FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of master's thesis

Title:	Segmentation of pancreatic islets and exocrine tissue from microscopic images using neural networks based approaches
Student:	Bc. Petra Čurdová
Supervisor:	Ing. Jan Kubant
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2023/2024

Instructions

The aim is to improve the existing solution for the segmentation of pancreatic islets and exocrine tissue.
Instructions:
1. Survey state-of-the-art techniques that are used for segmentation tasks in the medical imaging domain.
2. Explore the medical aspects of the task needed to understand the given data and propose how to evaluate the model results. The data will be provided by the supervisor.
3. Propose possible image augmentation.
4. Explore the current approach of pancreatic islets and exocrine tissue segmentation and its weaknesses. Source codes will be provided by the supervisor. The description of the current approach and results can be found at https://isletnet.com or https://owncloud.ikem.cz/index.php/s/MMURbfW1eeRi5M0?path=%2FIsletNet-info
5. Propose a new approach based on your research and implement it.
6. Compare the results of your model with the existing solution.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 21 November 2022 in Prague.

Master's thesis

SEGMENTATION OF PANCREATIC ISLETS AND EXOCRINE TISSUE FROM MICROSCOPIC IMAGES USING NEURAL NETWORKS BASED APPROACHES

Bc. Petra Čurdová

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Ing. Jan Kubant
January 11, 2024

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Čurdová Petra. *Segmentation of pancreatic islets and exocrine tissue from microscopic images using neural networks based approaches.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Contents

List of Figures

List of Tables

List of code listings

First, I would like to express my sincere gratitude to my supervisor, Ing. Jan Kubant, for his invaluable support, guidance, and insightful consultations that significantly contributed to the progress of this research.
Special thanks are owed to MUDr. David Habart, Ph.D. from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ, for professional consultations regarding the domain of pancreatic islets and for the reviewing and refining of the annotations that greatly enhanced their suitability for use in the proposed approach.
My thanks also go to the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ, for providing access to the dataset of pancreatic islets and their corresponding ground truth annotations. This dataset was instrumental in validating and refining the models.
Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

In Prague on January 11, 2024

Abstract

The success of pancreatic islet transplantation relies on the accurate estimation of beta cells amount within individual islets, primarily determined by the estimation of islet volumes from microscopic images of isolated islets samples. Previous research has applied digital image analysis to generate segmentation masks of islets, subsequently used for volume calculation. This thesis aims to analyse and address the limitations of an existing approach using the semantic segmentation model UNet to obtain islets masks from microscopic images. The analysis showed that most significant limitation is the inability of the model to distinguish individual instances of adjacent islets, leading to an overestimation of islet volumes and thus the amount of beta cells within the sample. In response to this problem, this thesis investigates instance segmentation models as an alternative to address the drawbacks of the UNet model. The model proposed in this thesis demonstrates the potential of this approach in pancreatic islets segmentation. It outperforms the state-of-the-art model in accurately separating individual islets while maintaining a comparable overall IoU score of the segmentations.

Keywords neural networks, instance segmentation, convolutional neural networks, deep learning, pancreatic islets, pancreas, microscopic images

Abstrakt

Úspěch transplantace pankreatických ostrůvků závisí na přesném odhadu množství beta buněk v jednotlivých ostrůvcích, které je primárně odhadováno na základě objemů ostrůvků získaných analýzou mikroskopických snímků vzorků izolovaných ostrůvků. Předchozí výzkum zkoumal aplikaci digitální obrazové analýzy pro získání binárních segmentačních masek ostrůvků následně použitých pro výpočet objemu. Cílem této práce je analýza a řešení nedostatků současného přístupu využívajícího sémantický segmentační model UNet k získání segmentačních masek ostrůvků z mikroskopických snímků. Analýza ukázala, že hlavním neodstatkem modelu je neschopnost rozlišit jednotlivé instance k sobě přiléhajících ostrůvků. To vede k nadhodnocení objemů ostrůvků a tím i množství beta buněk ve vzorku. Tato práce zkoumá modely segmentace instancí jako alternativu, která by mohla vyřešit nedostatky modelu UNet. Model navržený v této práci prokazuje potenciál tohoto přístupu pro použití v segmentaci pankreatických ostrůvků. Navržený model překonal UNet v přesnosti vymezení jednotlivých instancí ostrůvků a zároveň si zachoval porovnatelné celkové IoU skóre segmentací.

Klíčová slova neuronové sítě, segmentace instancí, konvoluční neuronové sítě, hluboké učení, Langerhansovy ostrůvky, slinivka, mikroskopické snímky

List of abbreviations

AP	Average Precision
AUC	Area Under the Curve
CNN	Convolutional Neural Network
COCO	Common Objects in Context
DIA	Digital Image Analysis
FC	Fully-Connected
FCN	Fully Convolutional Network
FFN	Feed-Forward Network
FN	False Negative
FP	False Positive
GT	Ground Truth
HTC	Hybrid Task Cascade
IE	Islet Equivalent
IoU	Intersection over Union
LN	Layer Normalization
LR	Learning Rate
MA	Multi-head Attention
MAE	Mean Absolute Error
mAP	Mean Average Precision
MSE	Mean Squared Error
NLP	Natural Language Processing
NMS	Non-Maximum Suppression
NN	Neural Network
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
RoI	Region of Interest
RPN	Region Proposal Network
SVM	Support Vector Machines
SW-MSA	Shifted Window Multi-head Self Attention
T1DM	Type 1 Diabetes Mellitus
TN	True Negative
TP	True Positive
W-MSA	Window Multi-head Self Attention

Introduction

The introductory section briefly describes the issue addressed in this thesis, explains why it is important to solve, and outlines the aim of this thesis.

Type 1 diabetes mellitus (T1DM) is a chronic illness that is estimated to affect 9 million people.[1] It arises from the inability of the pancreas to produce insulin due to autoimmune attack on insulin-producing beta cells within the pancreatic islets.[2] In most cases, managing T1DM involves continuous glucose monitoring and insulin therapy. Nevertheless, in severe cases, where insulin treatment might not be sufficient, a transplantation of pancreatic islets cells can be a more effective solution.[3]

Before transplantation, it is necessary to assess the quality and quantity of donor's pancreatic islets. This is done by evaluating multiple samples of an islet preparation obtained through procedure known as islet isolation.[4] Samples are observed under a microscope, where a trained operator manually counts the number of islets and measures the sizes of the individual islets to estimate the total islet mass of the samples.[5] Islet mass highly correlates with the clinical outcomes of the transplantation and thus is an important factor in determining whether the islets are suitable for transplantation.[6] However, manual measurement and estimation is time-consuming and prone to imprecision or high interoperator variability.[7, 8]

Consequently, various approaches based on digital image analysis or machine learning aim to speed up, automate, or improve accuracy of the measurement process.[8] Despite these efforts, none of the approaches have become widespread or been standardized across laboratories.[9] One of the most advanced approaches is based on the UNet semantic segmentation model, a convolutional neural network called IsletNet[10] designed to produce binary islet masks from microscopic images of isolated islets. These masks are subsequently used to determine individual islet sizes and to calculate their volumes. While this solution significantly accelerates and simplifies the workload for the operators, it still exhibits weaknesses that needs resolution before its clinical deployment across laboratories becomes viable.

Research in the field of convolutional neural networks is moving forward rapidly, and since UNet was implemented, many new architectures, approaches, and improvements to segmentation models have emerged that could address the shortcomings of the current model. Therefore, the aim of this thesis is to identify the weaknesses of the UNet model, and propose and implement an approach that will address its limitations. This new approach will be developed based on an exploration of state-of-the-art techniques in image segmentation, which will be described in this thesis.

This thesis will investigate the methods employed for the pancreatic islets segmentation, including the IsletNet model, and the medical aspects of this problem. It will summarize the limitations of the existing approaches and subsequently propose a novel solution based on the research of the state-of-the-art techniques in semantic and instance segmentation. The proposed

solution will be implemented, followed by suggestions for evaluating model outputs and the introduction of suitable data augmentation strategies, based on research and dataset analysis. A series of experiments will then be conducted to refine the proposed approach. Finally, this thesis will compare the best-performing model with the original UNet model to assess whether the identified weaknesses have been effectively resolved, and discuss the possible future improvements.

Chapter 1 introduces the theory related to pancreatic islets and islet cells transplantation, emphasizing the connection to islets segmentation from microscopic images. In Chapter 2, the research involving the utilization of digital image analysis and machine learning for islet segmentation is described. The rest of the theoretical part (Chapters 3-6) describes the important concepts and architectures of convolutional neural networks, transformers, semantic segmentation, and instance segmentation respectively. Chapter 7 presents the description of the dataset and data preparation process, while Chapter 8 provides a comprehensive description of conducted experiments and their comparison. Finally, in Chapter 9, the best model is evaluated against the IsletNet model on the test set.

Chapter 1
Background

This chapter will describe the fundamentals and function of pancreatic islets and their connection to type 1 diabetes mellitus (T1DM), and the methodology of treating T1DM through islet cell transplantation. It will also explore the quantity and quality assessment of islet cultures used in transplantation, highlighting existing limitations, and discussing the potential of neural networks in this assessment.

1.1 Type 1 diabetes mellitus

The pancreas consists of two distinct types of tissue: endocrine and exocrine. Endocrine tissue, which constitutes only around 1% of the pancreas, represents the pancreatic islets that consist of various cell types responsible for secreting essential hormones like insulin, glucagon, and somatostatin.[11] Specifically, insulin-producing beta cells are pivotal; their destruction, often by autoimmune or unknown causes, results in reduced insulin production, leading to insulin deficiency and hyperglycemia, characteristic of type 1 diabetes mellitus (T1DM).[12]

A study on the prevalence of T1DM estimates, that about 9 million people worldwide suffer from this disease and the number is increasing.[1] While most cases can be handled by insulin therapy and continuous glucose monitoring, some patients experience complications like hypoglycaemia unawareness (when patient is unable to recognize hypoglycemia due to the absence of typical warning symptoms[13]), severe hypoglycaemic episodes (state of hypoglycaemia when external assistance is required for the patient's recovery as they may be incapable of helping themselves[14]) or glycaemic lability (instability or frequent fluctuations in blood glucose levels over a certain period of time[15]).[16]

In severe cases where complications arise during insulin therapy, islet cell transplantation might be considered. However, this option requires careful evaluation of potential risks, including lifelong immunosuppressant use post-transplantation, which increases the risk of infections, cancer, and organ-system toxicity. As a result, islet cell transplantation is typically recommended only for the most severe cases due to the associated risks when compared to the risks of untreated complications of T1DM.[16]

1.2 Pancreatic islet cells transplantation

As mentioned above, islet cells transplantation is used only in severe forms of T1DM due to the considerable risks linked with the continual use of immunosuppressants. On the other hand, research from five leading islet transplantation centers has indicated a notable 50-70% insulin independence rate five years post-transplantation, a rate comparable to whole pancreas trans-

plantation, despite islet cell transplantation being a considerably less invasive procedure. Moreover, advancements in immunosuppression therapies are enhancing safety measures, potentially increasing the demand for islet cell transplantation in the future.[16]

Before the transplantation procedure, an islet culture needs to be prepared through a 5–7h process called islet isolation. At first, a pancreas is procured from a suitable deceased donor, followed by the digestion of the pancreas, mechanical separation, and purification through centrifugation.[17] As a result, a purified islet culture is obtained as shown in Figure 1.1. Prior to transplantation, the viability culture needs to be assessed and the culture is used only when it satisfies specific criteria.[18] Eventually, the prepared islet product is suspended in transplant media and is infused into the recipient's portal vein through catheterization.[16]

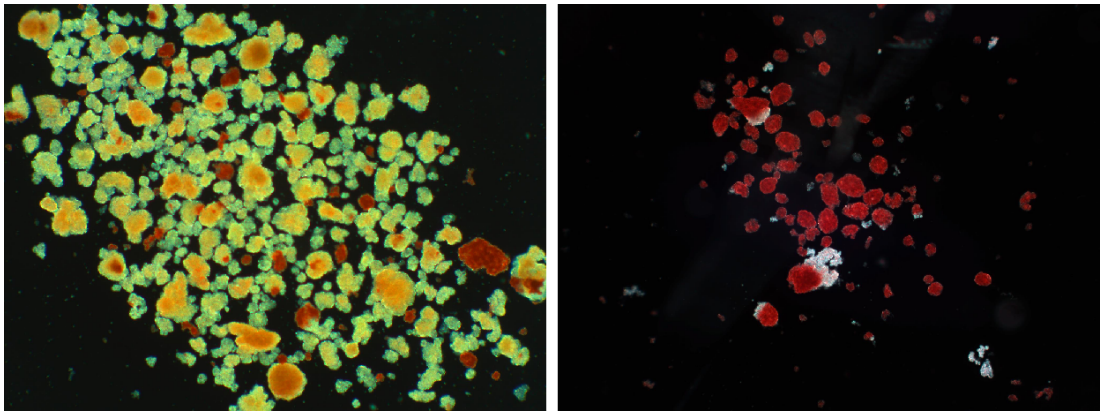

Figure 1.1 An image of isolated islet samples under a microscope, showing red-stained islets and yellow/white exocrine tissue. The left displays lower purity with more mixed exocrine tissue, while the right exhibits higher purity with predominantly red-stained islets. Images are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ

1.3 Islet quality and quantity assessment

After islet isolation, it is essential to assess the quality and quantity of islets in the isolated islet culture. This evaluation is conducted either before transplantation or for research purposes. Multiple samples are extracted from the culture for evaluation and stained with dithizone to differentiate the islets from surrounding tissue by giving the islets a distinct red coloration. Subsequently, a manual evaluation is performed.[5]

1.3.1 Islet number and volume

The conventional method to determine islet number and volume from the sample is usually referred as the manual method.[8] The manual method involves an experienced operator observing islets through a microscope, manually counting the number of islets in the sample. During the counting, the diameter of each islet is measured using a calibrated grid within the microscope. These islets are then categorized into size groups of 50 μm diameter ranges. Subsequently, each islet is converted into the number of islet equivalents (IE), representing an equivalent number of islets with a diameter of 150 μm, referencing Table 1.1.[5]

A study examining the relationship between islet product characteristics and transplantation outcomes revealed that the only characteristic significantly correlated with transplantation success was the total number of IEs. Consequently, achieving precise estimations of IE within the

Table 1.1 Conversion of each islet diameter group to IE[5]

Islet diameter range (μm)	Mean volume (μm^3)	Conversion into IE
50-100	294 525	$n/6.0$
100-150	1 145 373	$n/1.5$
150-200	2 977 968	$n \times 1.7$
200-250	6 185 010	$n \times 3.5$
250-300	11 159 198	$n \times 6.3$
300-350	18 293 231	$n \times 10.4$
350-400	27 979 808	$n \times 15.8$
400-450	40 611 628	$n \times 23.0$
450-500	56 581 390	$n \times 32.0$

islet sample is crucial.[6] However, the manual estimation method is prone to imprecision and significant variability among operators.[7, 8] To reduce counting errors, it is advisable to procure multiple samples of the preparation.[5] Nevertheless, the manual estimation process is notably time-consuming, often limiting the evaluation to only a few samples from the islet preparation.[4]

The method of estimating IE number using Table 1.1 introduces unnecessary rounding into the volume calculation and assumes the islet to have a spherical shape, which is not typically accurate.[19] Nevertheless, despite these limitations, it remains the most suitable approach for manual evaluation.

1.3.2 Purity

The purity of a sample is the percentage of endocrine tissue (islets) compared to all tissue in the sample.[20] An example comparison of a sample with low and high purity is depicted in Figure 1.1. This parameter is only roughly estimated by the operator and not calculated or measured as its exact determination is not considered essential. While purity alone may not predict transplantation outcomes, the presence of non-islet tissue can serve as an indicator of transplant suitability when considered alongside factors like islet count and volume. Given that the maximum graft volume allowed is usually based on the required islet volume, aiming for high purity is favorable, though precise estimation is not mandatory.[21]

1.3.3 Digitization of the islet sample assessment

The manual method for islet graft assessment presents several drawbacks, including high interoperator variability, the absence of image archiving for sample verification, and being time-consuming. Therefore, substantial efforts, that will be described in the next chapter, have been directed towards digitizing and automating the islet graft assessment process. The novel approaches employ digital image analysis, generating binary masks of islets and exocrine tissue with the assistance of experienced operators or through automated processes. These masks are subsequently used for the automated quantification of islets, volume computation, and purity estimation. This approach improves accuracy and reduces interoperator variability compared to the manual method.[8, 9]

Additionally, it enables the adoption of more sophisticated methods for islet volume calculation. For example, one such method considers the islet to have an elliptical shape, which, according to a few studies, outperforms the assumption of a spherical shape.[4, 19] In the ellipse method, an ellipse is fitted to the binary segmented mask of an islet, and the volume is estimated as that of an ellipsoid, calculated using the formula $V = (3/4) \cdot \pi a^2 b$, where a and b represent the lengths of the larger and smaller semiaxes, respectively.[19]

In this context, achieving high accuracy in segmenting masks of islets and exocrine tissue is crucial for precise calculations of islet count, volume, and purity. Convolutional neural networks (CNNs) emerge as a suitable solution for this task. Their ability to understand spatial hierarchies, extract features, and integrate both local and global context makes them well-suited for discerning the intricate structures present in pancreatic islet images. CNNs adapt to varied image characteristics, ensuring robust performance across different sizes, shapes, and intensities. This adaptability enhances robustness and performance across different datasets, ultimately reducing counting and estimation errors.[22]

Chapter 2

Pancreatic islet segmentation methods

In this chapter, an overview of existing methodologies for pancreatic islet segmentation through digital image analysis and machine learning will be presented.

