of these techniques in archaeological works. Moreover, we present a modification to an existing explainability technique which produces site boundaries close to the expert estimation. We observed differences in the image features that different architectures tend to highlight and we also showcased the explainability techniques' potential of highlighting biases or mis-annotated images. This contribution tackles problems in both the annotation and training processes and offers a dual benefit by providing a low-cost method for segmentation annotation in archaeology while simultaneously enhancing model interpretability. On one hand, alleviating the annotation scarcity facilitates the practical introduction of deep learning in archaeological workflows as the human operator does not need to focus much on producing annotations. On the other hand, the capacity to derive meaningful visual explanations from CNNs facilitates a greater understanding of model behaviour, which can build trust in the predictions of deep learning networks. This is important for interdisciplinary applications with non-technical fields where model transparency and interpretability are valued. The geographical area to which the study was applied constitutes the main limitation of this chapter. We did not apply our techniques to images belonging to other regions which could contain

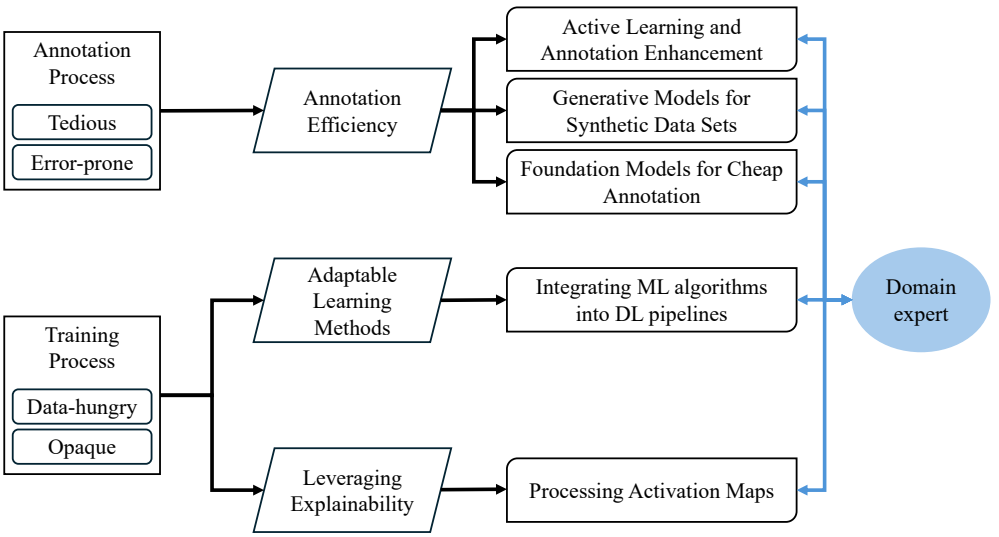


Figure 6.1: A schematic representation of future research directions emerging from this thesis, including enhanced annotation tools, adaptable learning methods and a focus on explainability techniques, highlighting the necessity of continuous collaboration with domain experts.

differently-looking sites or in which the landscape could pose more challenges in identifying a site image from a non-site one.

6.2 Outlook

Future research presents significant opportunities to streamline the advancement of deep learning within scientific fields where the demands of annotation and training processes pose challenges for human operators. In this section, we outline several potential directions for alleviating the costs associated with these processes. For a schematic representation of the envisioned directions, see Figure 6.1.

On the annotation side, one possibility emerges from combining the low-data requirements and short training time of few-shot learning with the enhancement of an upgradinig CNN into efficient annotation tools. Such tools can rely on an initial small set of manual annotations to train a few-shot model whose predictions can be further refined by an upgrading CNN. In this way, the expert can focus only on the most challenging samples, while the networks would also improve as more images are being annotated, in an active learning manner [107].

Alternatively, rather than focusing solely on increasing the throughput of the an-

notation process, this process may be accelerated by generating extensive synthetic segmentation datasets via generative networks, such as diffusion models, where input conditions (e.g., text queries, class labels) can be provided with significantly reduced human intervention compared to the generation of pixel-level segmentation annotations. Additionally, low-effort human input such as textual prompting, point, or box annotations can also be used to increase the number of available segmentation annotations by leveraging the predictions of foundation models, such as segment anything model (SAM)[83] or its more specialized variants, for instance, MedSAM [98].

On the training side, there remains a critical need for adaptable learning methods that can effectively exploit general image features derived from data sets with abundant annotations or even pseudo-annotations as in [112]. The key challenge lies in refining these generalized feature extractors to suit the specific requirements of the target domain. A viable solution involves embedding traditional machine learning algorithms within deep learning pipelines. This integration combines the feature extraction strength of deep learning with the efficiency and reduced data dependency of traditional machine learning methods. As demonstrated in Chapter 4, this hybrid approach has the potential to yield highly adaptable models, addressing the limitations posed by data scarcity in domain-specific applications.

When it comes to the detection of archaeological sites, the output of the explainability techniques (activation maps) can also be leveraged to derive more information about the sites than their boundaries. For example, by analysing their shape and distribution, activation maps can provide information about the morphology of archaeological sites without additional human input. This can then help in further clustering and categorization efforts.

Finally, one common theme that ties together the observations presented in this thesis is the need to strengthen collaboration between machine learning scientists and domain experts. Although scientific domains suffer from expensive data acquisition and annotation processes, these disadvantages can be mitigated by including expert knowledge directly into the development process of learning-based solutions. One way to do so is by introducing constraints based on prior knowledge. For example, in Chapter 5 we applied a Gaussian filter on site activations to generate accurate segmentation masks because we had the a priori knowledge that the area of interest contained round settlements. Thus, we were able to process the activations to better reflect this characteristic without the need of additional data or annotations. Similar approaches could also be applied in designing efficient annotation tools and accurate adaptable learning methods.

Bibliography

[1] Natalia Abellán, Blanca Jiménez-García, José Aznarte, Enrique Baquedano, and Manuel Domínguez-Rodrigo. Deep learning classification of tooth scores made by different carnivores: Achieving high accuracy when comparing African carnivore taxa and testing the hominin shift in the balance of power. Archaeological and Anthropological Sciences, 13(2):1–14, 2021.

[2] Abien Fred Agarap. Deep learning using rectified linear units (ReLU), 2019. arXiv:1803.08375.

[3] Hina Ajmal, Saad Rehman, Umar Farooq, Qurrat U. Ain, Farhan Riaz, and Ali Hassan. Convolutional neural network based image segmentation: A review. Pattern Recognition and Tracking, 10649:191–203, 2018.

[4] Saad Ullah Akram, Juho Kannala, Lauri Eklund, and Janne Heikkilä. Cell segmentation proposal network for microscopy image analysis. In Deep Learning and Data Labeling for Medical Applications, volume 10008 of Lecture Notes in Computer Science, pages 21–29. Springer, 2016.

[5] Yousef Al-Kofahi, Wiem Lassoued, William Lee, and Badrinath Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. IEEE Transactions on Biomedical Engineering, 57(4):841–852, 2009.

[6] Conrad M. Albrecht, Chris Fisher, Marcus Freitag, Hendrik F. Hamann, Sharathchandra Pankanti, Florencia Pezzutti, and Francesca Rossi. Learning and recognizing archeological features from LiDAR data. In IEEE International Conference on Big Data, pages 5630–5636. IEEE, 2019.

[7] Dimitris Alexakis, Theodoros Astaras, Apostolos Sarris, Kostas Vouzaxakis, and Lia Karimali. Reconstructing the neolithic landscape of Thessaly through a GIS and geological approach. In International Conference on Computer Applications and Quantitative Methods in Archaeology, volume 10, pages 405–410. Bonn: Dr. Rudolf Habelt GmbH, 2008.

[8] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1):53, 2021.

[9] Maruthamuthu Angulakshmi and Gnanapandithan G. Lakshmi Priya. Automated brain tumour segmentation techniques— A review. International Journal of Imaging Systems and Technology, 27(1):66–77, 2017.

[10] Ricardo J. Araújo, Jaime S. Cardoso, and Hélder P. Oliveira. A deep learning design for improving topology coherence in blood vessel segmentation. In Medical Image Computing and Computer Assisted Intervention, volume 11764 of Lecture Notes in Computer Science, pages 93–101. Springer, 2019.

[11] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12):2481–2495, 2017.

[12] Roberto Battiti. First-and second-order methods for learning: Between steepest descent and Newton's method. Neural Computation, 4(2):141–166, 1992.

[13] Mehdi Benchoufi, Eric Matzner-Lober, Nicolas Molinari, Anne-Sophie Jannot, and Philippe Soyer. Interobserver agreement issues in radiology. Diagnostic and Interventional Imaging, 101(10):639–641, 2020.

[14] Robert Bensch and Olaf Ronneberger. Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. In IEEE International Symposium on Biomedical Imaging, pages 1220–1223. IEEE, 2015.

[15] Iban Berganzo-Besga, Hector A. Orengo, Felipe Lumbreras, Miguel Carrero-Pazos, João Fonte, and Benito Vilas-Estévez. Hybrid MSRM-based deep learning and multitemporal sentinel 2-based machine learning algorithm detects near 10k archaeological tumuli in North-Western Iberia. Remote Sensing, 13(20):4181, 2021.

[16] Chandradeep Bhatt, Indrajeet Kumar, V. Vijayakumar, Kamred Udham Singh, and Abhishek Kumar. The state of the art of deep learning models in medical science and their challenges. Multimedia Systems, 27(4):599–613, 2021.

[17] Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, and Gabriel Chartrand. The liver tumor segmentation benchmark (LiTS). Medical Image Analysis, 84:102680, 2023.

[18] Alexander Bonhage, Mahmoud Eltaher, Thomas Raab, Michael Breuß, Alexandra Raab, and Anna Schneider. A modified Mask region-based convolutional neural network approach for the automated detection of archaeological sites on high-resolution light detection and ranging-derived digital elevation models in the North German Lowland. Archaeological Prospection, 28(2):177–186, 2021.

[19] Carolyn Burnett and Thomas Blaschke. A multi-scale segmentation/object relationship modelling methodology for landscape analysis. Ecological Modelling, 168(3):233–249, 2003.

[20] Stefano Campana and Salvatore Piro. Seeing the unseen. Geophysics and landscape archaeology. CRC Press, 2008.

[21] Yigit B. Can, Krishna Chaitanya, Basil Mustafa, Lisa M. Koch, Ender Konukoglu, and Christian F. Baumgartner. Learning to segment medical images with scribble-supervision alone. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, volume 11045 of Lecture Notes in Computer Science, pages 236–244. Springer, 2018.

[22] Aaron Carass, Snehashis Roy, Amod Jog, Jennifer L. Cuzzocreo, Elizabeth Magrath, Adrian Gherman, Julia Button, James Nguyen, Ferran Prados, and Carole H. Sudre. Longitudinal multiple sclerosis lesion segmentation: Resource and challenge. NeuroImage, 148:77–102, 2017.

[23] Jesse Casana. Remote sensing-based approaches to site morphology and historical geography in the northern fertile crescent. New Agendas in Remote Sensing and Landscape Archaeology in the Near East. Studies in honour of Tony J. Wilkinson: Archaeopress, pages 154–174, 2020.

[24] Jesse Casana, Jackson Cothren, and Tuna Kalayci. Swords into ploughshares: Archaeological applications of CORONA satellite imagery in the Near East. Internet Archaeology, 32(2), 2012.

[25] Gino Caspari and Pablo Crespo. Convolutional neural networks for archaeological site detection–Finding "princely" tombs. Journal of Archaeological Science, 110:104998, 2019.

[26] University of Arkansas/U.S. Geological Survey Center for Advanced Spatial Technologies. Corona @ CAST UA.

[27] Sixian Chan, Cheng Huang, Cong Bai, Weilong Ding, and Shengyong Chen. Res2-UNeXt: A novel deep learning framework for few-shot cell image segmentation. Multimedia Tools and Applications, 81(10):13275–13288, 2022.

[28] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4):834–848, 2017.

[29] Shuai Chen, Antonio Garcia-Uceda, Jiahang Su, Gijs van Tulder, Lennard Wolff, Theo van Walsum, and Marleen de Bruijne. Label refinement network from synthetic error augmentation for medical image segmentation. Medical Image Analysis, 99:103355, 2025.

[30] Guy Barrett Coleman and Harry C. Andrews. Image segmentation by clustering. Proceedings of the IEEE, 67(5):773–785, 1979.

[31] Frederick A. Cooper, Marvin E. Bauer, and Brenda C. Cullen. Satellite spectral data and archaeological reconnaissance in western Greece. NASA. Stennis Space Center, Applications of Space-Age Technology in Anthropology, 1991.

[32] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. IEEE Signal Processing Magazine, 35(1):53–65, 2018.

[33] Jay F. Custer, Timothy Eveleigh, Vytautas Klemas, and Ian Wells. Application of LANDSAT data and synoptic remote sensing to predictive models for prehistoric archaeological sites: An example from the Delaware coastal plain. American Antiquity, 51(3):572–588, 1986.

[34] Wei Dai, Nanqing Dong, Zeya Wang, Xiaodan Liang, Hao Zhang, and Eric P. Xing. SCAN: Structure correcting adversarial network for organ segmentation in chest X-rays. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, volume 11045 of Lecture Notes in Computer Science, pages 263–273. Springer, 2018.

[35] Yousef-Awwad Daraghmi, Tsung-Hsiang Wu, and Tsì-Uí İk. Crowdsourcing-based road surface evaluation and indexing. IEEE Transactions on Intelligent Transportation Systems, 23(5):4164–4175, 2020.

[36] Anurag Das, Yongqin Xian, Yang He, Zeynep Akata, and Bernt Schiele. Urban scene semantic segmentation with low-cost coarse annotation. In IEEE/CVF Winter Conference on Applications of Computer Vision, pages 5978–5987, 2023.

[37] Dylan S. Davis. Object-based image analysis: A review of developments and future directions of automated feature detection in landscape archaeology. Archaeological Prospection, 26(2):155–163, 2019.

[38] Dylan S. Davis, Gino Caspari, Carl P. Lipo, and Matthew C. Sanger. Deep learning reveals extent of Archaic Native American shell-ring building practices. Journal of Archaeological Science, 132:105433, 2021.

[39] Youssef Dawoud, Katharina Ernst, Gustavo Carneiro, and Vasileios Belagiannis. Edge-based self-supervision for semi-supervised few-shot microscopy image cell segmentation. In Medical Optical Imaging and Virtual Microscopy Image Analysis, volume 13578 of Lecture Notes in Computer Science, pages 22–31. Springer, 2022.