The use of digital image analysis for pancreatic islet segmentation already has been investigated in previous studies.[8, 21, 23] This was primarily driven by the drawback of manual measurement including its time-extensive nature[4] and inaccuracy of manual measurement as well as substantial interoperator variance[9]. Research has shown that digital image analysis offers many advantages over the classical procedure, namely speed, reduced subjectivity, grater reproducibility and the ability to archive segmentation for further evaluation[9, 8]. However, the proposed methods are typically only semi-automatic, require specific software and hardware, or their applicability was tested on a restricted dataset. For instance, they were exclusively evaluated on high-purity samples, making them unsuitable for samples with lower purity levels.[8]

2.1 Thresholding

The most commonly cited approach for pancreatic islet segmentation is thresholding. This method is usually done using specialized software for image analysis including ImageJ, MetaMorph and others[24, 7, 4]. Various approaches were described for implementing thresholding.

Stegemann et. al. [25] in 1998 proposed an automatic method for grayscale images preprocessed by a contrast-enhancing filter. They developed a macro for DIA software, enabling the segmentation of images into islets, unstained impurities and background using gray-level thresholding. Their study involved a comparison of islet and IE counts obtained through DIA versus manual methods. They evaluated their approach on 140 porcine isolations and reported relatively strong correlation in counts ($p < 0.001, r = 0.78$) and IE number on average of 46% lower when using the manual method than when determined using DIA. They noted challenges in distinguishing dark impurities from stained islets and noted that the majority of the images they worked with had purity $> 90\%$.

Subsequent research mostly focused on color images. Niclauss et. al. (2008) [24] employed a thresholding on red color for islet segmentation and white color for exocrinne tissue segmentation. Although there were small differencies between the islet count and IE obtained by this algorithm compared to manual estimating on the digital images by experts (islet counts were $100,578 \pm 8,44$ and $96,555 \pm 8,581$ respectively and IEs $93,280 \pm 8,110$ and $94,426 \pm 7,863$), this analysis was conducted on a limited sample size of only 12 samples.

Other studies have employed color-based thresholding methods, wherein the operator selects specific colors for segmentation within the DIA software[7, 26, 27]. Nevertheless, it is important to note that these techniques are only semi-automated.

Some algorithms have been integrated into microscope software as well[28, 9]. One publicly described approach presented by Gmyr at. al. (2015)[28], relies on substracting color channels of the input RGB images to generate segmentation mask. Specifically, islets were segmented by substracting the green channel from the red channel (green-red) to obtain the segmentation mask by subsequent thresholding. Similarly, for exocrinne tissue segmentation, red-green-blue channel subtraction is performed, following by thresholding to generate the mask.

This approach was compared with the manual method on 42 islet preparations. The result showed a strong correlation ($r^2 = 0.88$) for islet count and ($r^2 = 0.91$) for IE number. However, it is worth noting that this method is dependent on precise adherence to the isolation process and dithizone staining of the islets, consistent light conditions and requires to buy specific hardware.

Additionally, it is important to highlight that no study employing thresholding techniques has compared the obtained segmentation masks with GT segmentation masks. Instead comparisons have been done on counts, sizes, volumes, and purity with the manual method, which may not yield highly accurate results.

2.2 Linear classifier and SVM

Švihlík et. al. (2014)[4] introduced a new automated technique. This method involved training a pixelwise linear classifier and linear SVM using 11 x 11 pixel rectangular regions chosen by user. These rectangular selections were subsequently flattened into a vector of 121 training samples, which were used for the classifier training. Classifiers were trained only on one image using 20 rectangular frames for both the islet and background classes. The quality of segmentation was tested on the remaining images.

The segmentation masks obtained from linear classifier were compared with ground truth segmentation masks, which were derived from the consensus of labeling by three experts. The study demonstrates that the proposed algorithm is comparable to the experts' performance (see Figure 2.1).

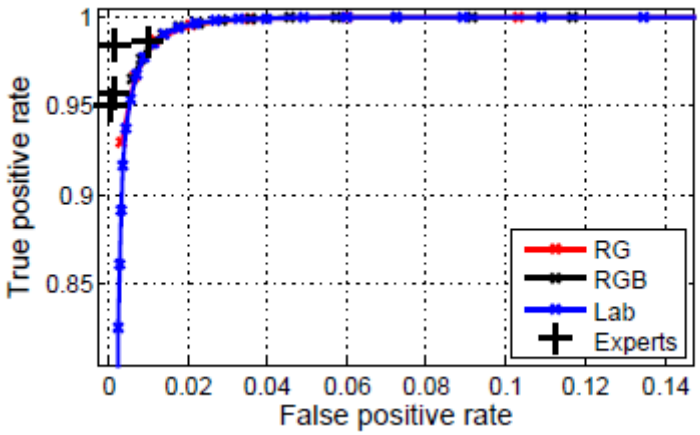

Figure 2.1 ROC calculated for linear classifier using all testing images and different color spaces. It compares the performance of the model with the performance of experts.[4]

Furthermore, to compare both algorithms, IE were computed based on their respective segmentation as well as from the GT segmentation. The relative error δ and absolute error ϵ were

calculated for both algorithms (see 2.1). The difference between the two methods were not statistically significant (by t-test at $\alpha = 0.05$).

Table 2.1 Comparison of SVM and linear classifier after calculation of IE numbers using ellipse algorithm (δ - mean relative error, ϵ - mean absolute error)[4]

Classifier	δ [%]	ϵ [−]	SD [−]	ϵ/SD
SVM	8.2	25.0	40.2	0.9
Lin. classifier	11.4	36.7	40.2	1.2

However, this approach requires a similarity in colors between the islets and the background in unseen images compared to the training set. Consequently, its ability to generalize over images with different color characteristics is limited. To overcome this problem, they introduced a color normalization algorithm for the input images[29]. Nonetheless, this particular approach has not been evaluated on images with significantly divergent colors or on images with low purity. Additionally, it operates under the assumption that the background class is the most prevalent in the images.

2.3 Random forest classifier

Another fully automated method for segmentation pancreatic islets was introduced by Habart et. al (2016)[21]. At first, the input images were preprocessed using color normalization, as previously mentioned[29]. Subsequently, a random forest classifier generated a probability map of islets based on individual pixels. This probability map was then transformed into a binary mask through the application of spatial regularization.

The classifier was trained using 46 images from 4 donors and then evaluated on 36 images from nine independent donors against GT segmentation masks obtained from trained experts. The assessment revealed a negligible pixel-wise relative error ($RE = 0.04$). However, it is important to note, that islet and background classes are typically highly imbalanced. Nonetheless, the method demonstrated a very strong correlation in islet counts ($r^2 = 0.92$) and volume ($r^2 = 1.0$).

Similar to the linear classifier, this method exhibits comparable limitations. Specifically, the preprocessing algorithm assumes that the most prevalent class is the background and has not been tested on images with varying color profiles. Furthermore, there are instances where two adjacent islets are occasionally identified as a single islet, and occasional misclassification of exocrine tissue as islets occurs.

2.4 Neural networks

In 2017, a novel approach was introduced using a neural network structured on an 18-layered UNet architecture[30] named IsletNet. The network was trained on data obtained by a manual segmentation conducted by an expert using a validated methodology. Evaluation on 128 test images revealed an average relative error of 10% for the islet count and 18% for volume estimation. The model successfully paired 87% of predicted individual islets with their ground truth counterparts, yet struggled with adjacent islets, leading to overestimated sample volumes. Despite this limitation, the approach demonstrated strong generalization capabilities and provided precise segmentation masks, displaying an F1 score of 0.895 ± 0.080 (mean $\pm$ std). Any discrepancies primarily arose near the islet boundaries. In summary, while this approach exhibits promising results, ongoing enhancements and training on larger datasets are underway.[10] One enhancement explored involves implementing the watershed transform, as discussed in the following section, which is applied on the segmentation mask to separate the islets. Nonetheless, this method tends to generate an excessive number of segments for the islets, presenting a suboptimal outcome.[31]

2.5 Watershed

Segmentation algorithms have problems to correctly separate individual islets. The separation is usually done after the segmentation as a post-processing in two steps.[31] First is called distance transform; a binary segmentation mask where ”1” denotes a pixel that corresponds to an islet and ”0” a pixel that corresponds to everything else. Then for each islet pixel a distance (typically Euclidean) to the closest non-islet pixel is calculated.[32]

Second step, watershed transform, is performed on the output of the previous step. It is based on the analogy of a topographic landscape, where pixel intensities in an image correspond to elevations on the landscape. The algorithm operates as if the pixel intensities represent a relief map. It simulates a flooding scenario starting from ”markers” or ”seeds” placed at specific points in the image. The flooding propagates throughout the image, filling basins (or regions) with water. When these ”flooded” basins meet, they form boundaries that delineate separate regions.[33] The output of this algorithm are separated islets.

An issue commonly associated with the watershed transform is its tendency to create excessive segmentation within images. Švihlík et. al.[31] introduced a potential solution to this challenge in the context of pancreatic islet segmentation. Their approach involved an evaluation of each potential division generated by the watershed transform based on various shape descriptors, such as circularity. Subsequently, the most plausible division was selected as the most accurate representation. This modified watershed technique outperformed the standard single watershed method and demonstrated comparable results to those achieved by medical experts when evaluated on a test set comprising 12 images. However, this algorithm is computationally demanding, requiring approximately 7 minutes to process a single image.

2.6 Summary

Multiple methods have been attempted to address pancreatic islet segmentation, ranging from basic thresholding to employing convolutional neural networks (CNNs). However, none of these approaches have gained widespread adoption due to inherent limitations. Most of the methods lack full automation, resulting in time-consuming processes. Additionally, they exhibit sensitivity to specific image subsets, variations in microscope settings, and alterations in lighting and colorspace. Among described methods, IsletNet, a CNN-based solution, stands out as one of the most robust; however, like other approaches, it fails to accurately delineate individual islet instances. The watershed transform, while offering some promise, tends to over-segment instances, presenting an undesirable outcome. This thesis endeavors to investigate the potential resolution of this challenge through the application of instance segmentation. Unlike traditional segmentation methods, instance segmentation not only generates segmentation masks for objects but also distinguishes individual object instances. By exploring CNNs and instance segmentation models in subsequent chapters, this thesis aims to address the limitations observed in previous approaches and potentially offer a solution to accurately delineate separate islet instances in pancreatic image analysis.

Chapter 3

Convolutional Neural Networks

This chapter briefly describes the functionality of Convolutional neural networks (CNNs) and their applications. Additionally, it focuses on detailing two CNN architectures, namely ResNet and ResNeXt, used in the practical part of this thesis.

Convolutional neural networks (CNNs) represent powerful frameworks for artificial intelligence and computer vision. Neural networks (NNs), inspired by the functioning of the human brain, comprise interconnected nodes that process information through layers, enabling pattern recognition, decision-making, and complex data analysis. CNNs, a type of NN, excel in visual perception tasks due to their unique architecture designed to analyze and interpret structured grid-like data, such as images and videos.

CNNs demonstrate remarkable adaptability across various fields, serving as a robust tools designed for a range of computer vision tasks visualized in Figure 3.1:

Image classification Involves categorizing images into predefined classes or categories. CNNs excel in recognizing patterns within images to assign them to distinct labels, facilitating tasks like identifying whether an image contains a cat or a dog, recognizing handwritten digits, or classifying different types of vehicles.

Semantic segmentation Focuses on pixel-level understanding by assigning class labels to each pixel in an image, thereby segmenting the image into meaningful parts or regions. CNNs in semantic segmentation contribute to tasks like delineating object boundaries or segmenting areas of interest, used for example in medical image analysis or scene understanding in autonomous driving.

Object detection Involves the recognition and precise localization of multiple objects within an image by outlining their specific positions using bounding boxes. CNNs applied in object detection not only identify various objects within an image but also determine their exact spatial positions, enabling their use in security surveillance, autonomous vehicles, and content-based image retrieval.

Instance segmentation Extends object detection by not only identifying objects using bounding boxes but also by providing detailed segmentation masks for each instance, crucial in tasks where precise delineation of multiple objects is necessary, such as robotic manipulation, interactive image editing, or medical imaging for organ segmentation.

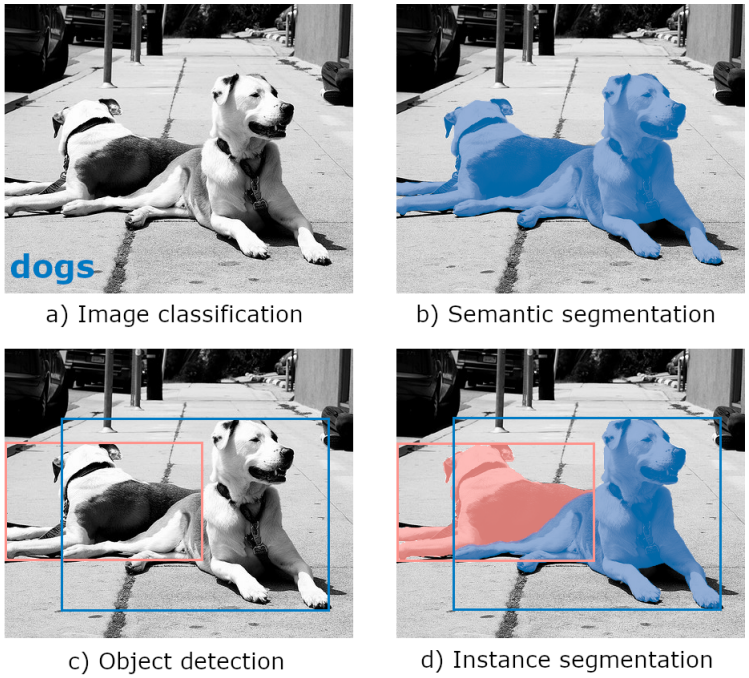

Figure 3.1 Computer vision tasks: a) image classification, b) semantic segmentation, c) object detection, d) instance segmentation. Image is from the COCO 2017 dataset[34], modified.

3.1 CNN architectures

Neural network architectures defines the structured organization of interconnected layers and nodes and influences the capacity of the network to process complex data, notably in tasks like semantic and instance segmentation in computer vision. ResNet[35] and ResNeXt[36] represent important architectural advancements that have revolutionized the domain of deep learning. ResNet addresses the challenges of training deep neural networks and enabled their training, while ResNeXt emphasizes the width of the network for more efficient feature extraction. These architectures, ResNet and ResNeXt, continue to be widely utilized in semantic and instance segmentation tasks due to their enduring significance and efficacy.

3.1.1 ResNet

ResNet, short for Residual Network, is a deep convolutional neural network architecture introduced by He et. al. in 2015 [35]. For earlier architectures it was impossible to train a deep network as they suffered from the vanishing gradient problem. ResNet addresses this problem by introducing shortcut connections, which allow information to flow directly from one layer to another, bypassing some layers in between. This made possible to train a very deep networks which are crucial for tasks like object detection or instance segmentation.

The ResNet architecture's, as shown in Figure 3.2, fundamental building blocks are residual blocks. Each block consists of two or more convolutional layers, typically followed by batch normalization and ReLU activation functions. The important feature of the architecture is the shortcut connection which directly passes the input to the output of the block. This allows the network to learn a residual function that represents the difference between the input and output of the block. By focusing on learning these residuals, the network can effectively learn the finer details and nuances in the data.

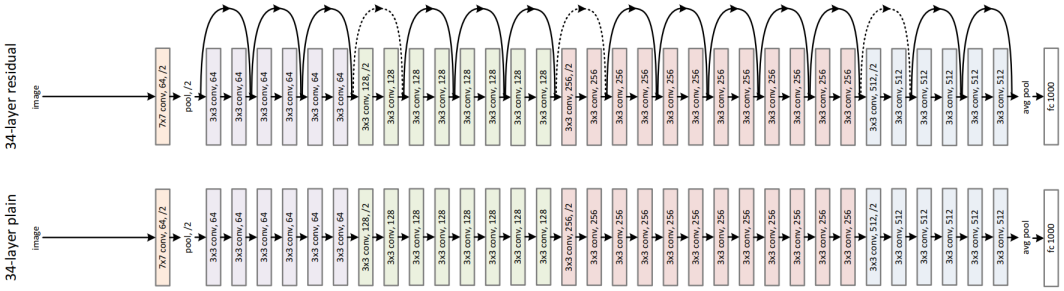

Figure 3.2 Bottom: a plain network with 34 layers, **Top**: ResNet with 34 layers.[35]

ResNet architectures come in various depths, for instance ResNet-50, ResNet-101, or ResNet-150, where the number in the name indicates the total number of layers in the network. Deeper versions have demonstrated high performance on challenging tasks like object detection, semantic segmentation, and instance segmentation.