[40] Véronique De Laet, Etienne Paulissen, and Marc Waelkens. Methods for the extraction of archaeological features from very high-resolution Ikonos-2 remote sensing imagery, Hisar (southwest Turkey). Journal of Archaeological Science, 34(5):830–841, 2007.

[41] Katleen Deckers and Simone Riehl. Fluvial environmental contexts for archaeological sites in the Upper Khabur basin (Northeastern Syria). Quaternary Research, 67(3):337–348, 2007.

[42] Katleen Deckers and Simone Riehl. Resource exploitation of the Upper Khabur Basin (NE Syria) during the 3rd millennium BC. Paléorient, 34(2):173–189, 2008.

[43] Jean-Paul Demoule and Catherine Perlès. The Greek Neolithic: A new review. Journal of World Prehistory, 7(4):355–416, 1993.

[44] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

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

[46] Kurt Diem, Amalia Magaret, Alexis Klock, Lei Jin, Jia Zhu, and Lawrence Corey. Image analysis for accurately counting CD4+ and CD8+ T cells in human tissue. Journal of Virological Methods, 222:117–121, 2015.

[47] Zhipeng Ding, Xu Han, and Marc Niethammer. VoteNet: A deep learning label fusion method for multi-atlas segmentation. In Medical Image Computing and Computer Assisted Intervention, volume 11766, pages 202–210. Springer, 2019.

[48] Manuel Domínguez-Rodrigo, Gabriel Cifuentes-Alcobendas, Blanca Jiménez-García, Natalia Abellán, Marcos Pizarro-Monzo, Elia Organista, and Enrique Baquedano. Artificial intelligence provides greater accuracy in the classification of modern and ancient bone surface modifications. Nature Scientific Reports, 10(1):18862, 2020.

[49] Venkata Gopal Edupuganti, Akshay Chawla, and Amit Kale. Automatic optic disk and cup segmentation of fundus images using deep learning. In IEEE International Conference on Image Processing, pages 2227–2231, 2018.

[50] Andreas Ess, Tobias Mueller, Helmut Grabner, and Luc Van Gool. Segmentation-based urban traffic scene understanding. In Procedings of the British Machine Vision Conference 2009, pages 84.1–84.11. British Machine Vision Association, 2009.

[51] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639):115–118, 2017.

[52] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, Alexander Dovzhenko, Olaf Tietz, Cristina Dal Bosco, Sean Walsh, Deniz Saltukoglu, Tuan Leng Tay, Marco Prinz, Klaus Palme, Matias Simons, Ilka Diester, Thomas Brox, and Olaf Ronneberger. U-Net: Deep learning for cell counting, detection, and morphometry. Nature Methods, 16(1):67–70, 2019.

[53] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 9(61):1871–1874, 2008.

[54] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision, 59(2):167–181, 2004.

[55] Ruiwei Feng, Xiangshang Zheng, Tianxiang Gao, Jintai Chen, Wenzhe Wang, Danny Z. Chen, and Jian Wu. Interactive few-shot learning: Limited supervision, better medical image segmentation. IEEE Transactions on Medical Imaging, 40(10):2575–2588, 2021.

[56] Abdur R. Feyjie, Reza Azad, Marco Pedersoli, Claude Kauffman, Ismail Ben Ayed, and Jose Dolz. Semi-supervised few-shot learning for medical image segmentation, 2020. arXiv:2003.08462.

[57] Ruth C. Fong and Andrea Vedaldi. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In IEEE International Conference on Computer Vision), pages 3449–3457. IEEE, 2017.

[58] Martin J. Fowler. The archaeological potential of declassified HEXAGON KH-9 panoramic camera satellite photographs. AARG News, 53(210):30–36, 2016.

[59] Simon Graham, Mostafa Jahanifar, Ayesha Azam, Mohammed Nimir, Yee-Wah Tsang, Katherine Dodd, Emily Hero, Harvir Sahota, Atisha Tank, Ksenija Benes, et al. Lizard: A large-scale dataset for colonic nuclear instance segmentation and classification. In IEEE/CVF International Conference on Computer Vision, pages 684–693, 2021.

[60] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. Medical Image Analysis, 58:101563, 2019.

[61] Noah F. Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J. McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, Cole Pavelchek, Sunny Cui, Isabella Camplisson, Omer Bar-Tal, Jaiveer Singh, Mara Fong, Gautam Chaudhry, Zion Abraham, Jackson Moseley, Shiri Warshawsky, Erin Soon, Shirley Greenbaum, Tyler Risom, Travis Hollmann, Sean C. Bendall, Leeat Keren, William Graf, Michael Angelo, and David Van Valen. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. Nature Biotechnology, 40(4):555–565, 2022.

[62] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S. Lew. A review of semantic segmentation using deep neural networks. International Journal of Multimedia Information Retrieval, 7(2):87–93, 2018.

[63] Alexandre Guyot, Marc Lennon, Thierry Lorho, and Laurence Hubert-Moy. Combined detection and segmentation of archeological structures from LiDAR data using a deep learning approach. Journal of Computer Applications in Archaeology, 4(1):x, 2021.

[64] Emily Hammer, Mackinley FitzPatrick, and Jason A. Ur. Succeeding CORONA: Declassified HEXAGON intelligence imagery for archaeological and historical research. Antiquity, 96(387):679–695, 2022.

[65] Emily Hammer and Jason A. Ur. Near Eastern landscapes and declassified U2 aerial imagery. Advances in Archaeological Practice, 7(2):107–126, 2019.

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[67] Nicholas Heller, Joshua Dean, and Nikolaos Papanikolopoulos. Imperfect segmentation labels: How much do they matter? In Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, pages 112–120. Springer, 2018.

[68] Théophraste Henry, Alexandre Carré, Marvin Lerousseau, Théo Estienne, Charlotte Robert, Nikos Paragios, and Eric Deutsch. Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-Net neural networks: A BraTS 2020 challenge solution. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, pages 327–339. Springer, 2021.

[69] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. arXiv:1503.02531 [stat].

[70] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In Wavelets, pages 286–297. Springer, 1990.

[71] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2261–2269. IEEE, 2017.

[72] Elise Jakoby. Considering Kranzhügeln: An exploration into the structural variation and environmental/spatial distribution of the third millennium "Kranzhügel" sites. PhD thesis, University of Arkansas, Fayetteville, 2013.

[73] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. LayerCAM: Exploring hierarchical class activation maps for localization. IEEE Transactions on Image Processing, 30:5875–5888, 2021.

[74] Licheng Jiao and Jin Zhao. A survey on the new generation of deep learning in image processing. IEEE Access, 7:172231–172263, 2019.

[75] Brian Alan Johnson and Lei Ma. Image segmentation and object-based image analysis for environmental monitoring: Recent areas of interest, researchers' views on the future priorities. Remote Sensing, 12(11):1772, 2020.

[76] Singaraju Jyothi. A survey on threshold based segmentation technique in image processing. International Journal of Innovative Research and Development, 3(12):234–239, 2014.

[77] Uday Kamath, John Liu, and James Whitaker. Deep learning for NLP and speech recognition. Springer, 2019.

[78] Nitin Kanagaraj, David Hicks, Ayush Goyal, Sanju Tiwari, and Ghanapriya Singh. Deep learning using computer vision in self driving cars for lane and traffic sign detection. International Journal of System Assurance Engineering and Management, 12(6):1011–1025, 2021.

[79] Sota Kato and Kazuhiro Hotta. One-shot and partially-supervised cell image segmentation using small visual prompt. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 4295–4304, 2023.

[80] A. Emre Kavur, N. Sinem Gezer, Mustafa Barış, Sinem Aslan, Pierre-Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, Savaş Özkan, Bora Baydar, Dmitry Lachinov, Shuo Han, Josef Pauli, Fabian Isensee, Matthias Perkonigg, Rachana Sathish, Ronnie Rajan, Debdoot Sheet, Gurbandurdy Dovletov, Oliver Speck, Andreas Nürnberger, Klaus H. Maier-Hein, Gözde Bozdağı Akar, Gözde Ünal, Oğuz Dicle, and M. Alper Selver. CHAOS challenge - combined (CT-MR) healthy abdominal organ segmentation. Medical Image Analysis, 69:101950, 2021.

[81] Matthew R. Keaton, Ram J. Zaveri, and Gianfranco Doretto. CellTranspose: Few-shot domain adaptation for cellular instance segmentation. In IEEE/CVF Winter Conference on Applications of Computer Vision, pages 455–466, 2023.

[82] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations. Ithaca, 2014.

[83] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In IEEE/CVF International Conference on Computer Vision, pages 3992–4003, 2023.

[84] Andrew B. Knott, Guy Perkins, Robert Schwarzenbacher, and Ella Bossy-Wetzel. Mitochondrial fragmentation in neurodegeneration. Nature Reviews Neuroscience, 9(7):505–518, 2008.

[85] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch, 2020. arXiv:2009.07896 [cs].

[86] Daniel Krentzel, Spencer L. Shorte, and Christophe Zimmer. Deep learning in image-based phenotypic drug discovery. Trends in Cell Biology, 33(7):538–554, 2023.

[87] Melda Küçükdemirci and Apostolos Sarris. Deep learning based automated analysis of archaeo-geophysical images. Archaeological Prospection, 27(2):107–118, 2020.

[88] Karsten Lambers, Wouter Verschoof-van Der Vaart, and Quentin Bourgeois. Integrating remote sensing, machine learning, and citizen science in Dutch archaeological prospection. Remote Sensing, 11(7):794, 2019.

[89] Bennett Landman, Zhoubing Xu, J. Iglesias, Martin Styner, T. Langerak, and Arno Klein. Multi-atlas labeling beyond the cranial vault – workshop and challenge. In Proceedings of MICCAI Workshop Challenge, volume 5, page 12, 2015.

[90] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8047–8057, 2022.

[91] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

[92] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems, 33(12):6999–7019, 2022.

[93] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. A survey of deep neural network architectures and their applications. Neurocomputing, 234:11–26, 2017.

[94] Zhiyan Liu, Andrey Bychkov, Chan Kwon Jung, Mitsuyoshi Hirokawa, Shaofeng Sui, SoonWon Hong, Chiung-Ru Lai, Deepali Jain, Sule Canberk, and Kennichi Kakudo. Interobserver and intraobserver variation in the morphological evaluation of noninvasive follicular thyroid neoplasm with papillary-like nuclear features in Asian practice. Pathology International, 69(4):202–210, 2019.

[95] Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Graham Knott, and Pascal Fua. Supervoxel-based segmentation of mitochondria in EM image stacks with learned shape features. IEEE Transactions on Medical Imaging, 31(2):474–486, 2012.

[96] Carmen Alina Lupascu, Domenico Tegolo, and Emanuele Trucco. FABC: Retinal vessel segmentation using AdaBoost. IEEE Transactions on Information Technology in Biomedicine, 14(5):1267–1274, 2010.

[97] Bertille Lyonnet. Settlement pattern in the Upper Khabur (NE Syria) from the Achaemenids to the Abbasid Period: Methods and preliminary results from a survey. Continuity and Change in Northern Mesopotamia from the Hellenistic to the Early Islamic Period, pages 349–361, 1996.

[98] Jun Ma, Yuting He, Fei-Fei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. Nature Communications, 15(1):654, 2024.

[99] TorchVision maintainers and contributors. TorchVision: PyTorch's computer vision library, November 2016.

[100] Francesca Matrone, Andrea Felicetti, Marina Paolanti, and Roberto Pierdicca. Explaining AI: Understanding deep learning models for heritage point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pages 207–214, 2023.

[101] Wendy Matthews. Tells in Archaeology. In Encyclopedia of Global Archaeology, pages 10553–10556. Springer, 2020.

[102] Damian J. Matuszewski and Ida-Maria Sintorn. Minimal annotation training for segmentation of microscopy images. In IEEE International Symposium on Biomedical Imaging, pages 387–390, 2018.

[103] Bjoern H. Menze and Jason A. Ur. Mapping patterns of long-term settlement in Northern Mesopotamia at a large scale. Proceedings of the National Academy of Sciences, 109(14):E778–E787, 2012.

[104] Bjoern H. Menze, Jason A. Ur, and Andrew G. Sherratt. Detection of ancient settlement mounds. Photogrammetric Engineering & Remote Sensing, 72(3):321–327, 2006.

[105] Shaobo Min, Xuejin Chen, Zheng-Jun Zha, Feng Wu, and Yongdong Zhang. A two-stream mutual attention network for semi-supervised biomedical segmentation with noisy labels. In AAAI Conference on Artificial Intelligence, volume 33, pages 4578–4585, 2019.

[106] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7):3523–3542, 2022.

[107] Robert Munro Monarch. Human-in-the-loop machine learning: Active learning and annotation for human-centered AI. Simon and Schuster, 2021.

[108] John Ashworth Nelder and Robert Wedderburn. Generalized linear models. Journal of the Royal Statistical Society Series A: Statistics in Society, 135(3):370–384, 1972.

[109] Dong Nie, Yaozong Gao, Li Wang, and Dinggang Shen. ASDNet: Attention based semi-supervised deep networks for medical image segmentation. In Medical Image Computing and Computer Assisted Intervention, pages 370–378. Springer, 2018.

[110] Favour Olaoye, Chris Bell, and Peter Broklyn. Crowdsourcing platforms for collaborative analysis of archaeological big data. EasyChair Preprint 14252, EasyChair, 2024.

[111] Nobuyuki Otsu. A threshold selection method from gray-Level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1):62–66, 1979.

[112] Cheng Ouyang, Carlo Biffi, Chen Chen, Turkay Kart, Huaqi Qiu, and Daniel Rueckert. Self-supervised learning for Few-Shot Medical Image segmentation. IEEE Transactions on Medical Imaging, 41(7):1837–1848, 2022.

[113] William A. Parkinson, Attila Gyucha, Panagiotis Karkanas, Nikos Papadopoulos, Georgia Tsartsidou, Apostolos Sarris, Paul R. Duffy, and Richard W. Yerkes. A landscape of tells: Geophysics and microstratigraphy at two Neolithic tell sites on the Great Hungarian Plain. Journal of Archaeological Science: Reports, 19:903–924, 2018.

[114] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In International Conference on Neural Information Processing Systems, pages 8026–8037. Curran Associates Inc., 2019.