3.1.2 ResNeXt

In 2016, Xie et. al. [36] proposed new deep learning CNN architecture ResNeXt that builds upon the concepts introduced by ResNet. The aim was to further enhance the performance of deep neureal networks. They introduced new building block called cardinality bottleneck that allows more efficient feature learning.

Traditional ResNet architectures focus on increasing the depth of the network to improve performance. Instead, ResNeXt emphasizes the importance of width in the network. The cardinality bottleneck introduces a new dimension, called ”cardinality”, which refers to the number of multiple smaller pathways within each building block, as shown in Figure 3.3. Furthermore, instead of processing all the channels together, they are divided into groups, and each group is processed separately within the cardinality bottleneck block.

ResNeXt has demonstrated state-of-the-art performance on a wide range of computer vision tasks, including image classification, object detection, and segmentation.

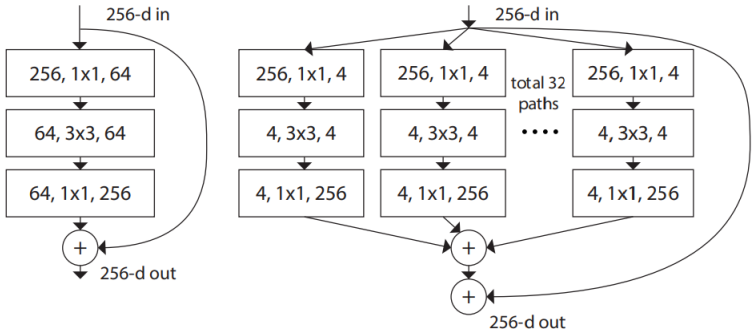

Figure 3.3 Left: a ResNet block, **Right**: a ResNeXt block with cardinality = 32.[36]

Chapter 4
Transformers

This chapter introduces transformers as an alternative method for instance segmentation, alongside CNNs. It describes the fundamental components within transformers, namely the attention mechanism and the feed-forward network (FFN). Furthermore, it details a specific transformer architecture known as the Swin Transformer, specifically designed for computer vision tasks.

In addition to CNNs, transformers are another viable option for performing instance segmentation tasks. Transformers represent an advancement in deep learning architectures initially developed for sequence-to-sequence tasks, predominantly in natural language processing (NLP). Originally, transformers revolutionized language modeling and translation tasks by introducing the self-attention mechanism, enabling efficient capturing of long-range dependencies in sequences without recurrent connections.[37]

Currently, transformers have surpassed their initial focus on NLP and have been employed in various domains beyond sequential data processing. Their adaptability, scalability, and ability to capture global dependencies and contextual relationships has led to their integration into visual tasks, such as image segmentation, object detection, and other computer vision tasks. [38, 39]

4.1 Transformer architecture

The transformer, introduced by Vaswani et al.[37] in 2017 is and encoder-decoder architecture. The encoder processes input sequences, while the decoder generates output sequences. Both the encoder and decoder consist of N identical layers stacked on top of each other, each layer involving two sub-modules: self-attention and position-wise feed-forward networks (FFNs). Additionally, the block contains a residual connection that surrounds each sub-module followed by a layer normalization.

4.1.1 Attention mechanism

The attention mechanisms in transformers are important for the ability of the model to to focus on different parts of input data when processing sequences. Attention mechanisms allow the model to assign varying degrees of importance or relevance to different elements within the input sequence. It is a function of three vectors: query, key, and value. These vectors are derived from the input sequence and used to calculate attention scores that represent the importance of each word in relation to all other words within the sequence.

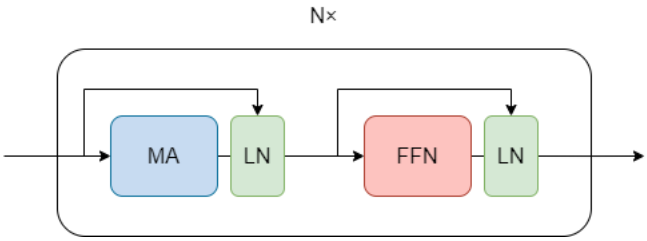

Figure 4.1 A diagram of one layer within the encoder of a transformer - MA: multi-head attention, LN: layer normalization, FFN: feed-forward network.[37]

For self-attention mechanism, all of the queries, keys, and values come from the same source. For instance, all of them come from the output of the previous layer. Multi-head attention extends the self-attention mechanism, enhancing the capability of the model to focus on different parts of the input sequence simultaneously. It accomplishes this by performing multiple parallel self-attention computations, each with its own set of query, key, and value transformations. The outputs from these multiple attention heads are then concatenated and linearly transformed to generate the final attention output. By employing multiple attention heads, the model can focus on different parts of the sequence concurrently, enabling more sophisticated and diverse patterns to be captured and utilized in subsequent tasks.[37, 40]

4.1.2 Position-wise feed-forward network

The positional-wise FFN is a fully connected FFN, typically consisting of two fully connected layers, that follows the self-attention mechanism. This network operates independently on each position within the sequence. It processes the outputs of the self-attention mechanism by applying linear transformations, followed by non-linear activation functions, such as the rectified linear unit (ReLU).[37]

4.2 Swin Transformer

Swin transformer is an innovative deep learning architecture proposed by Liu et. al. in 2021[41]. They address the challenge of adapting transformers to computer vision tasks by implementing an operation called Shifted windows, which leads to much greater efficiency of transformers on image data.

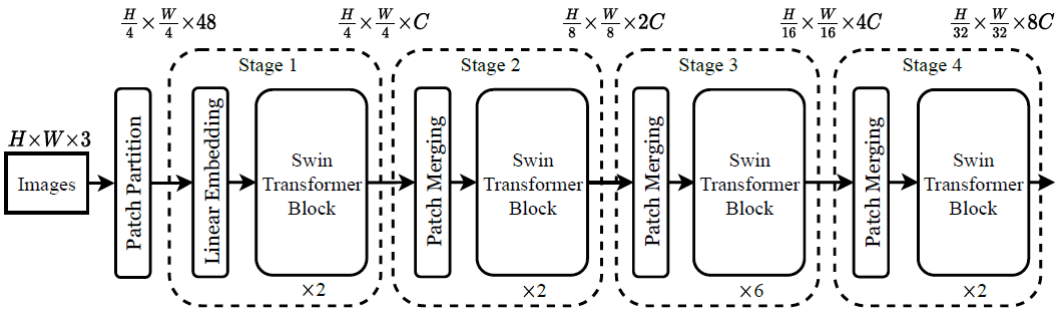

Figure 4.2 Swin Transformer architecture[41]

The overall architecture is outlined in Figure 4.2. The input image is, at first, split in the Patch Partition layer into non-overlapping patches, usually of size 4 x 4. The pixel values of the

patch are then concatenated into a vector over all image channels. That produces a vector of length 4 x 4 x 3 = 48. Then in stage 1, the vector is processed by a Linear Embedding layer, that produces a vector of length C. Then a Swin Transformer Block is applied on the vector, which is then passed to the next stages consisting of Patch Merging layers and Swin Transformer Blocks, as shown in Figure 4.2.

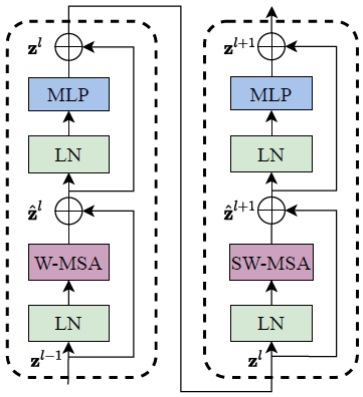

Figure 4.3 Two consecutive Swin Transformer blocks - W-MSA and SW-MSA are regular and shifted window multi-head self attention modules, MLP is multi-layer perceptron also called feed-forward network.[41]

Swin Transformer block (see Figure 4.3) replaces the traditional multi-head self attention block in transformer architecture by a block that uses shifted windows. Standard self-attention mechanisms operate on a fixed grid, which may not effectively capture spatial relationships in the image. In contrast, the shifted windows approach incorporates positional shifts with every additional self attention layer as shown in Figure 4.4. At first, a regular window partitioning is employed in the first Swin Transformer block, as showed on the left side of Figure 4.4. Subsequently, the next block uses a windowing configuration that is spatially shifted from the arrangement of the preceding layer, as illustrated on the right side of Figure 4.4.

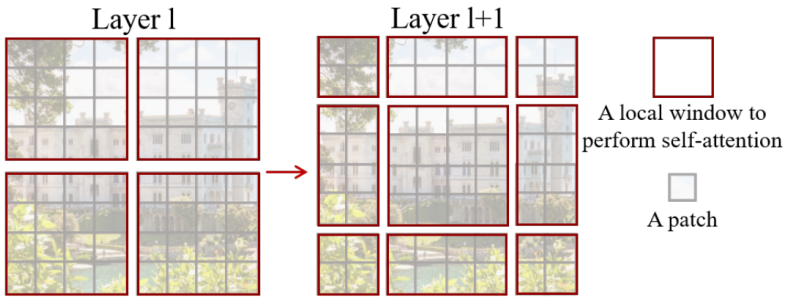

Figure 4.4 Shifted windows algorithm. Layer l uses regular window partitioning, layer l + 1 uses shifted windows partitioning that provides connections between the boundaries of previous windows.[41]

The proposed architecture surpassed the previous state-of-the-art in many computer vision tasks like object detection, instance segmentation, and semantic segmentation.

Chapter 5

Semantic segmentation

This chapter provides an overview of semantic segmentation, its practical applications, and delves into the widely adopted semantic segmentation architecture, the UNet, frequently used in medical image segmentation. Furthermore, it outlines commonly employed evaluation metrics while discussing the challenges and limitations in semantic segmentation.

Semantic segmentation is an important task in the domain of computer vision aiming to partition an image into coherent regions or segments, where each pixel is labeled with a specific class or category, thereby enabling a comprehensive understanding of its contents. For instance, in medical imaging, it aids in precise organ segmentation and disease diagnosis[42]. In autonomous driving, it enables vehicles to perceive and interpret the surrounding environment accurately[43]. Moreover, in satellite imagery analysis or robotics, semantic segmentation plays a crucial role in scene understanding and decision-making[44].

Typically, semantic segmentation tasks employs deep learning architectures, especially convolutional neural networks (CNNs) or more advanced models that capture complex spatial details, contextual relationships, and class-specific features within the image data. Especially prominent within medical imaging, the UNet architecture serves as a foundational model for semantic segmentation tasks, often inspiring subsequent models that adopt the architecture of the UNet while implementing variations or adaptations to enhance performance.[42]

5.1 UNet

Developed by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015, UNet[30] represents an effective architecture for semantic segmentation, particularly on smaller datasets, enabling its usage in medical imaging tasks.

UNet architecture, illustrated in Figure 5.1, is comprised of two paths: the contracting path, situated on the left side, and the expansive path, positioned on the right side. The contracting path initiates the network with convolutional layers, typically utilizing 3x3 convolutions, combined with rectified linear unit (ReLU) activation functions. Following each convolution block, the spatial dimensions are reduced through max-pooling operations employing a 2x2 window and a stride of 2. This process sequentially doubles the number of feature channels, enabling the capture of hierarchical features while progressively reducing spatial dimensions.

In contrast, the expansive path involves the upsampling of feature maps to amplify spatial dimensions. At each stage, the upsampled feature maps are concatenated with the corresponding cropped feature maps derived from the contracting path. The concatenated feature maps then undergo two 3x3 convolutions with ReLU activation functions. Final layer, a 1x1 convolutional

layer maps the final 64-component feature vector to the number of output classes.

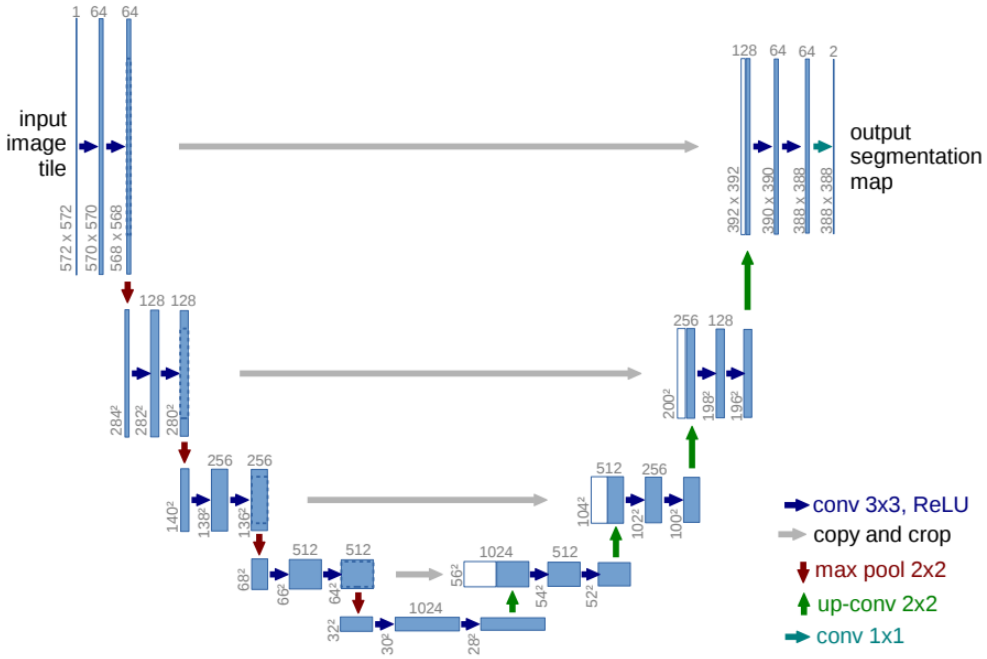

Figure 5.1 UNet architecture.[30]

5.2 Evaluation metrics

Semantic segmentation involves specific evaluation metrics that measure the overlap between the predicted and ground truth segmentation masks. That at first involves classifying each pixel as true positive (TP), true negative (TN), false positive (FP), or false negative (FN). The most widespread metrics are intersection over union (IoU) or Dice Similarity Coefficient (DSC), also known as F1 score. These metrics are very similar and calculated as follows[45]:

$$IoU = \frac{TP}{TP + FP + FN} \tag{5.1}$$

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{5.2}$$

5.3 Challenges and limitations

The primary limitation in the context of this thesis involves the inability to distinguish individual objects within the segmentation mask. This becomes particularly significant in certain applications like medical image segmentation, where precise identification of individual cells or structures is crucial, or in scene segmentation for autonomous driving that demands the tracking of separate objects. A plausible solution to this issue involves implementing an instance segmentation model instead of semantic segmentation.

Moreover, the annotation process required for semantic segmentation is exceedingly time-consuming, leading to limited training data, especially prevalent in the domain of medical imaging. To mitigate this challenge, two strategies are commonly employed:

Data augmentation With limited data availability, data augmentation is a helpful technique to enhance network robustness and minimize the likelihood of overfitting.[30] Augmentation methods commonly applied to image data include transformations like mirroring, rotation, resizing, as well as adjustments in colorspace such as modifying contrast, brightness, or shifting the color spectrum. These techniques effectively expand the dataset and adds more variability into the training set.

Transfer learning Transfer learning refers to a strategy where knowledge gained from solving one problem is applied to a different but related problem. In this approach, a pre-trained model, trained on a large dataset for a specific task, serves as a starting point for training a model on a different task or dataset. By leveraging the learned features or parameters of the pre-trained model, typically through fine-tuning or feature extraction, the model can adapt and perform effectively on the new task with a smaller dataset.[46]

Chapter 6

Instance segmentation

This chapter covers key elements in instance segmentation models, including essential building blocks, prevalent loss functions, and evaluation metrics. It also explores three major frameworks: Mask R-CNN, Cascade Mask R-CNN, and HTC.

Instance segmentation is a computer vision task that combines two other computer vision tasks, which are semantic segmentation and object detection. Semantic segmentation classifies each pixel of an image into fixed number of classes without identifying different object instances and thus creates a segmentation mask. On the other hand, object detection aims to localize each individual object using bounding boxes and classifying the object into fixed number of classes. Instance segmentation localizes individual objects within image, classifies the object in a one of given classes, and produces a segmentation mask for each object.