[115] Giacomo Patrucco and Francesco Setragno. Multiclass semantic segmentation for digitisation of movable heritage using deep learning techniques. Virtual Archaeology Review, 12(25):85, 2021.

[116] Daniël M. Pelt and James A. Sethian. A mixed-scale dense convolutional neural network for image analysis. Proceedings of the National Academy of Sciences, 115(2):254–259, 2018.

[117] Liying Peng, Lanfen Lin, Hongjie Hu, Yue Zhang, Huali Li, Yutaro Iwamoto, Xian-Hua Han, and Yen-Wei Chen. Semi-supervised learning for semantic segmentation of emphysema with partial annotations. IEEE Journal of Biomedical and Health Informatics, 24(8):2327–2336, 2020.

[118] Catherine Perlès. A case study in Early Neolithic settlement patterns: Eastern Thessaly. In The Early Neolithic in Greece: The First Farming Communities in Europe, Cambridge World Archaeology, pages 121–151. Cambridge University Press, 2001.

[119] Graham Philip, Daniel Donoghue, Anthony Beck, and Nikolaos Galiatsatos. CORONA satellite photography: An archaeological application from the Middle East. Antiquity, 76(291):109–118, 2002.

[120] Tran Minh Quan, David Grant Colburn Hildebrand, and Won-Ki Jeong. FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics. Frontiers in Computer Science, 3, 2021.

[121] Martin Rajchl, Matthew C. H. Lee, Ozan Oktay, Konstantinos Kamnitsas, Jonathan Passerat-Palmbach, Wenjia Bai, Mellisa Damodaram, Mary A. Rutherford, Joseph V. Hajnal, Bernhard Kainz, and Daniel Rueckert. DeepCut: Object segmentation from bounding box annotations using convolutional neural networks. IEEE Transactions on Medical Imaging, 36(2):674–683, 2017.

[122] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A. Efros, and Sergey Levine. Few-Shot segmentation propagation with guided networks, 2018. arXiv:1806.07373 [cs].

[123] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. ACM Computing Surveys, 54(9):180:1–180:40, 2021.

[124] Denise Rey and Markus Neuhäuser. Wilcoxon-signed-rank test. In International Encyclopedia of Statistical Science, pages 1658–1659. Springer, 2011.

[125] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention, pages 234–241. Springer, 2015.

[126] Arlene Miller Rosen. Cities of clay: The geoarcheology of tells. Prehistoric Archeology and Ecology Series. University of Chicago Press, 1986.

[127] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. 'Squeeze & excite' guided few-shot segmentation of volumetric images. Medical Image Analysis, 59:101587, 2020.

[128] Walther Sallaberger and Jason A. Ur. Tell Beydar/Nabada in its regional setting. In Third millennium cuneiform texts from Tell Beydar (seasons 1996-2002), Subartu 12, pages 51–71. Brepols, 2004.

[129] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In IEEE International Conference on Computer Vision, pages 618–626, 2017.

[130] Jean Serra. Image Analysis and Mathematical Morphology. Academic Press, Inc., 1983.

[131] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):640–651, 2017.

[132] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, pages 1–14. Computational and Biological Learning Society, 2015.

[133] Abraham George Smith, Eusun Han, Jens Petersen, Niels Alvin Faircloth Olsen, Christian Giese, Miriam Athmann, Dorte Bodin Dresbøll, and Kristian Thorup-Kristensen. RootPainter: Deep learning segmentation of biological images with corrective annotation. The New Phytologist, 236(2):774–791, 2022.

[134] Andrew Smith. Image segmentation scale parameter optimization and land cover classification using the Random Forest algorithm. Journal of Spatial Science, 55(1):69–79, 2010.

[135] Stefan L. Smith. A morphological typology for the "Kranzhügel" of the Greater Western Jazira and its impact upon interpretations of Early Bronze Age north-eastern Syria. Archaeological Research in Asia, 29:100339, 2022.

[136] Maja Somrak, Sašo Džeroski, and Žiga Kokalj. Learning to classify structures in ALS-derived visualizations of ancient Maya settlements with CNN. Remote Sensing, 12(14):2215, 2020.

[137] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: A generalist algorithm for cellular segmentation. Nature Methods, 18(1):100–106, 2021.

[138] Liyan Sun, Chenxin Li, Xinghao Ding, Yue Huang, Zhong Chen, Guisheng Wang, Yizhou Yu, and John Paisley. Few-shot medical image segmentation using a global correlation network with discriminative embedding. Computers in Biology and Medicine, 140:105067, 2022.

[139] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. Foundations and Trends® in Machine Learning, 4(4):267–373, 2012.

[140] David Svoboda, Michal Kozubek, and Stanislav Stejskal. Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry. Cytometry Part A, 75A(6):494–509, 2009.

[141] Hao Tang, Xingwei Liu, Shanlin Sun, Xiangyi Yan, and Xiaohui Xie. Recurrent mask refinement for few-shot medical image segmentation. In IEEE/CVF International Conference on Computer Vision, pages 3918–3928, 2021.

[142] Martina Tenzer, Giada Pistilli, Alex Bransden, and Alex Shenfield. Debating AI in Archaeology: Applications, implications, and ethical considerations. Internet Archaeology, (67), 2024.

[143] Thanh Tran, Lam Binh Minh, Suk-Hwan Lee, and Ki-Ryong Kwon. Blood cell count using deep learning semantic segmentation, 2019.

[144] Øivind Due Trier, David C. Cowley, and Anders Ueland Waldeland. Using deep neural networks on airborne laser scanning data: Results from a case study of semi-automatic mapping of archaeological topography on Arran, Scotland. Archaeological Prospection, 26(2):165–175, 2019.

[145] Øivind Due Trier, Jarle Hamar Reksten, and Kristian Løseth. Automated mapping of cultural heritage in Norway from airborne lidar data using faster R-CNN. International Journal of Applied Earth Observation and Geoinformation, 95:102241, 2021.

[146] Maurizio Troiano, Eugenio Nobile, Flavia Grignaffini, Fabio Mangini, Marco Mastrogiuseppe, Cecilia Conati Barbaro, and Fabrizio Frezza. A Comparative analysis of machine learning algorithms for identifying cultural and technological groups in archaeological datasets through clustering analysis of homogeneous data. Electronics, 13(14):2752, 2024.

[147] Maurizio Troiano, Eugenio Nobile, Fabio Mangini, Marco Mastrogiuseppe, Cecilia Conati Barbaro, and Fabrizio Frezza. A comparative analysis of the bayesian regularization and Levenberg–Marquardt training algorithms in neural networks for small datasets: A metrics prediction of Neolithic laminar artefacts. Information, 15(5):270, 2024.

[148] Jason A. Ur. Surface collection and offsite studies at Tell Hamoukar, 1999. Iraq, 64:15–43, 2002.

[149] Jason A. Ur. The morphology of Neo-Assyrian cities. Subartu, 6-7:11–22, 2013.

[150] Jason A. Ur and Tony J. Wilkinson. Settlement and economic landscapes of Tell Beydar and its hinterland, pages 305–327. Brepols, 2008.

[151] Ryanne van Dalen, Sultan Mehmood, Paul Verstraten, and Karen van der Wiell. Public funding of science: An international comparison. Technical report, CPB Netherlands Bureau for Economic Policy Analysis, 2014.

[152] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: Image processing in Python. PeerJ, 2:e453, 2014.