6.1 Instance segmentation model architecture

This thesis focuses on two-stage instance segmentation model architectures illustrated in Figure 6.1. In the initial stage, the model identifies objects' bounding boxes through a region proposal mechanism. This involves leveraging a convolutional neural network (CNN) backbone, such as ResNet or ResNeXt, to extract high-level features from the input image. These features are then passed through a region proposal network (RPN) or a similar mechanism, which generates potential regions of interest (RoIs) where objects might be present.

The second stage of the architecture involves detailed instance segmentation within these proposed regions. This is achieved by simultaneously predicting class labels, bounding box refinements, and pixel-wise masks for each object instance within the proposed regions. This process often utilizes a pixel-level segmentation network applied to each RoI, enabling precise delineation of object boundaries and the creation of masks indicating the pixel-level object segments.[47]

6.1.1 Region Proposal Network

Region Proposal Network (RPN) is an integral component in instance segmentation frameworks, particularly in models like Mask R-CNN[47], and related architectures. The primary function of the RPN is to generate a set of candidate bounding boxes or regions of interest (RoIs) where objects might be located. It achieves this by efficiently scanning the entire image using sliding windows of varying sizes and aspect ratios. Additionally, RPN employs a CNN that learns to predict objectness scores and refine them based on learned features.

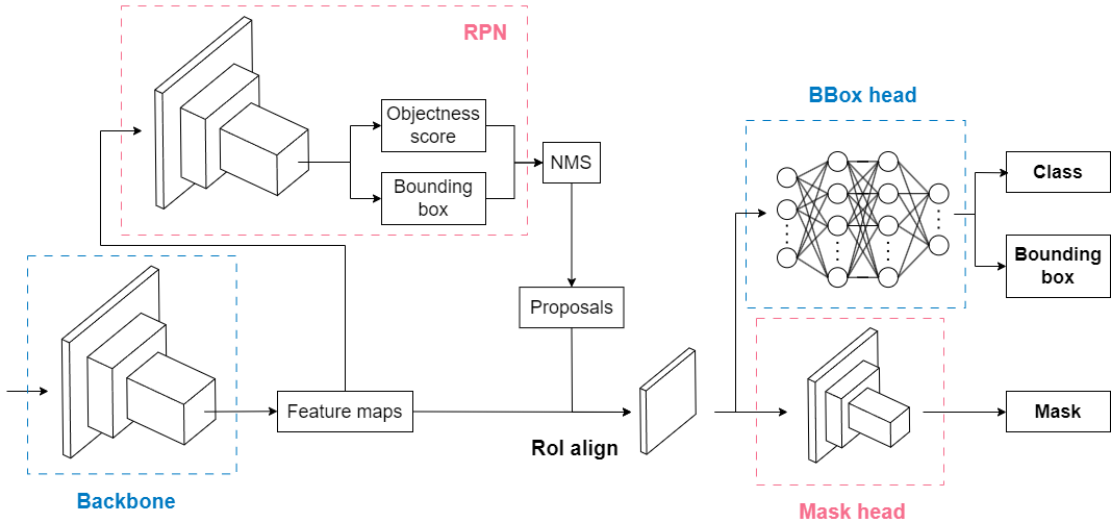

Figure 6.1 Instance segmentation architecture overview (Mask R-CNN[47]).

RPN operates by analyzing features extracted from a shared CNN backbone network, often pre-trained on large-scale datasets for feature learning. These features serve as a basis for generating region proposals. RPN consists of convolutional layers followed by a set of anchor boxes, which are pre-defined bounding boxes of different scales and aspect ratios.

By convolving feature maps from the shared CNN backbone with anchor boxes, RPN predicts two key elements for each anchor box: objectness scores (probability of containing an object) and bounding box regression offsets (adjustments to refine the anchor boxes to better fit objects). Subsequently, based on these predictions, RPN filters out anchor boxes that are less likely to contain objects and generates a set of high-confidence proposals that are further refined for subsequent segmentation.

Following proposal generation, Non-Maximum Suppression (NMS) refines predicted bounding boxes by prioritizing high-confidence detections and removing redundant ones. It sorts boxes based on their objectness scores, selects the highest-scoring box while suppressing overlapping ones based on a defined threshold (such as IoU), and retains only the best-scoring, non-overlapping boxes.[48]

6.1.2 Bounding box regression and classification

Post RPN, the proposals pass through a refinement phase where a secondary network refines the candidate bounding boxes to precisely fit the objects. This refinement step typically involves Region of Interest (RoI) pooling or RoI-align layers. These layers extract fixed-size feature maps for each proposed region from the backbone network's feature maps, aligning them to a standard size. This process ensures consistent inputs for subsequent stages regardless of the varying sizes and aspect ratios of the proposals.[47]

These RoI-aligned feature maps then feed into fully connected layers to perform two fundamental tasks: classification and bounding box regression. The classification task involves predicting the probability of an object class being present within each refined region, while bounding box regression aims to further adjust the proposed coordinates of the bounding boxes for better alignment with the true boundaries of the object.[47, 49]

The classification head of the network employs softmax or sigmoid activation functions to predict the likelihood scores for each object class. Simultaneously, the regression head estimates

adjustments (shifts in width, height, x, and y coordinates) to refine the initial bounding box proposals, aligning them more accurately with the actual object boundaries.[49]

6.1.3 Mask prediction

The mask prediction stage operates on refined bounding box proposals and employs a dedicated branch in the neural network architecture. This branch aims to generate pixel-wise masks corresponding to the identified objects. Initially, RoI (Region of Interest)-aligned features are extracted from the refined bounding boxes. These features encapsulate the object-specific information necessary for accurate mask generation.

Subsequently, these features pass through a series of convolutional layers, often coupled with upsampling or deconvolutional layers. These layers serve to increase the spatial resolution of the features, enabling the model to capture intricate details essential for accurately delineating object boundaries.

The final layer of the mask prediction branch typically involves activation functions, such as sigmoid or softmax, producing pixel-wise probabilities representing the likelihood of each pixel belonging to the foreground (object) or background. Thresholding is then applied to convert these probabilities into binary masks, segmenting the object from the background based on a predefined threshold.[47, 50]

6.2 Loss functions and metrics

Instance segmentation involves a triad of core tasks—classification, bounding box regression, and segmentation mask prediction—each demanding specific loss functions to measure errors during model training. The classification task focuses on assigning object categories to proposed regions, demanding the application of loss functions like Cross-Entropy to assess class probabilities against ground truth labels. Simultaneously, bounding box regression aims at refining and adjusting proposed bounding box coordinates, often utilizing Smooth L1 Loss to minimize deviations between predicted and actual bounding box positions. Finally, the mask prediction task concentrates on generating pixel-wise masks for precise object delineation, commonly optimized using Binary Cross-Entropy Loss for accurate mask prediction against the annotated ground truth masks.[47, 50, 51]

6.2.1 Cross-entropy loss

The cross-entropy loss function has two common variants: the binary cross-entropy and the categorical cross-entropy. he binary cross-entropy loss is typically used for binary classification tasks, wherein only two classes are present. Its formulation is defined as:

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})), \tag{6.1}$$

where y signifies the ground truth label (either 0 or 1), and $\hat{y}$ denotes the predicted probability of the sample belonging to class 1. In the context of instance segmentation, the binary cross-entropy loss is utilized for the objectness score within the RPN, which estimates whether there is an object in the proposed RoI or not[48]. Moreover, the binary cross-entropy loss is applied to the predicted masks. It computes the binary cross-entropy individually for each pixel in the mask, and the final loss is determined as the average of losses across all mask pixels.[47]

On the other hand, the categorical cross-entropy loss is defined as:

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{6.2}$$

where y_i represents the ground truth label for sample and the ith class (1 if the sample belongs to the class, 0 otherwise) and $\hat{y}_i$ is the predicted probability of the sample belonging to the ith class. The categorical cross-entropy loss is employed in the classification head, determining the class to which the object in the RoI belongs.[50]

6.2.2 Smooth L1 loss

The Smooth L1 loss function[49], also known as the Huber loss, is often used in tasks involving regression, such as bounding box prediction. It is a variation of the Mean Absolute Error (MAE) and Mean Squared Error (MSE) loss functions, addressing their limitations.

The formula for the Smooth L1 loss between a predicted value $\hat{y}$ and a ground truth value y is expressed as:

$$L(y, \hat{y}) = \begin{cases} 0.5 \times (y - \hat{y})^2 & \text{if } |y - \hat{y}| < \text{threshold} \\ |y - \hat{y}| - 0.5 \times \text{threshold} & \text{otherwise} \end{cases} \tag{6.3}$$

This loss function has a quadratic behavior when the absolute error is small (i.e., less than a certain threshold), similar to the MSE loss. However, when the absolute error is larger than the threshold, it linearly grows with the error, akin to the MAE loss. The "smoothness" of the transition between these two regimes is controlled by the threshold parameter.

In the context of instance segmentation, the Smooth L1 loss is used for both bounding box proposals generated by the RPN and bounding box predictions from the box regression branch. When comparing a ground truth bounding box $v = (v_x, v_y, v_w, v_h)$ and predicted bounding box $t = (t_x, t_y, t_w, t_h)$, where the vectors represent the top-left corner coordinates, width, and height of the bounding boxes respectively, the loss is defined as:

$$L(t, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i - v_i), \tag{6.4}$$

and the threshold of the smooth L1 loss is set to 1.

6.2.3 Mean average precision (mAP)

Mean Average Precision (mAP) is a widely used evaluation metric in object detection and instance segmentation tasks. It combines precision and recall to assess the accuracy of model predictions across multiple object categories or instances.

To calculate mAP, the precision-recall curve is employed for each class or object category. Precision signifies the ratio of correctly predicted positive instances to the total predicted positive instances, while recall measures the ratio of correctly predicted positive instances to the total ground truth positive instances. The precision-recall curve is plotted by varying the confidence threshold for prediction scores, resulting in different precision and recall values.

The interpolated precision-recall curve is derived by calculating the maximum precision for each level of recall. This technique involves interpolating the precision at each recall level by considering the maximum precision found for recalls greater than or equal to that level. The interpolated precision-recall curve allows for a smoother representation of model performance, aiding in better comparison between different models or thresholds.

The Average Precision (AP) for a specific class is computed by calculating the area under the interpolated precision-recall curve for that class. This area represents the average precision across all recall levels for that specific class. Finally, the mean of the AP scores for all classes in the dataset gives the mAP, reflecting the overall performance of the model across different object categories.

mAP provides a comprehensive evaluation of the accuracy of the model across multiple classes, balancing both precision and recall. Higher mAP values indicate better model performance in

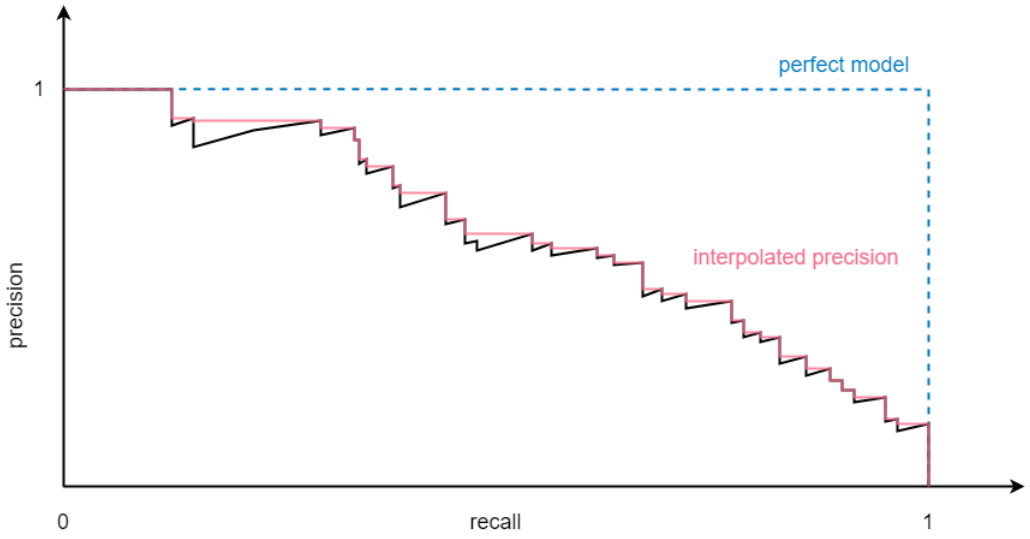

Figure 6.2 Precision-Recall curve (black), interpolated PR curve (pink), PR curve for a perfect model (blue).

accurately detecting and segmenting objects across different categories, making it a crucial metric in evaluating these models.[52]

6.3 Frameworks

Instance segmentation frameworks are filling the gap between object detection and precise pixel-level segmentation. These frameworks integrate object localization, classification, and pixel-level segmentation into a unified architecture, enhancing visual data analysis across diverse domains like medical imaging, autonomous driving, and robotics. Some of the commonly used frameworks are Mask R-CNN, Cascade Mask R-CNN, and Hybrid Task Cascade (HTC).

6.3.1 Mask R-CNN

Mask R-CNN, introduced in 2017 by He et al. [47], extends the Faster R-CNN object detection architecture [48]. Faster R-CNN has two stages: the Region Proposal Network (RPN) initially suggests candidate bounding boxes known as regions of interest (RoIs). Subsequently, the framework performs feature extraction, object classification, and bounding box regression for each RoI, producing outputs for object classes and bounding boxes.

In comparison to Faster R-CNN, Mask R-CNN introduces two architectural enhancements (refer to Figure 6.3). First, it incorporates an additional branch dedicated to semantic segmentation. This branch utilizes a small fully convolutional network (FCN) to perform semantic segmentation in parallel with the branch responsible for bounding box detection and classification. The semantic segmentation is performed across all classes for each RoI, and only the mask of the predicted class is obtained.

The original Fast R-CNN lacked pixel-to-pixel alignment between the network input and output, resulting in inaccurate masks for Mask R-CNN due to misalignments introduced by operations like RoIPool that extracts a small feature map from each RoI in the second stage. To address this issue, He et al. proposed RoIAlign, an operation that avoids spatial quantization, preserving the spatial locations of RoIs and correcting the misalignment problem in output masks

of Mask R-CNN.

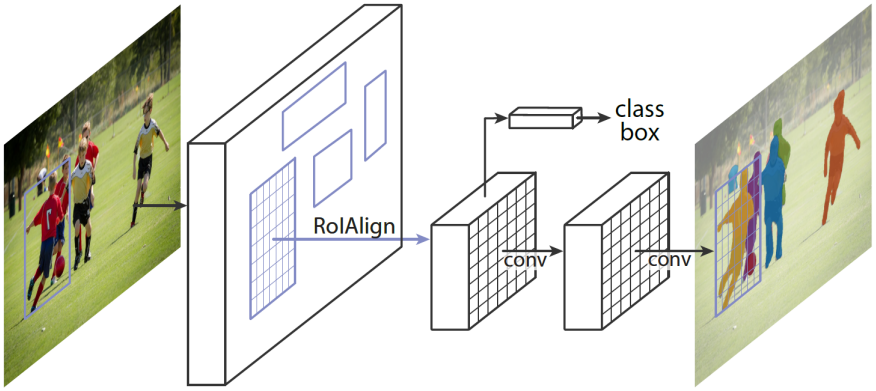

Figure 6.3 Mask R-CNN framework - Faster R-CNN with addition of the RoIAlign operation and a branch for semantic segmentation on RoIs.[47]

6.3.2 Cascade Mask R-CNN

Candidate bounding boxes in instance segmentation undergo classification as positives or negatives based on an intersection over union (IoU) threshold. Typically, a threshold of 0.5 is used during training, resulting in suboptimal detections. However, simply increasing the threshold does not enhance detections and often leads to poorer performance, known as the paradox of high-quality detection (refer to Figure 6.4). This issue arises due to the generation of numerous false positives by architectures, which are close to true positives but not entirely accurate [50].

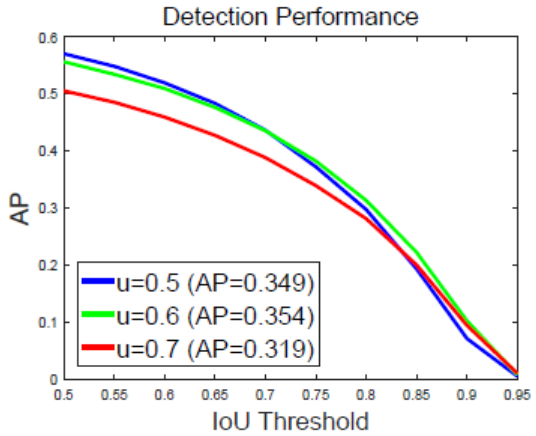

Figure 6.4 Decreasing AP for detectors trained with different IoU threshold u. [50]

The problem is caused by generating low-quality candidate boxes by the proposal algorithm. Increasing the IoU threshold reduces the number of positive training samples significantly, causing potential overfitting due to a smaller dataset. Moreover, detectors trained with high thresholds are effective only for high-quality proposals, while the actual proposals by RPN tend to be of lower quality, resulting in a degradation in detection performance [50].

To tackle this issue, Cai et al. introduced Cascade Mask R-CNN in 2019 [50], an extension of Mask R-CNN that utilizes multiple IoU thresholds. This framework employs a cascade of

detection heads, each with an increasing IoU threshold. For instance, a sequence of detectors could feature thresholds like [0.5, 0.6, 0.7]. Subsequent detectors refine the bounding boxes generated by earlier ones, while the semantic segmentation branch used for generating masks of instances is appended only to the last stage, illustrated in the framework diagram in Figure 6.5.

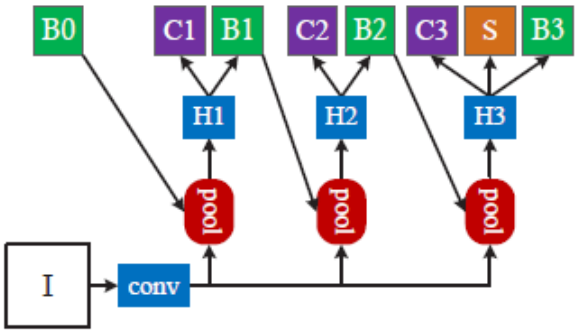

Figure 6.5 Architecture of Cascade R-CNN framework "I" is input image, "conv" backbone convolutions, pool" region-wise feature extraction, "H" network head, "B" bounding box, "S" segmentation branch and "C" classification. "B0" is proposals. [50]

This approach comes from two observations. detectors can produce high-quality detections when their quality matches the candidate bounding box inputs closely, and the output IoU of the bounding box regressor is typically superior to the input IoU, as demonstrated in Figure 6.6. Thus, improving bounding box quality allows higher threshold detectors without compromising predictions. Additionally, there is not a substantial reduction in the positive sample distribution size, and the higher-quality detectors demonstrate less susceptibility to overfitting.

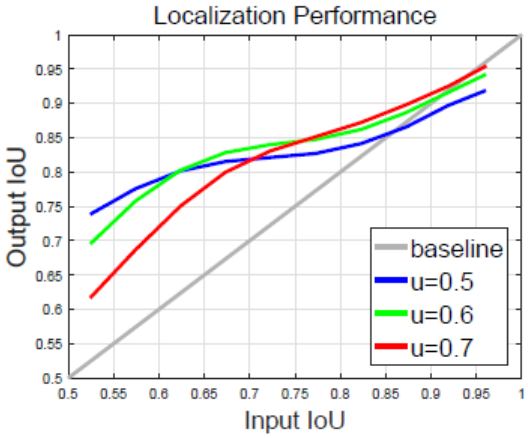

Figure 6.6 Output IoU of a regressor is almost always higher that the input IoU. [50]

It was shown by Cai et. al. that this framework outperforms Mask R-CNN across various tasks, including bounding box regression, classification, and segmentation mask quality, across multiple challenging datasets.

6.3.3 Hybrid Task Cascade (HTC)

Later in 2019, Chen et al. introduced a framework called Hybrid Task Cascade (HTC) as a response to Cascade Mask R-CNN. While Cascade Mask R-CNN displayed substantial im-

provement in bounding box regression (by 3.5% for bbox AP), the enhancement for instance segmentation was relatively modest (only 1.2% for AP). The identified issue was the limited flow of information between the mask branches across different stages. In Cascade Mask R-CNN, the mask branches solely benefit from improved bounding boxes without receiving any input from enhanced segmentation masks.

To address this limitation, they introduced the HTC framework, which integrates direct connections between mask branches. Additionally, they introduced a fully convolutional branch responsible for semantic segmentation of the input image. This branch extracts crucial features and information from the input image, and its output is integrated and shared with the mask branches, as depicted in Figure [51].

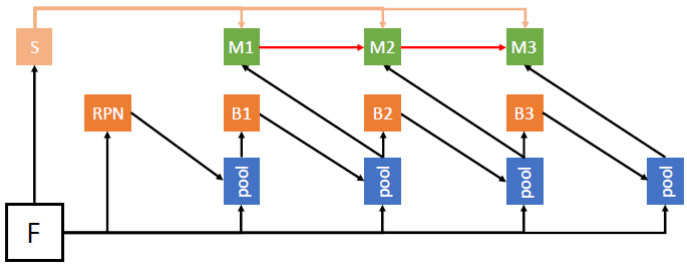

Figure 6.7 Hybrid Task Cascade (HTC) framework: "F" stands for input, "RPN" for region proposal network network, "pool" region-wise feature extraction, "B" bounding box regressor, "M" mask branch, and "S" semantic segmentation branch.[51]

Chapter 7
Dataset

The following chapter will provide a description and examples of the image data used in this thesis and their features, along with corresponding segmentation masks, and will discuss the annotation process. Additionally, it will cover the data pre-processing steps undertaken and describe the augmentations applied to the dataset.

7.1 Data description

For this thesis, a dataset comprising 419 microscopic images of pancreatic islets, along with corresponding ground truth annotations was graciously provided by the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM) in Prague, CZ.

These images were acquired between 2019 and 2022. The islets within the images are stained red, while the appearance of the exocrine tissue varies across images due to distinct microscope settings and lighting conditions, presenting in shades ranging from blue, green, white, to orange, and red. Additionally, each image is accompanied by metadata providing microscope magnification details, crucial for accurate islet size and volume calculations.

This dataset exhibits a high level of diversity, encompassing variations in lighting conditions, microscope settings, magnification levels, and the islet cultures themselves, which differ in islet count, sizes, and purity. This diversity is visually depicted in Figure 7.1. Notably, the images occasionally feature non-islet or non-exocrine tissue, and sometimes exhibit defects like bubbles, hair, or reflections.

Furthermore, each image in the dataset is associated with corresponding ground truth annotations, illustrated in Figure 7.2. These annotations consist of pixel-level segmentation masks, where pixel intensity values are utilized: ”255” signifies an islet pixel, ”128” denotes exocrine tissue, and ”0” represents the background.

Over a period of three years, several experts meticulously annotated these images. The annotation process involved manual segmentation using image processing software such as Gimp or ImageJ. In some cases, initial segmentation was performed in ImageJ using thresholding methods, and subsequent refinement was accomplished manually in Gimp to ensure precise delineation of islets and exocrine tissue. This hybrid annotation strategy was adopted to streamline the process while maintaining annotation accuracy.

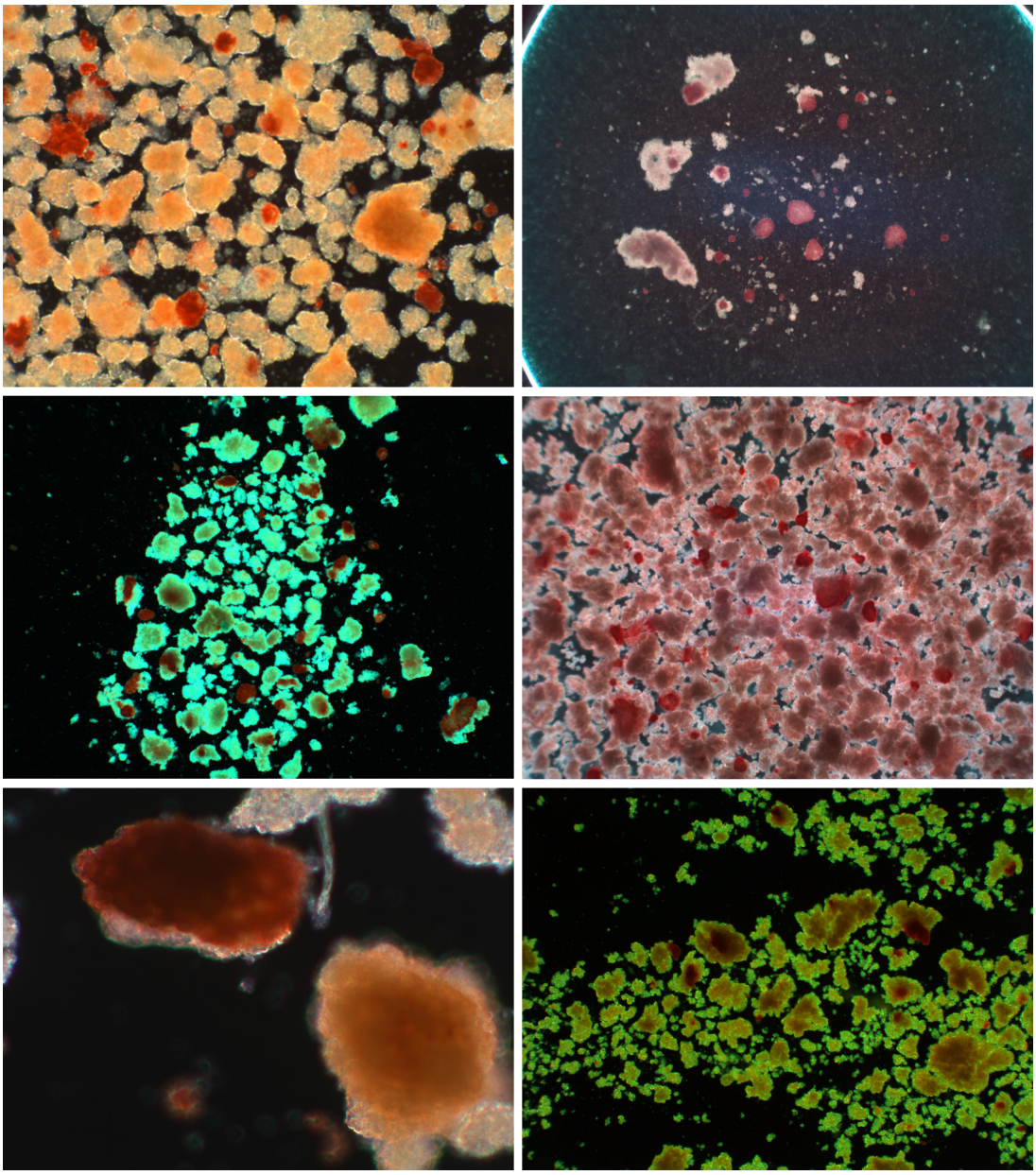

Figure 7.1 Diversity of microscopic images of pancreatic islets. Images are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ

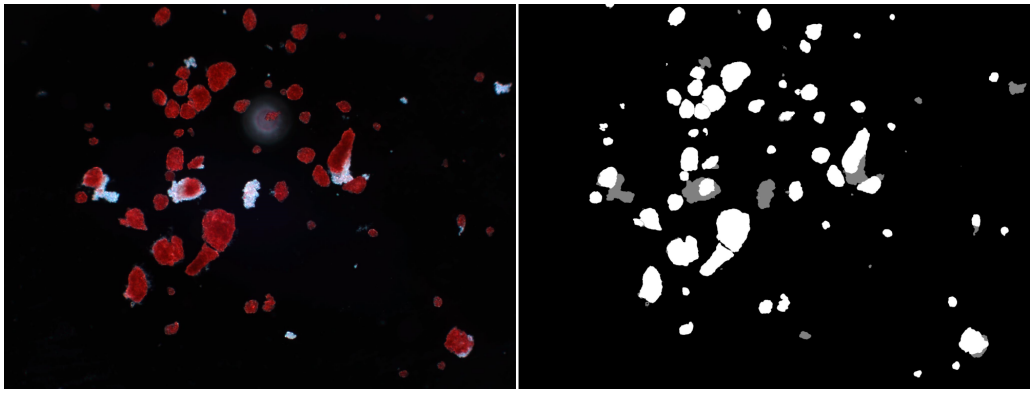

Figure 7.2 Right: microscopic image of pancreatic islets, **left**: ground truth annotation made by an expert. Images are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ

7.2 Data preprocessing

Initially, the dataset was separated into three subsets: training, validation, and test data, with a distribution ratio of 330:44:45, respectively. The selection process for the validation and test sets was performed in a way that preserves the variability of the dataset. Images were divided in different groups according to the following attributes:

- **Occurrence of adjacent islets in the image**: images with and without adjacent islets
- **Purity**: low purity ($< 40\%$), middle purity ($40 - 69\%$), and high purity ($\geq 70\%$)
- **Image scale**: 0.47, 1.22, 2.36, and 3.76 μm/px

The distribution of these groups, as depicted in 7.3 , highlights a notable imbalance among them. It is important to include all groups within the validation and test sets to ensure the network performs effectively across different attributes. Given the challenge of distinguishing islets encountered in previous and current approaches, it becomes crucial to have an adequate representation of images containing adjacent islets within the validation and test sets (at least $\geq 40\%$ of the images). Therefore, for the test and validation sets a unique subset of images from each group were chosen to include all of the groups in both sets.

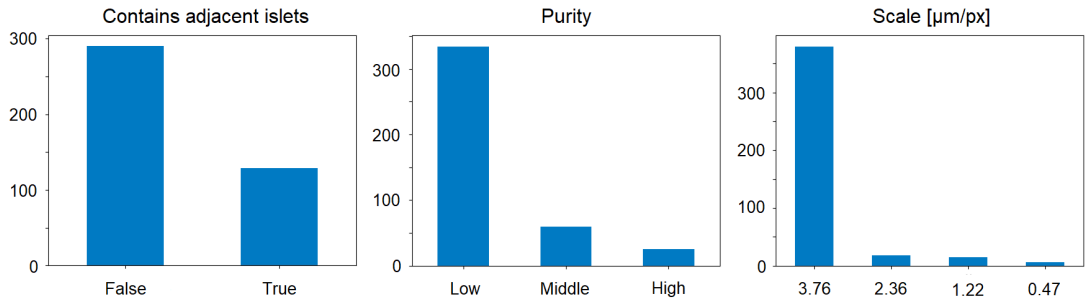

Figure 7.3 The representation of values for individual attributes in the dataset. **Left**: number of images without and with adjacent islets, **middle**: number of images with low, middle, and high purity, **right**: number of images with scale 3.76, 2.36, 1.22, and 0.47 μm/px.

After separating the dataset into training, validation, and test set, the dataset needed to be preprocessed for the training. The dataset was initially annotated for semantic segmentation,

which required specific modifications before its use in instance segmentation. First of all, data includes islets that are adjacent and there is no gap between them. Consequently, the segmentation masks for these adjacent islets are connected, causing two distinct islets to appear as a single instance in the segmentation mask. To ensure precise delineation of individual islet instances, an expert manually separated these instances.

The masks required conversion into the COCO annotation format[53], specifically designed for instance segmentation annotations. The COCO annotation format is a JSON file that encompasses annotations for the complete dataset, organized in a structure demonstrated in Code listing 7.1.

```
{
    'info': {
        'description': 'pancreatic-islets'
    }
    'images': [
        {
            'file_name': 'islet_sample_001.jpg',
            'height': 1536,
            'width': 2048,
            'id': 123
        },
        ...
    ],
    'annotations': [
        {
            'segmentation': [[735,
                1201,
                ...
                736,
                1201]],
            'area': 6599,
            'iscrowd': 0,
            'image_id': 123,
            'bbox': [675, 1201, 114, 96],
            'category_id': 0,
            'id': 24
        },
        ...
    ],

    'categories': [
        {'id': 0, 'name': 'islet'},
    ]
}
```

Code listing 7.1 COCO annotation format for instance segmentation. The "info" section provides an overview of the complete dataset, while "images" contains a list of all images. "Annotations" is a list of all annotations across the images, with "segmentation" referring to the XY coordinates of the annotation contour. Lastly, "categories" encompasses a list of all the distinct categories in the dataset.

For each image and individual islet, the contours and bounding boxes were determined using Python's OpenCV library[54]. The pixel area of each islet was computed, and subsequently, the annotations were transformed into the COCO format. However, an issue arose with the `cv2.findContours()` function, which utilized 8-way connectivity, leading to most adjacent islets being connected. To resolve this, the islets were initially separated into individual instances by using `cv2.connectedComponents()` with 4-way connectivity. Subsequently, contours were identified for each connected component, defining individual islets.

Despite this approach, a manual inspection revealed that the expert's initial separations did not accurately isolate all islets. Consequently, manual corrections were made to refine the separations and the COCO annotations were re-generated based on the corrected masks. After these adjustments, the data was prepared for the instance segmentation task.

7.3 Data augmentation

Due to the limited size of the dataset (containing only 419 images), additional augmentations were essential to reduce the risk of overfitting, as typically in instance segmentation tasks, thousands of images are used for training. Several standard augmentations were implemented, including various image transformations like rotation, stretching, flipping, and perspective transformations. These alterations modified the scale of the image, as well as the shape, size, or rotation of the islets. Moreover, color space transformations such as changes in saturation, brightness, and contrast were applied to simulate different lighting conditions and microscope settings.