[153] Ruchika Verma, Neeraj Kumar, Abhijeet Patil, Nikhil Cherian Kurian, Swapnil Rane, Simon Graham, Quoc Dang Vu, Mieke Zwager, Shan E. Ahmed Raza, Nasir Rajpoot, Xiyi Wu, Huai Chen, Yijie Huang, Lisheng Wang, Hyun Jung, G. Thomas Brown, Yanling Liu, Shuolin Liu, Seyed Alireza Fatemi Jahromi, Ali Asghar Khani, Ehsan Montahaei, Mahdieh Soleymani Baghshah, Hamid Behroozi, Pavel Semkin, Alexandr Rassadin, Prasad Dutande, Romil Lodaya, Ujjwal Baid, Bhakti Baheti, Sanjay Talbar, Amirreza Mahbod, Rupert Ecker, Isabella Ellinger, Zhipeng Luo, Bin Dong, Zhengyu Xu, Yuehan Yao, Shuai Lv, Ming Feng, Kele Xu, Hasib Zunair, Abdessamad Ben Hamza, Steven Smiley, Tang-Kai Yin, Qi-Rui Fang, Shikhar Srivastava, Dwarikanath Mahapatra,

Lubomira Trnavska, Hanyun Zhang, Priya Lakshmi Narayanan, Justin Law, Yinyin Yuan, Abhiroop Tejomay, Aditya Mitkari, Dinesh Koka, Vikas Ramachandra, Lata Kini, and Amit Sethi. MoNuSAC2020: A Multi-organ nuclei segmentation and classification challenge. IEEE Transactions on Medical Imaging, 40(12):3413–3423, 2021.

[154] Wouter B. Verschoof-van der Vaart, Karsten Lambers, Wojtek Kowalczyk, and Quentin P.J. Bourgeois. Combining deep learning and location-based ranking for large-scale archaeological prospection of LiDAR data from the Netherlands. ISPRS International Journal of Geo-Information, 9(5):293, 2020.

[155] Tomas Vicar, Jan Balvan, Josef Jaros, Florian Jug, Radim Kolar, Michal Masarik, and Jaromir Gumulec. Cell segmentation methods for label-free contrast microscopy: Review and comprehensive comparison. BMC Bioinformatics, 20(1):360, 2019.

[156] Eugene Vorontsov and Samuel Kadoury. Label noise in segmentation networks: Mitigation must deal with bias. In Deep Generative Models, and Data Augmentation, Labelling, and Imperfections, volume 1, pages 251–258. Springer, 2021.

[157] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. Computational Intelligence and Neuroscience, 2018(1):7068349:1–7068349:13, 2018.

[158] Șerban Vădineanu, Daniël M. Pelt, Oleh Dzyubachyk, and Kees Joost Batenburg. An analysis of the impact of annotation errors on the accuracy of deep learning for cell segmentation. In International Conference on Medical Imaging with Deep Learning, Proceedings of Machine Learning Research, pages 1251–1267. PMLR, 2022.

[159] Șerban Vădineanu, Daniël M. Pelt, Oleh Dzyubachyk, and Kees Joost Batenburg. Reducing manual annotation costs for cell segmentation by upgrading low-quality annotations. Journal of Imaging, 10(7):172, 2024.

[160] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. PANet: Few-shot image semantic segmentation with prototype alignment. In IEEE/CVF International Conference on Computer Vision, pages 9196–9205, 2019.

[161] Shanshan Wang, Cheng Li, Rongpin Wang, Zaiyi Liu, Meiyun Wang, Hongna Tan, Yaping Wu, Xinfeng Liu, Hui Sun, Rui Yang, Xin Liu, Jie Chen, Huihui Zhou, Ismail Ben Ayed, and Hairong Zheng. Annotation-efficient deep learning for automatic medical image segmentation. Nature Communications, 12(1):5915, 2021.

[162] Xiang-Yang Wang, Ting Wang, and Juan Bu. Color image segmentation using pixel wise support vector machine classification. Pattern Recognition, 44(4):777–787, 2011.

[163] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys, 53(3):63:1–63:34, 2020.

[164] Mohammed A. Wani and Bruce G. Batchelor. Edge-region-based segmentation of range images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(3):314–319, 1994.

[165] Simon K. Warfield, Kelly H. Zou, and William M. Wells. Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation. IEEE Transactions on Medical Imaging, 23(7):903–921, 2004.

[166] David Wiesner, David Svoboda, Martin Maška, and Michal Kozubek. CytoPacq: A web-interface for simulating multi-dimensional cell imaging. Bioinformatics, 35(21):4531–4533, 2019.

[167] Tony J. Wilkinson. Regional approaches to Mesopotamian archaeology: The contribution of archaeological surveys. Journal of Archaeological Research, 8(3):219–267, 2000.

[168] Tony J. Wilkinson, John Bintliff, Hans H. Curvers, Paul Halstead, Phillip L. Kohl, Mario Liverani, Joy McCorriston, Joan Oates, Glenn M. Schwartz, Ingolf Thuesen, Harvey Weiss, and Marie-Agnes Courty. The structure and dynamics of dry-farming states in Upper Mesopotamia. Current Anthropology, 35(5):483–520, 1994.

[169] Christoph Witzgall and Roger Fletcher. Practical methods of optimization. In Mathematics of Computation, volume 53, page 768, 1989.

[170] Huisi Wu, Fangyan Xiao, and Chongxin Liang. Dual contrastive learning with anatomical auxiliary supervision for few-shot medical image segmentation. In European Conference on Computer Vision, pages 417–434. Springer, 2022.

[171] Peng Xu, Fred Roosta, and Michael W. Mahoney. Second-order optimization for non-convex machine learning: An empirical study. In SIAM International Conference on Data Mining, Proceedings, pages 199–207. Society for Industrial and Applied Mathematics, 2020.

[172] Yunqiao Yang, Zhiwei Wang, Jingen Liu, Kwang-Ting Cheng, and Xin Yang. Label refinement with an iterative generative adversarial network for boosting retinal vessel segmentation, 2019. arXiv:1912.02589 [eess].

[173] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5212–5221, 2019.

[174] Huahong Zhang, Alessandra M. Valcarcel, Rohit Bakshi, Renxin Chu, Francesca Bagnato, Russell T. Shinohara, Kilian Hett, and Ipek Oguz. Multiple sclerosis lesion segmentation with tiramisu and 2.5D stacked slices. In Medical Image Computing and Computer Assisted Intervention, pages 338–346. Springer, 2019.

[175] Le Zhang, Ryutaro Tanno, Kevin Bronik, Chen Jin, Parashkev Nachev, Frederik Barkhof, Olga Ciccarelli, and Daniel C. Alexander. Learning to segment when experts disagree. In Medical Image Computing and Computer Assisted Intervention, pages 179–190. Springer, 2020.

[176] Le Zhang, Ryutaro Tanno, Mou-Cheng Xu, Chen Jin, Joseph Jacob, Olga Cicarrelli, Frederik Barkhof, and Daniel Alexander. Disentangling human error from ground truth in segmentation of medical images. In Advances in Neural Information Processing Systems, volume 33, pages 15750–15762. Curran Associates, Inc., 2020.