During the model training, it was noted that certain defects were occasionally misclassified as islets by the neural network. These defects encompassed non-islet or non-exocrine tissue, bubbles, hair, streaks on glass, and reflections. To minimize misclassification, an augmentation technique was used to integrate defects into the images. Images containing these defects were obtained for this thesis together with the microscopic images and their annotations. These defect-containing images served as background images, onto which islets were added, as depicted in Figure 7.4.

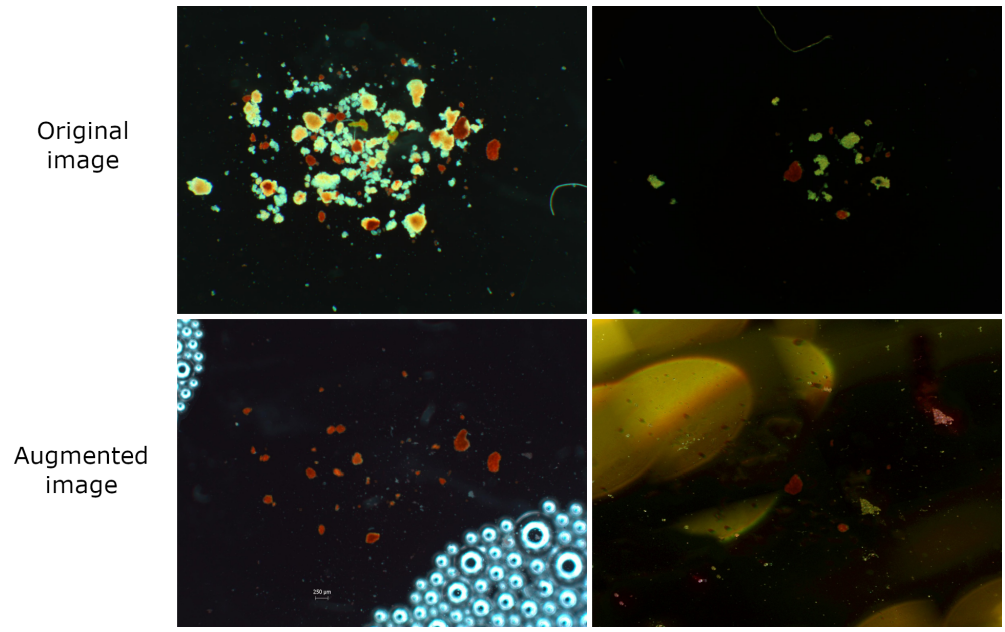

Figure 7.4 Defects addition augmentation - islets are segmented from the original image and place onto an image with a defect. **Left**: addition of bubbles, **right**: addition of reflections. Images are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ

Additionally, a specific augmentation was implemented to handle cases where models struggled to recognize small islets, leading to numerous false negative ground truth islets. This augmentation approach involved selecting a random subset of small islets, augmenting them through rotation, stretching, and perspective transformation, and then randomly placing these transformed islets onto background or exocrine tissue pixels in the image, as shown in Figure 7.5.

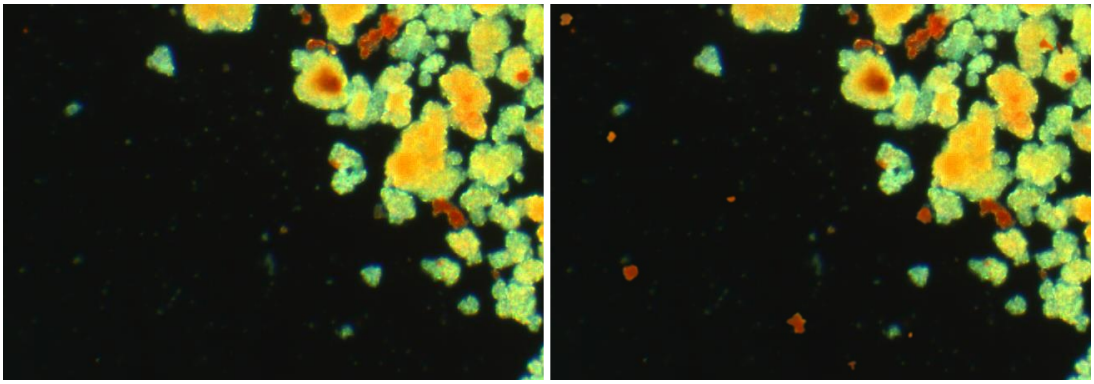

Figure 7.5 Small islets addition augmentation. **Left**: original image, **right**: image with small islets added. Images are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ

The effects of implementing the mentioned augmentations will be discussed in the subsequent chapter, alongside the outcomes of the conducted experiments.

Chapter 8

Experiments and results

In this chapter, a detailed description of the performed experiments will be provided. This will encompass a description of the training setup and evaluation metrics. Additionally, this chapter will explain the chosen algorithms and models for addressing the issue described in this thesis, along with the reasons behind their selection, as well as the process of training and parameter configuration. Furthermore, an evaluation of the quality and comparison of the trained models will be included. Finally, the best model will be chosen and used for the comparison with the state-of-the-art model in the next chapter.

The primary objective of these experiments is to develop a model capable of accurately identifying individual pancreatic islets within microscopic images-an aspect where previous methods have faced challenges. The central focus is on refining the Mask R-CNN framework with a ResNet50 backbone. These experiments involve fine-tuning the hyperparameters of the network and employing image augmentation techniques. Additionally, alternative backbones within the Mask R-CNN framework are assessed, alongside the evaluation of other instance segmentation frameworks.

8.1 Setup

The experiments and training pipeline were implemented through MMDetection v3.2.0[55], a PyTorch[56]-based framework that contains various modules for instance segmentation models including frameworks, backbones, as well as schedulers and optimizers tailored for training. The training was performed on a GPU cluster named ”galdor” within MetaCentrum (MetaVO)[57], utilizing 4x nVidia A40 GPUs.

MMDetection contains configurations covering multiple frameworks and backbones that were discussed in the preceding sections. These configurations serve as a basis for experimentation where certain parameters are fine-tuned for each specific trial outlined in the experimental section. The augmentations employed have been detailed in the previous chapter and have been implemented as modular components that can be readily imported into MMDetection for training purposes.

Throughout this chapter, all models are evaluated based on their predictions on the validation dataset comprising 44 images. The subsequent chapter will present a comparative analysis with the IsletNet model, leveraging the test dataset.

8.1.1 Evaluation metrics

The aim is to find a model that performs well on identifying individual islets, for that a special metric will be necessary. At first, GT and predicted islets are categorized into four distinct groups:

- **Matched islets**: GT and predicted islets overlap each other without any other islet interference.
- **Incorrectly separated islets**: GT (resp. predicted) islets overlap with multiple predicted (resp. GT) islets or intersect with only one islet, yet the intersected islet overlaps with multiple GT (resp. predicted) islets.
- **False positive islets**: Predicted islets without any intersection with GT islets.
- **False negative islets**: GT islets without any intersection with predicted islets.

This categorization is crucial for assessing the capability of the model to correctly separate islets. However, when two islets are erroneously separated, it does not provide information about the quality of the separation. Due to the absence of a suitable metric capable of evaluating the accuracy of such separations, a novel approach has been developed for this thesis. Given the inability of IsletNet model to separate adjacent islets, this approach focuses solely on them. Thus, a mask containing only adjacent islets is derived from each ground truth mask. The complete mask undergoes dilation using a 5 x 5 kernel, and any islets connected to others after dilation are recognized as adjacent and added into the mask of adjacent islets.

Subsequently, a custom metric is calculated within this mask. It identifies all predicted islets intersecting with each GT islet, calculates IoU for each GT-predicted pair, and selects the maximum IoU as the primary IoU for the respective GT islet. Then it constructs a curve (depicted in Figure 8.2), plotting IoU thresholds (ranging from 0 to 1) on the x-axis against the ratio of GT islets having a maximal IoU surpassing the IoU threshold on the y-axis. Adjacent islets that are matched or incorrectly separated by the model but exhibit closeness to the correct separation will display a higher IoU (and AUC) compared to those that are significantly misplaced or not separated at all, illustrating the concept in Figure 8.1. The area under the curve (AUC) of this curve will be referred as "adjGT-maxIoU AUC" and is calculated only for the adjacent islets obtained by the algorithm described above. The primary aim of this metric is to evaluate the models on the adjacent islets, which are problematic for the state-of-the-art approach in pancreatic islet segmentation, and to measure the quality of separating adjacent islets.

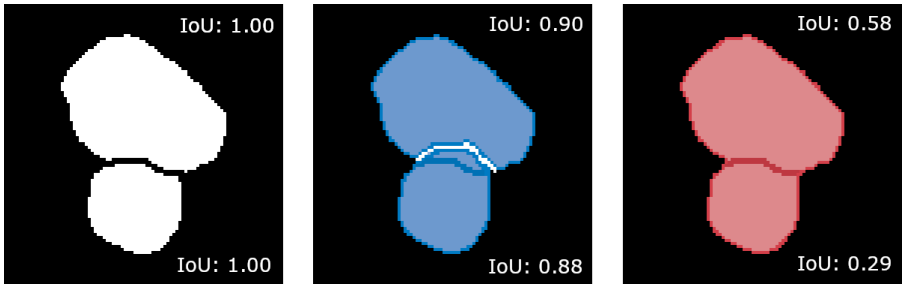

Figure 8.1 The maximum IoU is computed for both sets of islets. On the left, the GT islets reveal a perfect IoU of 1.0, indicating precise overlap. In the middle, although the GT islets are incorrectly separated, the prediction is close to the GT, resulting in a lower IoU, albeit not drastically low. On the right, the islets are completely unseparated, resulting in a considerably lower IoU.

Additionally, in comparison of the instance segmentation models, metrics as bounding box mAP and segmentation mAP are used. As the semantic model cannot be assessed using mAP, an

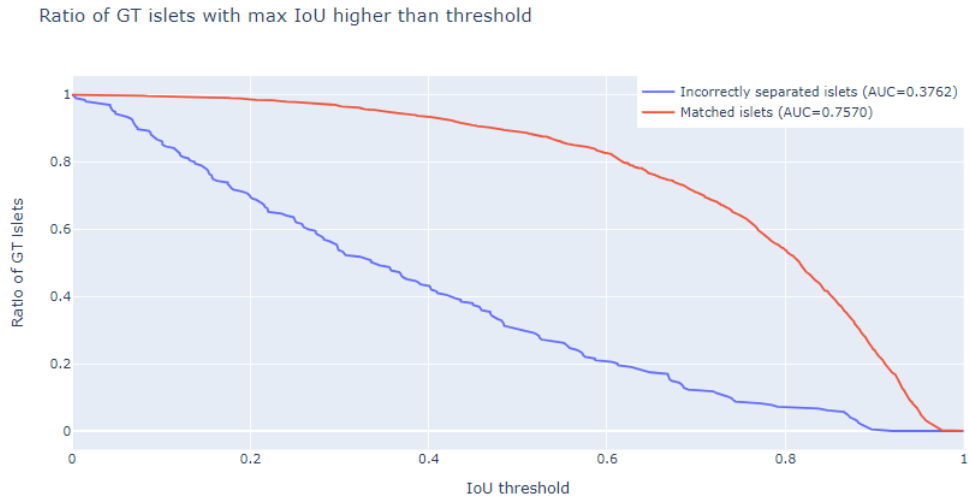

Figure 8.2 The GT ratio over IoU metric for the IsletNet model demonstrates a significantly higher area under the curve (AUC) for islets classified as matched, contrasting with incorrectly separated islets that exhibit notably lower IoU values.

alternative approach will monitor two additional metrics to characterize the segmentation quality of the semantic model. This ensures that instance segmentation models do not significantly lag in segmentation quality. This tracking involves two metrics: "IoU all," which computes the IoU metric on the entire semantic mask, and "IoU adj," which calculates the IoU metric solely on the semantic mask of adjacent islets.

To evaluate the instance segmentation models, the masks of all predicted islets are combined into a single semantic mask. This allows for the calculation of these metrics for the instance segmentation model. The IoU metrics will be represented as "mean ± standard deviation."

In summary, the key metrics for evaluation include the count of matched islets and the adjGT-maxIoU AUC, offering crucial insights into the ability of the model in correctly matching and accurately separating islets. Additionally, the count of false negative islets emerged as an important metric during the experiments. It was noted that instance segmentation models struggle with a significant number of false negatives, prompting efforts to mitigate this issue. The other metrics contribute to a deeper understanding of the behavior and segmentation quality of the model.

8.2 Perfomance of the IsletNet model

The IsletNet model, trained on the 330 training images, was evaluated on the validation set to provide a benchmark for the experiments with instance segmentation models. It achieved an IoU of 0.836 ± 0.071 on the entire mask and 0.834 ± 0.210 on the adjacent islets, indicative of commendable performance.

Following the application of the watershed transform on the masks of the IsletNet model, notable changes occurred. There was an apparent enhancement in the quality of islet separation, as evidenced by the improvement in the GT-maxIoU AUC, along with a slight reduction in the number of incorrectly separated islets, as showed in Table 8.1.

Table 8.1 Comparison of the performance of IsletNet model with and without Watershed transform

Watershed transform	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
False	1256	195	169	128	0.513
True	1261	189	169	129	0.6462

8.3 Initial model

The initial model, serving as the foundation for further experiments, is a Mask R-CNN with a ResNet50 backbone trained on a COCO-based dataset using schedule 1x. The settings for the default Mask R-CNN are as follows:

- **Backbone**: ResNet50 with 4 stages pretrained on the ImageNet dataset[58]
- **Anchor Generator in RPN**: Scales=[8], Ratios=[0.5, 1.0, 2.0]
- **Proposals**: RPN with NMS settings of 1000 proposals and IoU threshold of 0.7
- **ROI Head**: SingleRoIExtractor with RoIAlign of output size 7 for bounding box feature extraction
- **Mask Head**: FCN with 4 convolutional layers, and output of mask prediction for pancreatic islets
- **Bbox head**: 2 FC layers that are shared across all RoIs, and output of bounding box regression for pancreatic islets

The training process is epoch-based, set to run for a maximum of 12 epochs. Two learning rate scheduling methods are employed. First, a LinearLR schedule is used with a starting learning rate of 0.001, maintained for 500 iterations (not epochs). Second, a MultiStepLR schedule reduces the learning rate at specified milestones during the training. The milestones are set at epochs 8 and 11, where the learning rate decreases by a factor of 0.1. Stochastic Gradient Descent (SGD) is chosen as the optimizer with a learning rate of 0.02, momentum of 0.9, and weight decay of 0.0001.

The training set comprises 330 images sized 2048 x 1536. To prepare the data for training, the default dataloader applies random flip augmentation with a probability of 0.5 and utilizes a batch size of 2 during the training process.

8.4 Experiments

8.4.1 Initial model optimization

The initial model detailed above, having bbox mAP 0.479 and segmentation mAP 0.442, exhibited suboptimal performance, primarily due to a notable presence of false negative islets. About 415 (26.3%) GT islets were classified as false negatives (compared to 129 (8.1%) of FN of the IsletNet model). This problem was observed especially in smaller islets, where 78% of islets smaller than $50\mu m$ and 10% of islets between $50-100\mu m$ were false negative. On the other hand, only 29 (2.5%) islets were classified as false positive in comparison with the IsletNet model with 169 (10.5%) false positive islets. Furthermore, the model demonstrated the potential of instance segmentation in accurately delineating individual instances, as evidenced by the adjGT-maxIoU AUC (0.5869).

8.4.1.1 Hyper-parameter tuning

The presence of numerous false negative islets might be attributed to the limitation posed by having only one anchor scale. Having only one anchor scale can cause false negatives as it limits the ability of the model to detect objects of different sizes effectively. With a single scale, some objects might not fit well within the generated bounding boxes, leading to missed detections or incorrect identification. Using multiple anchor scales may help to improve the ability of the model to detect objects of various sizes, reducing false negatives, in this case for smaller objects. Subsequently, three models were trained: one with scales = $[4, 8]$, a second with scales = $[2, 4, 8]$, and a third with scales = $[1, 2, 4, 8]$.

All these new scale settings led to reductions in false negatives. Model with scales = $[4, 8]$ has 345 (21.8%) false negative GT islets, model with scales = $[2, 4, 8]$ 324 (20.5%), and model with scales = $[1, 2, 4, 8]$ 313 (19.8%) demonstrating the most promising performance. This configuration demonstrated superior performance across all metrics, with the exception of the false positive islets count, as outlined in Tables 8.2 and 8.3. Due to its overall better performance, it was selected as the base model for subsequent experiments.

Table 8.2 Comparison of initial Mask R-CNN model across various anchor scale settings.

Model	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
Initial	1068	96	**29**	415	0.5869
scales = $[4, 8]$	1141	93	36	345	0.5791
scales = $[2, 4, 8]$	1159	96	42	324	0.5953
scales = $[1, 2, 4, 8]$	**1176**	**90**	31	**313**	**0.5985**

Table 8.3 Assessment of the mAP and IoU metrics across the initial Mask R-CNN model, evaluating the impact of different anchor scale settings.

Model	bbox mAP	segm mAP	IoU all	IoU adj
Initial	0.479	0.442	0.814 ± 0.060	0.784 ± 0.213
scales = $[4, 8]$	0.507	0.458	0.807 ± 0.067	0.780 ± 0.196
scales = $[2, 4, 8]$	0.509	0.458	0.811 ± 0.068	$\mathbf{0.788 \pm 0.199}$
scales = $[1, 2, 4, 8]$	**0.516**	**0.463**	$\mathbf{0.822 \pm 0.051}$	0.766 ± 0.463

While modifying anchor scales provided some improvements, the issue of false negative islets persisted significantly. Another strategy employed to alleviate this problem involved increasing the number of proposals generated by the Region Proposal Network (RPN). By expanding the proposal count, the model could gain better coverage across potential object regions in images, which can enhances recall by capturing more potential instances, particularly smaller or less distinct objects that might otherwise be overlooked. By increasing the number of proposals by the RPN from 1000 to 1500, the amount of false negative islets decreased to 287 (18.2%). However, this adjustment also resulted in a slight rise in false positive islets, from 31 (2.4%) to 46 (3.5%), and decrease of the adjGT-maxIoU AUC from 0.5985 to 0.5881.

An alternative method to enhance the performance of the model involves conducting experiments with different training schedules and adjusting the learning rate. Due to the limited number of training images (330 images), training for only 12 epochs might not suffice for the network to comprehensively learn the features relevant to islet segmentation. Hence, an extended training schedule 2x (24 epochs) was employed. While most settings remained unchanged, distinct milestones (16 and 22) were incorporated to adjust the learning rate by a factor of 0.1. Furthermore, varying learning rate values (0.02, 0.01, 0.005, and 0.025) were utilized under

this setting. This extended schedule aimed to allow the network more time for learning islet segmentation features, potentially improving performance.

The comparison between the base model employing anchor scales of $[1, 2, 4, 8]$ and the various learning rates using the 2x training schedule, is showed in Tables 8.4 and 8.5. The outcomes indicate that the model with a learning rate of 0.0025 was insufficient demonstrating lower values of almost all metrics. Models trained with other learning rates demonstrated improved capabilities in minimizing false negative islets, enhancing the count of matched islets and improving the segmentation of adjacent islets as evident from higher adjGT-maxIoU AUC. Specifically, models trained with learning rates of 0.01 and 0.005 displayed the most promising results, yielding 288 and 291 false negative islets alongside 1200 and 1189 matched islets, respectively. Consequently, despite more false negatives, these models closely approached 1261 matched islets by IsletNet, indicating an improved ratio of correctly identified to incorrectly separated islets for instance segmentation models.

Table 8.4 Comparison of base model with anchor scales = 1, 2, 4, 8 alongside various learning rates with the 2x training schedule.

Model	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
Base model	1176	90	31	313	0.5985
lr = [0.02]	1189	**89**	46	301	**0.6260**
lr = [0.01]	**1200**	91	42	**288**	0.6251
lr = [0.005]	1189	99	38	291	0.6088
lr = [0.0025]	1156	101	**27**	322	0.5709

Table 8.5 Evaluation of the mAP and IoU metrics on the base model with anchor scales = 1, 2, 4, 8 using different learning rates using schedule 2x.

Model	bbox mAP	segm mAP	IoU all	IoU adj
Base model	0.516	0.463	0.822 ± 0.051	0.766 ± 0.244
lr = [0.02]	0.506	0.458	0.817 ± 0.060	0.779 ± 0.242
lr = [0.01]	0.520	0.470	$\mathbf{0.824 \pm 0.053}$	$\mathbf{0.798 \pm 0.188}$
lr = [0.005]	**0.524**	**0.474**	0.820 ± 0.055	0.782 ± 0.198
lr = [0.0025]	0.514	0.466	0.812 ± 0.053	0.746 ± 0.233

8.4.1.2 Data augmentations

An alternative strategy to enhance the predictions of a model trained on a limited dataset involves integrating image augmentations during the training phase. Several experiments employing augmentations outlined in the preceding chapter were conducted to enhance the performance of the model.

Initially, one experiment (model A1) aimed at addressing the high incidence of false negative predictions by introducing random small islets into images with a probability of 0.5. Subsequently, another experiment (model A2) integrated rotations (probability 0.2), perspective transforms (probability 0.2), image stretching (probability 0.1), and flips (probability 0.2) as augmentations. A subsequent experiment (model A3) incorporated background replacement with defects (probability 0.05), flips (probability 0.2), saturation changes (probability 0.1), brightness changes (probability 1), and contrast variations (probability 1). Lastly, a final experiment (model A4) merged the two previously mentioned experiments into a unified augmentation strategy comprising of mentioned augmentations except adding small islets to the image.

The augmentation that introduced extra islets into an image did not enhance the islet segmentation performance. Instead, there was a small increase of in the number of false negative islets from 313 (19.8%) to 320 (20.2%), accompanied by a reduction in the adjGT-maxIoU AUC as well as most of the other metrics. Furthermore, it caused a significant increase in the data preprocessing time from 2 minutes per epoch to 21 minutes. As a result, this augmentation method seems to offer no discernible advantages for enhancing the performance of the model in islet segmentation.

Likewise, as showed in Tables 8.6 and 8.7, the remaining augmentation experiments resulted in an increase in the count of false negative islets and a deterioration in the adjGT-maxIoU AUC and other evaluation metrics. However, it is important to note that only a limited number of experiments were conducted focusing various data augmentations and the experiments combined multiple augmentations at once. Therefore, future experiments should focus on individually evaluating the benefits of each augmentation separately to better understand their impact on the performance of the model.

Table 8.6 Comparison of base model with anchor scales = 1, 2, 4, 8 and different augmentation strategies.

Model	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
Base model	**1176**	90	**31**	**313**	**0.5985**
Model A1	1173	**86**	37	320	0.5965
Model A2	1154	95	37	330	0.5835
Model A3	1165	87	43	327	0.5883
Model A4	1134	91	44	354	0.5735

Table 8.7 Comparison of the mAP and IoU metrics on the base model with anchor scales = 1, 2, 4, 8 using various data augmentation strategies.

Model	bbox mAP	segm mAP	IoU all	IoU adj
Base model	**0.516**	**0.463**	$\mathbf{0.822 \pm 0.051}$	0.766 ± 0.244
Model A1	0.507	0.461	0.813 ± 0.057	$\mathbf{0.788 \pm 0.190}$
Model A2	0.507	0.460	0.812 ± 0.068	$\mathbf{0.788 \pm 0.190}$
Model A3	0.504	**0.463**	0.810 ± 0.069	0.767 ± 0.546
Model A4	0.507	0.461	0.806 ± 0.084	0.782 ± 0.200

8.4.2 Other backbones and frameworks

8.4.2.1 Mask R-CNN backbones

The subsequent experiments were performed using the base model configuration with scales = $[1, 2, 4, 8]$, incorporating different backbone architectures. Specifically, three distinct backbones were tested: ResNet101, ResNeXt101 featuring a cardinality of 32, both comprising 4 stages and pretrained on the ImageNet dataset. Additionally, the experimentation included the Swin-T transformer, characterized by an embedding dimension of 96 and a hierarchical structure consisting of four stages. Each stage was equipped with varying numbers of transformer layers (2, 2, 6, 2) and employed different counts of attention heads (3, 6, 12, 24). Similar to the previous backbones, the Swin-T was also pretrained on the ImageNet dataset.

Neither ResNet101 nor Swin-T showed performance improvements as evident from Tables 8.8 and 8.9. In fact, both models exhibited deteriorated results in islet segmentation. With

ResNet101, the count of false negative islets surged to 331, while the adjGT-maxIoU AUC slightly decreased, as well as the count of matched islets and other associated metrics worsened. Similarly, employing the Swin-T model led to an increase in false negative islets to 363, a significant decline in the adjGT-maxIoU AUC, a reduction in matched islets count, and overall deterioration in segmentation quality.

On the other hand, the ResNeXt101 backbone exhibited notable improvements in the performance of the model. It notably decreased the count of false negative islets to 283 (17.9%) while simultaneously increasing the number of matched islets and adjGT-maxIoU AUC. Additionally, it demonstrated enhanced results in both bbox and segmentation mAP.

Table 8.8 Comparison of various backbones employed in the Mask R-CNN framework.

Backbone	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
ResNet50	1176	**90**	31	313	0.5985
ResNet101	1158	**90**	39	331	0.5951
ResNeXt101	**1192**	104	54	**283**	**0.6069**
Swin-T	1094	122	**24**	363	0.5551

Table 8.9 Assessment of the mAP and IoU metrics across the initial Mask R-CNN framework, evaluating the impact of using different backbones.

Backbone	bbox mAP	segm mAP	IoU all	IoU adj
ResNet50	0.516	0.463	$\mathbf{0.822 \pm 0.051}$	0.766 ± 0.244
ResNet101	0.506	0.458	0.812 ± 0.068	$\mathbf{0.786 \pm 0.190}$
ResNeXt101	**0.521**	**0.470**	0.818 ± 0.075	0.773 ± 0.250
Swin-T	0.515	**0.470**	0.806 ± 0.065	0.777 ± 0.182

8.4.2.2 Cascade Mask R-CNN and HTC

Apart from more complex backbones, the results of the model can be improved by using more complex frameworks such as Cascade Mask R-CNN or HTC. Initially, Cascade Mask R-CNN was trained with the same ResNet50 backbone utilized in the initial model. This model architecture contains an RPN utilizing anchor generators with scales set to [1, 2, 4, 8], ratios as [0.5, 1.0, 2.0], and employing NMS settings of 1000 proposals and an IoU threshold of 0.7 for region proposal. It incorporates a Cascade RoI Head with three stages, each equipped with a shared 2-layer fully connected (FC) bounding box head. Noteworthy is the use of different loss weights [1, 0.5, 0.25] across cascade stages, aiding in the refinement of bounding box predictions at each stage of the cascade process.

The HTC framework employs the same ResNet50 backbone alongside the RPN and bbox head configurations. Additionally, the model is equipped with an HTCMaskHead for mask prediction, structured across three stages, each consisting of four convolutional layers, with mask sizes set to 28. Furthermore, the architecture integrates a semantic head composed of four convolutional layers to extract semantic segmentation mask from the input images that is subsequently passed to the all the stages of the HTCMaskHead.

The HTC framework managed to achieve the highest bbox mAP (0.537) and mask mAP (0.48). However, while these metrics exhibited considerable improvement, other crucial aspects like the number of false negatives and matched islets presented noticeably worse results compared to the Mask R-CNN model. Similarly, employing Cascade Mask R-CNN did not yield significant performance enhancements. Additionally, training these more complex frameworks—Cascade

Mask R-CNN and HTC—required more time compared to the Mask R-CNN framework, with Cascade Mask R-CNN taking 1.2 times longer and HTC requiring 1.3 times more time for training.

It is important to note that the less favorable results obtained from these different frameworks do not necessarily imply their unsuitability for addressing the pancreatic islet separation problem. Conducting numerous experiments aimed at fine-tuning various hyperparameters within these frameworks is essential to decisively evaluate their effectiveness in pancreatic islet separation. However, due to scope limitations in this thesis, this optimization was not pursued, leaving an opportunity for further research in the future.

Table 8.10 Comparison of various instance segmentation frameworks.

Framework	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
Mask R-CNN	**1176**	**90**	31	**313**	**0.5985**
Cascade Mask R-CNN	1098	**90**	**21**	391	0.5876
HTC	1107	97	24	375	0.5830

Table 8.11 Assessment of the mAP and IoU metrics across various instance segmentation frameworks.

Framework	bbox mAP	segm mAP	IoU all	IoU adj
Mask R-CNN	0.516	0.463	$\mathbf{0.822 \pm 0.051}$	0.766 ± 0.244
Cascade Mask R-CNN	0.517	0.461	0.813 ± 0.070	$\mathbf{0.770 \pm 0.240}$
HTC	**0.537**	**0.480**	0.811 ± 0.070	0.761 ± 0.258

8.4.3 Selection of the best model

The model chosen as the basis for experimentation was the Mask R-CNN framework, integrating the ResNet50 backbone with the initial configurations outlined in preceding sections, albeit with a specific modification - adjusting anchor scales to a value of scales $= [1, 2, 4, 8]$. Although no other model emerged as superior across all metrics, several showcased commendable performance in pivotal metrics vital for this task, such as the count of correctly matched islets and adjGT-maxIoU AUC. Considering the challenge posed by a high count of false negative islets, the reduction of these inaccuracies holds significant importance. The comparison of the top-performing four models is delineated in Table 8.12.

Table 8.12 Comparison of the four top-performing models.

Model	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
Base model	1176	**90**	**31**	313	0.5985
RPN proposals = 1500	1194	98	46	287	0.5881
LR = [0.01]	**1200**	91	42	288	**0.6251**
ResNeXt101	1192	104	54	**283**	0.6069

An additional experiment was conducted by combining the features of the best-performing models aimed to leverage their effective traits. Three additional networks were trained, all using the Mask R-CNN framework with the ResNeXt101 backbone and the anchor scales setting

scales = $[1, 2, 4, 8]$. The initial model (referenced as model RNX 1) employed the schedule 2x with a learning rate of 0.01 and had 1500 RPN proposals. The second model (model RNX 2) employed only the schedule 2x with a learning rate of 0.01. Lastly, the third model (RNX 3) solely adjusted the number of RPN proposals to 1500.

Table 8.13 Comparison of the models combining features of the four top-performing models.

Model	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
RNX 1	1221	107	67	**251**	**0.6397**
RNX 2	**1236**	**92**	61	**251**	0.6283
RNX 3	1212	101	**46**	266	0.6059

Upon comparison of the outcomes presented in both Tables 8.12 and 8.13, the RNX 2 model emerges as the most promising. It excels particularly in the count of matched islets and false negative islets, demonstrating remarkable performance. While it slightly increases the count of incorrectly separated islets to 92, this figure remains close to the best result achieved in this metric (90 islets). Although it ranks as the second highest in the count of false positive islets (61 islets), this increase is relatively less impactful considering the substantial reduction in false negative islets compared to the IsletNet model, which recorded 169 FP islets. Moreover, while there is a slight decline in adjGT-maxIoU AUC, the better ratio of matched, incorrectly separated, false positive, and false negative holds greater significance.

Before directly comparing this model to the IsletNet model, a final assessment was conducted. The bbox head generates a score with each bounding box, typically set to a default threshold of 0.5, filtering out instances with scores below this threshold. Lowering this score threshold accepts more instances, leading to a decrease in false negative islets and an increase in false positive islets count. The IsletNet model recorded significantly higher false positive islets (169 islets) compared to RNX 2 (61 islets), while the false negative islets of the IsletNet model were notably lower (129 islets) compared to RNX 2 (251 islets). This knowledge suggests the possibility of reducing the threshold to minimize the count of FN islets while increasing FP, yet ensuring that the count of FP islets remains lower than that of the IsletNet model. Therefore, an analysis of different thresholds was conducted and is presented in Table 8.14.

Table 8.14 Comparison of various bbox score thresholds and their influence on islets segmentation results.

Threshold	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
T 0.5	1236	**92**	**61**	251	0.6283
T 0.4	1261	96	80	222	0.6374
T 0.3	1284	96	102	199	0.6453
T 0.2	1307	98	139	174	0.6477
T 0.15	1322	101	160	156	0.6494
T 0.1	**1343**	103	184	**133**	**0.6567**

For the final evaluation, the model with bbox score threshold of 0.15 was selected. This choice ensured that the count of false positive islets remained lower compared to the IsletNet model, while significantly reducing the number of false negative islets.

Chapter 9

Comparison with the state-of-the-art model

This chapter will show a comparative analysis of the finest model proposed in this thesis and the current state-of-the-art approach, the IsletNet model.

In the final comparison, both the state-of-the-art model IsletNet and the model with the best performance in the experimental section were evaluated using the test set comprising 45 images. The evaluation metrics employed for these models remain consistent with those utilized in the experimental section, incorporating an additional IoU metric to compare the semantic masks produced by the two models.

The findings presented in Tables 9.1 and 9.2 highlight the promising outcomes achieved by the proposed instance segmentation model in discerning individual instances of pancreatic islets. Although the proposed model detected slightly fewer GT islets (1865 of 2036 GT islets) compared to the IsletNet model (1894 of 2036 GT islets), it successfully matched 91.2% of the identified islets, in contrast to the IsletNet model with Watershed transform that matched 86.3% of the GT islets. Therefore, despite encountering difficulty in identifying all GT islets, resulting in a higher count of false negative islets (181 compared to 142 from IsletNet), the proposed network identified the islet instances more precisely than applying Watershed transform on segmentation masks from the IsletNet model. Additionally, the proposed model demonstrates a substantially higher adjGT-maxIoU AUC compared to the IsletNet model with Watershed. This evidence reinforces the assertion that the proposed model excels in accurately segregating adjacent islets while maintaining comparable semantic segmentation quality, as reflected by the IoU metrics. Examples of the difference in the ability to distinguish individual islets by both networks are visually compared in Figures 9.1 and 9.2.