[177] Minqing Zhang, Jiantao Gao, Zhen Lyu, Weibing Zhao, Qin Wang, Weizhen Ding, Sheng Wang, Zhen Li, and Shuguang Cui. Characterizing label errors: Confident learning for noisy-labeled image segmentation. In Medical Image Computing and Computer Assisted Intervention, pages 721–730. Springer, 2020.

[178] Yihuan Zhang, Jun Wang, Xiaonian Wang, and John M. Dolan. Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor. IEEE Transactions on Intelligent Transportation Systems, 19(12):3981–3991, 2018.

[179] Aleksandar Zlateski, Ronnachai Jaroensri, Prafull Sharma, and Fredo Durand. On the importance of label quality for semantic segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1479–1487, 2018.

List of Publications

Publications that are part of this thesis:

1. Şerban Vădineanu, Daniël M. Pelt, Oleh Dzyubachyk, and Kees Joost Batenburg. An analysis of the impact of annotation errors on the accuracy of deep learning for cell segmentation. *In International Conference on Medical Imaging with Deep Learning*, Proceedings of Machine Learning Research, pages 1251–1267. PMLR, 2022.

2. Şerban Vădineanu, Daniël M. Pelt, Oleh Dzyubachyk, and Kees Joost Batenburg. Reducing manual annotation costs for cell segmentation by upgrading low-quality annotations. *In International Workshop on Medical Image Learning with Limited and Noisy Data*, Lecture Notes in Computer Science, pages 3-13. Springer, 2023.*

3. Şerban Vădineanu, Daniël M. Pelt, Oleh Dzyubachyk, and Kees Joost Batenburg. Reducing manual annotation costs for cell segmentation by upgrading low-quality annotations. *Journal of Imaging*, 10(7):172, 2024.*

4. Şerban Vădineanu, Tuna Kalayci, Daniël M. Pelt, and Kees Joost Batenburg. Convolutional neural networks and their activations: An exploratory case study on mounded settlements. *Journal of Computer Applications in Archaeology*, 7(1), 2024.

5. Şerban Vădineanu, Daniël M. Pelt, Oleh Dzyubachyk, and Kees Joost Batenburg. From feature maps to few-shot cell segmentation. *In International Workshop on Medical Optical Imaging and Virtual Microscopy Image Analysis*, Lecture Notes in Computer Science. Springer, 2025.†

Publications that are not part of this thesis:

1. Şerban Vădineanu and Mitra Nasri. Robust and accurate regression-based techniques for period inference in real-time systems. *Real-Time Systems*, 58(3), 2022.

∗: Paper 3 is the journal extension of Paper 2.

†: Paper 5 was awarded runner-up best paper at MOVI2024 workshop.

Summary

Image segmentation is a process that divides an image into distinct regions, identifying and categorising them based on shared characteristics such as colour, texture, or boundaries (see Figure S1 for a schematic representation of cell segmentation). This process has the potential to address a wide range of problems in specialised fields, such as detecting tumours in computed tomography scans for medical imaging or identifying sites in archaeological research. Currently, the best-performing image segmentation algorithms are based on deep learning.

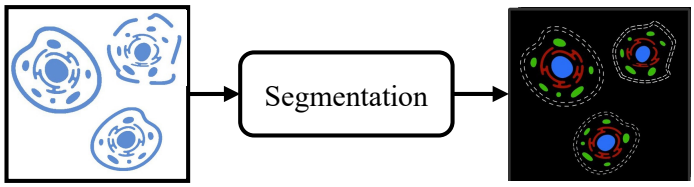


Figure S1: Example of image segmentation. The cell shapes and the segmentation are from [86].

Deep learning refers to a category of statistical models trained to perform tasks by learning from large datasets. For example, in image segmentation, the model is trained using pairs of input images and their corresponding annotations (i.e., categorised regions within the image). Consequently, a learning pipeline for deep learning segmentation algorithms typically includes an initial step in which annotations are generated to prepare for training (annotation process) and a second step where the algorithm is trained using the input images and the previously created annotations (training process). Figure S2 shows an overview of this pipeline.

Despite the promising results of deep learning in image segmentation, its widespread adoption in specialised domains remains hindered by challenges related to annotation

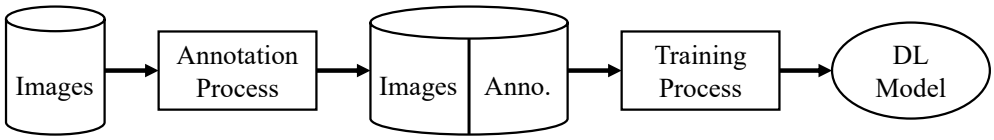


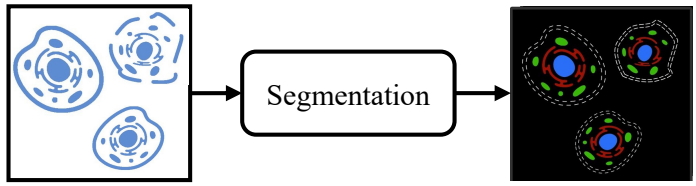
Figure S2: Deep learning pipeline from an initial set of images to a trained model.

and training. When it comes to annotation, marking the regions of an image is not only time-consuming—since it's often done manually—but also prone to mistakes due to human error or unclear boundaries. In specialised domains, this becomes even more difficult because annotators need expert knowledge, which adds costs in terms of time and availability. Experts in these fields often have demanding schedules, and the necessary knowledge may only be possessed by a small group of people. On the training side, deep learning algorithms need large amounts of data and operate with an opaque decision-making process, posing additional barriers. Collecting enough data can be expensive, requiring special equipment, preparing samples (as in medical imaging), or even travelling to specific locations (as in archaeology). Additionally, because deep learning models don't provide a clear, step-by-step explanation of their decisions, professionals in non-technical fields may be hesitant to rely on them.

This thesis provides a set of solutions to address the challenges associated with the annotation and training processes of deep learning algorithms for image segmentation in specialised domains. Specifically, we focus on two applications: cell segmentation in biomedical imaging and the detection of archaeological sites from satellite images. In Chapter 2, we investigate the impact of annotation errors on the performance of deep learning models for cell segmentation. Building on these findings, Chapter 3 introduces a training technique that enables deep learning models to learn from low-quality annotations, such as those with missing regions or imprecise boundaries. In Chapter 4, we propose a novel deep learning algorithm capable of performing cell segmentation using only a few annotated image-annotation pairs (e.g., 5 pairs). Chapter 5 explores the application of explainability techniques to deep learning models trained for image classification, generating visual representations of the image regions the model considers relevant. We analyse these visualisations to gain insights into the learned patterns and further refine them to create semi-automated annotations for archaeological site segmentation, reducing the time required from domain experts compared to manual annotation.

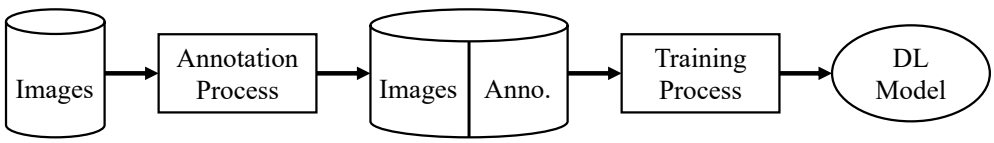
Samenvatting

Beeldsegmentatie is een proces dat een afbeelding opdeelt in afzonderlijke regio's, waarbij deze worden geïdentificeerd en gecategoriseerd op basis van gedeelde kenmerken zoals kleur, textuur of grenzen (zie Figuur S1 voor een schematische weergave van celsegmentatie). Deze techniek biedt mogelijkheden om een breed scala aan problemen in gespecialiseerde domeinen aan te pakken, van het detecteren van tumoren in computertomografiescans voor medische beeldvorming tot het identificeren van vindplaatsen in archeologisch onderzoek. Momenteel zijn de best presterende algoritmen voor beeldsegmentatie gebaseerd op deep learning.



Figuur S1: Voorbeeld van beeldsegmentatie. De celvormen en de segmentatie zijn afkomstig uit [86].

Deep learning verwijst naar een categorie statistische modellen die worden getraind om taken uit te voeren door te leren van grote datasets. Bij beeldsegmentatie bijvoorbeeld wordt het model getraind met invoerafbeeldingen en de bijbehorende annotaties (gecategoriseerde regio's binnen de afbeelding). Een training pipeline voor deep learning-segmentatie-algoritmen omvat doorgaans een eerste fase waarin annotaties worden gemaakt (annotatieproces), gevolgd door een tweede fase waarin het algoritme wordt getraind met de invoerafbeeldingen en de eerder gemaakte annotaties (trainingsproces). Figuur S2 toont een overzicht van dit traject.



Figuur S2: Deep learning ontwikkel pipeline van een initiële set afbeeldingen naar een getraind model.

Ondanks de veelbelovende resultaten van deep learning voor beeldsegmentatie, wordt de brede toepassing ervan in gespecialiseerde domeinen beperkt door uitdagingen in de annotatie en training. Binnen de annotatie fase is het marken van regio's in afbeeldingen niet alleen tijdrovend maar ook foutgevoelig door onnauwkeurig of verkeerd aangegeven grenzen, gezien dit proces vaak handmatig moet gebeuren. In gespecialiseerde domeinen is dit nog uitdagender, omdat deskundige kennis nodig is voor het annoteren, wat extra kosten met zich meebrengt en limitaties kent in beschikbaarheid van deskundigen. Experts in deze vakgebieden hebben vaak drukke agenda's, en de benodigde kennis is soms slechts aanwezig bij een kleine groep mensen. Wat de training betreft, vereisen deep learning-algoritmen grote hoeveelheden voorbeelden en hebben ze een slecht-inzichtelijk werkingsmechanisme voor het maken van beslissingen, wat ook weer extra obstakels oplevert. Het verzamelen van voldoende gegevens kan kostbaar zijn en speciale apparatuur vereisen, het voorbereiden van samples (zoals in medische beeldvorming), of zelfs reizen naar specifieke locaties (zoals voor archeologie). Een aanvullende uitdaging is dat deep learning-modellen geen duidelijke, stapsgewijze uitleg van hun beslissingsproces geven, waardoor eindgebruikers terughoudend zijn om erop te vertrouwen.

Dit proefschrift biedt een reeks oplossingen om de uitdagingen aan te pakken die gepaard gaan met de annotatie- en trainingsprocessen van deep learning-algoritmen voor beeldsegmentatie in gespecialiseerde domeinen. We richten ons daarbij specifiek op twee toepassingen: celsegmentatie in biomedische beeldvorming en de detectie van archeologische sites vanuit satellietbeelden. In Hoofdstuk 2 onderzoeken we de impact van annotatiefouten op de prestaties van deep learning-modellen voor celsegmentatie. Hierop voortbouwend, introduceert Hoofdstuk 3 een trainingstechniek die deep learning-modellen in staat stelt te leren van annotaties van lage kwaliteit, zoals annotaties met ontbrekende regio's of onnauwkeurige grenzen. In Hoofdstuk 4 presenteren we een nieuw deep learning-algoritme dat in staat is celsegmentatie uit te voeren met slechts een klein aantal geannoteerde afbeelding-annotatieparen (bijvoorbeeld 5 paren). Hoofdstuk 5 onderzoekt de toepassing van explainability-technieken op deep learning-modellen die zijn getraind voor beeldclassificatie. Hierbij worden visuele representaties gegenereerd van de afbeeldingsregio's die het model als relevant beschouwt. We analyseren deze visualisaties om inzicht te krijgen in de geleerde patronen en verfijnen ze verder om semi-automatische annotaties voor segmentatie van archeologische sites te creëren. Dit vermindert de benodigde tijd van domeinexperts in vergelijking met handmatige annotatie.

Curriculum Vitae

Șerban Vădineanu was born in 1995 in Corabia, Romania. In 2018, he graduated the Bachelor of Applied Electronics program at the Polytechnic University of Bucharest (second among the 2018 cohort). He subsequently obtained a Master's degree cum laude in embedded systems from TU Delft in 2020. His thesis, titled "Deriving timing properties from system traces using data-driven techniques", was supervised by Dr. Mitra Nasri. In 2021, he began his Ph.D. studies at Leiden University under the supervision of Prof.dr. K.J. Batenburg. During his Ph.D. studies, he took courses in speed reading, presentation skills, communication in science, and scientific conduct, among others.

Acknowledgements

Firstly, I would like to express my gratitude towards Prof.dr. Joost Batenburg and Dr. Daniël Pelt for the incredible support they offered me throughout my whole journey as a Ph.D. candidate. I consider myself very lucky to have collaborated with such sympathetic, kind and enthusiastic people who created a working environment in which I could thrive. I am especially grateful for the feedback they offered me, aimed not only at improving my academic skills, but also at polishing my interpersonal abilities.

I would also like to thank Dr. Oleh Dzyubachyk for making sure that my ideas were not only cool deep learning applications, but also that they held relevance to the biomedical field. In addition, I want to thank Dr. Tuna Kalayci for the fruitful conversations we had regarding both work and all sorts of life stories. This made me to always look forward to the next meeting in van Steenis building.

Additionally, I am grateful to have met Jiayang, my office mate for more than three years, with whom I shared struggles, successes and stories, and who made coming to the office all the more enjoyable. Moreover, I am happy to have been colleagues with André, an enthusiastic ambassador of all things Portuguese, and MC, whose contagious cheerfulness made sure everyone was in a good mood.

Among my LIACS colleagues, I would be remiss not to thank Sengim, Zhao, Tianyuan, Hazel, Thomas, Koen, Felix, Luc, Matthias, Alexander, Richard, Mischa, Irina, and Willem-Jan for being such an inclusive group that made any sort of conversation possible.

I owe special thanks to Yevhenii, who never missed an opportunity for a chat over a cup of tea, a great listener and an even greater friend.

Many thanks to Hermes, my first friend from Leiden and the best cook I know, for helping me translate my thesis summary.

A heartfelt "mulțumesc" goes to my Romanian friends, especially to Andrei and Bogdan, who proved to me that distance can actually strengthen the bonds between friends.

My sincere gratitude goes to my parents, whose sacrifices, love and understanding nature made my journey to The Netherlands—and this Ph.D.—possible to begin with. They engraved into me the respect for education and research, and I am glad that I can make them proud with the work I did for this thesis.

Lastly, I am deeply grateful to Lorena, my partner and my staunchest ally. It is difficult to imagine this journey without your unwavering support, love and compassion. You doubled the joy of every success and halved the burden of every setback—for that, and so much more, I thank you wholeheartedly.