Table 9.1 Comparison of IsletNet model with Watershed transform and the proposed instance segmentation model.

Model	Matched islets	Incorrectly separated islets	FP islets	FN islets	adjGT-maxIoU AUC
IsletNet	1649	245	242	**142**	0.4830
IsletNet + Watershed	1635	259	248	**142**	0.6166
Proposed model	**1702**	**153**	**177**	181	**0.6549**

Table 9.2 Comparison of IsletNet model with Watershed transform and the proposed instance segmentation model using IoU metrics.

Model	IoU all	IoU adj
IsletNet	**0.833 ± 0.085**	**0.803 ± 0.206**
IsletNet + Watershed	**0.833 ± 0.085**	**0.803 ± 0.206**
Proposed model	0.825 ± 0.065	0.779 ± 0.198

Nevertheless, although the proposed model exhibited improvements in distinguishing individual islet instances, it is noteworthy that the count of false negative islets remains relatively high. This is particularly noticeable in the case of smaller islets, where 27.4% of those under $50\mu m$ and 9% within the range of $50 - 100\mu m$ were classified as false negatives. To address this issue, further experiments are necessary. One potential approach could involve modifying the loss function to add more weight to the segmentation of smaller islets, potentially improving their detection and reducing false negatives.

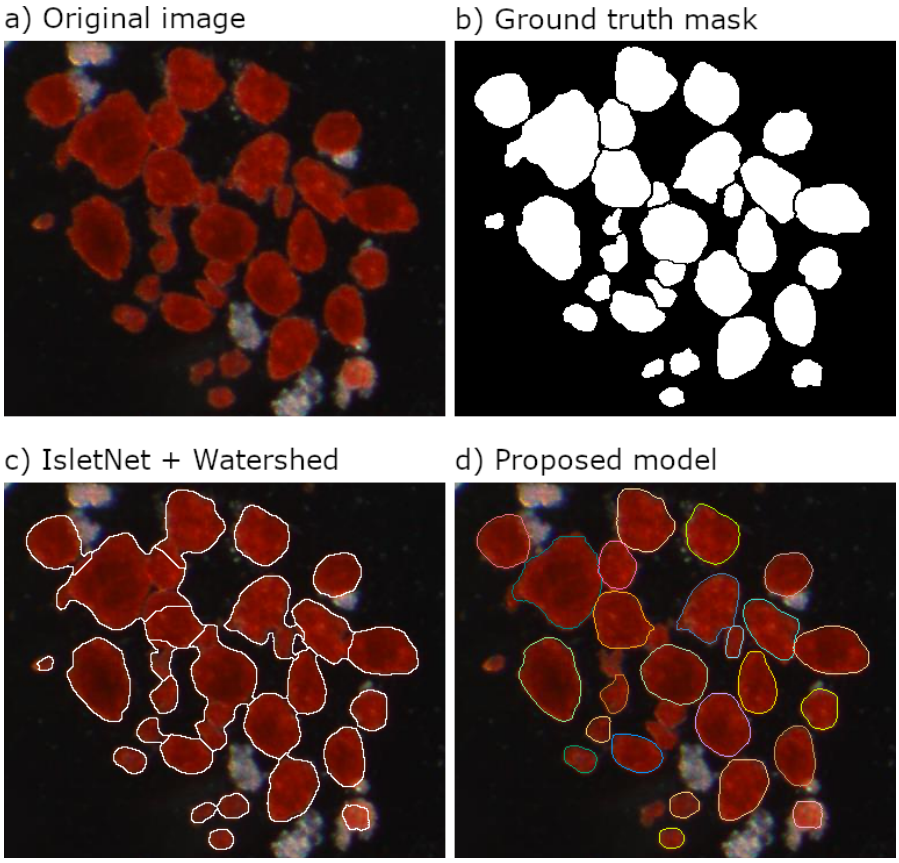

Figure 9.1 The comparison between the islet contours identified by the IsletNet model, processed with the Watershed transform, and those delineated by the proposed model shows improved islet separation with the proposed model. However, the proposed model presents a higher count of false negative islets. Original image and GT mask are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ.

Additionally, it is important to highlight that while both networks exhibited comparable IoU across the images, the proposed instance segmentation model encountered challenges in accurately segmenting the islet boundaries. This discrepancy might not significantly lower the

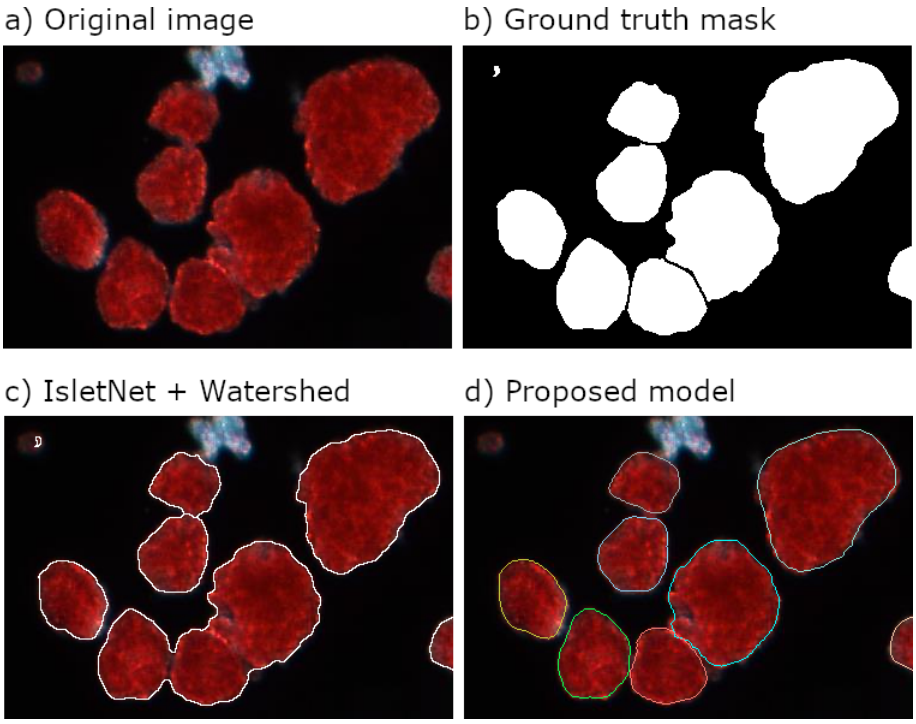

Figure 9.2 Comparison between the islet contours identified by the IsletNet model with Watershed transform and those detected by the proposed model highlights a significant difference in identifying adjacent islets. The IsletNet model, even after applying the Watershed transform, failed to separate the three adjacent islets. In contrast, the proposed model effectively distinguished and clearly separated these islet instances. Original image and GT mask are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ.

IoU but is noticeable upon visual examination of the segmentation, as illustrated in Figure 9.3. This limitation could be attributed to the relatively small mask segmentation head, which comprises only 4 convolutional layers, in stark contrast to the 18-layered IsletNet. Enhancing the accuracy in segmenting the islet boundaries could potentially be achieved by employing a more complex convolutional network within the mask head.

Moreover, the experiments primarily addressed a prominent issue observed in current approaches, namely the challenge in accurately segmenting individual islets when they are adjacent. As a result, the focus did not extend to segmenting the exocrine tissue. Therefore, an useful enhancement would involve training the network to discern exocrine tissue in addition to pancreatic islets.

In conclusion, the objective set for the experiments, which aimed to develop a model capable of more effectively segmenting adjacent islets, has been achieved. Nonetheless, the current state-of-the-art IsletNet model, employing the watershed transform, remains irreplaceable with the new approach due to three primary limitations observed in the proposed model. These limitations encompass imprecise segmentation of islet boundaries, a higher count of false negative islets, and the incapability to segment exocrine tissue. Addressing these limitations would require additional research and further experimentation.

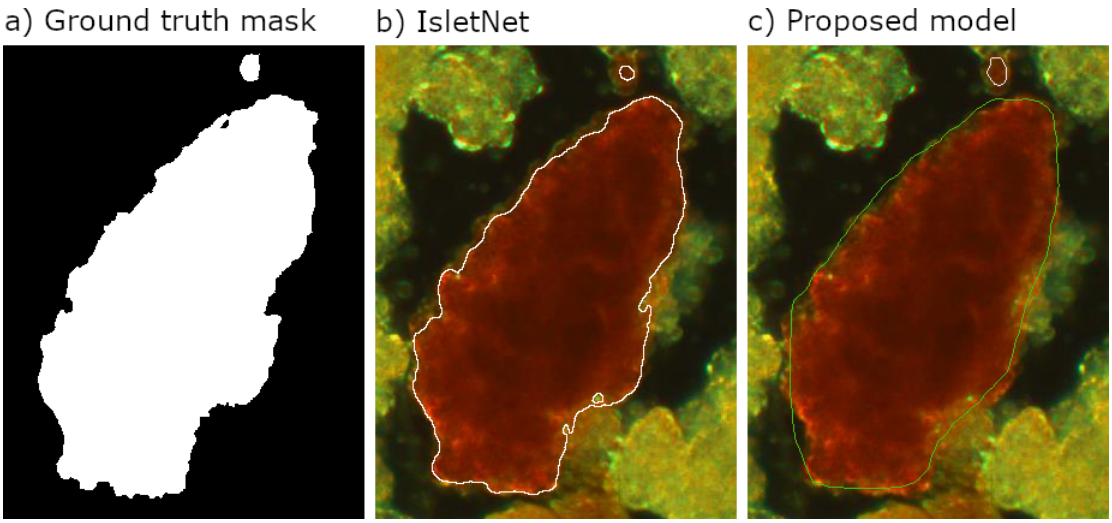

Figure 9.3 Comparison of the delineation of the islet borders made by IsletNet and the proposed model. Original image and GT mask are from the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ.

Discussion

The experiments based on the default configuration of the Mask R-CNN framework with a ResNet50 backbone, albeit with adjustments in the number of object classes and input resolution. However, this configuration is designed for instance segmentation tasks on the COCO dataset, characterized by 80 object classes and an thousands of training images. In contrast, the dataset employed in this thesis comprises only two classes (pancreatic islets and exocrine tissue) and a significantly smaller set of 330 images. Consequently, the configuration of the network, including the training schedule, might not be optimally suited for this dataset. Therefore, conducting further experiments and fine-tuning different parameters is necessary to evaluate their impact on the performance of the model.

Moreover, employing cross-validation could provide a more accurate estimation of the effects of various configurations. Nevertheless, this approach can be exceedingly time-consuming, particularly when dealing with high-resolution image data. Due to this time constraint, cross-validation was not utilized in this thesis. The primary objective was to explore whether instance segmentation models have the potential to outperform the state-of-the-art semantic segmentation model with watershed transform applied to its predicted masks. Hence, the approach focused on swiftly exploring the influence and potential of diverse configurations, backbones, and frameworks. The intent was to lay the groundwork for further research to conduct more sophisticated and novel experiments building upon these initial explorations.

The initial experiments revealed that instance segmentation models encounter difficulties in accurately identifying small objects, resulting in numerous false negative ground truth islets. Consequently, subsequent experiments were primarily aimed at diminishing the count of false negative islets while sustaining or enhancing the ability to correctly separate individual islet instances. Certain experiments displayed advancements, notably by extending the range of anchor generator scales in the Region Proposal Network (RPN), increasing the quantity of RPN proposals, extending the number of training epochs alongside with a reduction in the learning rate, and adopting a more complex backbone architecture - ResNeXt101.

Conversely, certain experiments did not yield improvements in the performance for the model. Specifically, attempts involving image augmentations, a transformer backbone, and the utilization of more complex frameworks like Cascade Mask R-CNN and HTC did not exhibit noticeable enhancements. The probable cause may not lie in the unsuitability of these approaches for this problem, but rather in the absence of comprehensive experiments examining the influence of each augmentation or the lack of hyperparameter tuning for alternative backbones and frameworks. Conducting such in-depth experiments would require a considerable amount of time, which falls beyond the scope of this thesis. Subsequent studies could concentrate on evaluating the specific influence of individual augmentations and refining the optimization of the more sophisticated backbones and frameworks.

The final experiments combined the features that enhanced the performance of the model.

The top-performing model was a Mask R-CNN framework incorporating ResNeXt101, anchor scales of [1, 2, 4, 8], and schedule 2x with a learning rate set at 0.01. This model underwent assessment by varying bbox score thresholds, and the most optimal threshold was selected based on a criterion: it ensured a lower count of false positive islets compared to IsletNet and minimized the count of false negative islets. Ultimately, the threshold of 0.015 was identified as the most suitable configuration, and it was chosen for the final comparison with the state-of-the-art model.

In comparison to the IsletNet model utilizing the watershed transform, the proposed instance segmentation model exhibited superior performance in correctly delineating individual islets. Notably, the IsletNet model failed to accurately separate 12.7% of the ground truth islets, whereas the proposed model reduced this error significantly to 7.5%. However, while showcasing promising advancements, the proposed model cannot entirely replace the IsletNet model due to its existing limitations. These limitations encompass a notable count of false negative islets (181 for the proposed model compared to 142 for the IsletNet model), less precise segmentation masks along the boundaries of the islets, and the absence of training for the segmentation of exocrine tissue alongside pancreatic islets.

The limitations of the proposed model can be eliminated by further experiments. Issue with high number of false negative islets could be improved by introducing new loss weighted function that will add more weight to the small islets making them more important for the network training. Additionally, further experiments with augmentation, more hyperparameter tuning of Mask R-CNN framework utilizing other backbones or fine-tuning other instance segmentation also can have impact on the number of false negative islets. The accuracy of the islet masks could be improved by implementing deeper, more complex CNN in the mask segmentation head.

Training the network to delineate exocrine tissue presents challenges due to its lack of formation into individual instances. The COCO format for instance segmentation typically supports masks accommodating multiple instances, which could be explored for representing exocrine tissue. Alternatively, implementing and experimenting with panoptic segmentation models might offer a solution. These models are specifically engineered to distinguish between objects that possess individual instances (such as pancreatic islets) and elements that do not exhibit individual instance characteristics (like exocrine tissue).

Lastly, it is noteworthy that annotating images poses a significant challenge due to its time-consuming nature, resulting in a limited number of available training images. Furthermore, variations in annotations can arise not only between different experts but also within annotations made by a single expert over time. This discrepancy is especially evident in the task of delineating individual islets. While some separations are clear and straightforward to annotate, in many instances, discerning whether the structures in the microscopic image represent one or two islets is not always evident. This ambiguity complicates the task for annotators to achieve perfect and consistent annotations. Consequently, this discrepancy can hinder the development of a network trained to perform exceptionally well on the dataset. Inconsistencies in annotating whether islets are separated in the same dataset can vary, impacting the ability of the network to identify consistent patterns due to differing annotations.

Conclusion

The primary objective of this thesis is the segmentation of pancreatic islets from microscopic images. The goal was to analyze the current solution, IsletNet, that uses the UNet semantic segmentation model, identify its weaknesses, and propose and implement an improved model to addresses these shortcomings.

In the theoretical part (Chapters 1-6), the thesis describes the theoretical aspects related to pancreatic islet transplantation, emphasizing the significance of automating the segmentation of islets from microscopic images and subsequently computing their parameters, particularly volume estimation. Furthermore, it examines the current state of research in automating the segmentation process and describes one of the most promising solution, IsletNet. The theoretical part also delves into the principles and architectures of convolutional neural networks and transformers employed for image processing tasks. Finally, the semantic architecture UNet and the most commonly used frameworks and architectures of instance segmentation are described.

The practical part (Chapters 7-9) begins with the description, analysis and preprocessing of microscopic images of pancreatic islets, provided by the Laboratory for the Islets of Langerhans, Experimental Medicine Centre (EMC), Institute for Clinical and Experimental Medicine (IKEM), Prague, CZ. It also provides details of the implementation of instance segmentation models using the MMDetection framework, along with description of the conducted experiments and their evaluation using the validation dataset. The concluding part involves comparing the best-performing model and IsletNet including suggestions for a further improvement of the proposed model.

After the IsletNet model analysis, it turned out that the most significant limitation was the inability of the model to separate individual islet instances, resulting into connecting several adjacent islets into a single islet. Consequently, this led to an overestimation of islet volumes, a critical factor in evaluating the quality of islet samples. In response to this issue, the proposed solution involves the implementation of an instance segmentation model. These models are designed to differentiate between individual instances, generating separate segmentation masks for each instance rather than creating one binary mask for the entire input image.

Through a series of experiments, the Mask R-CNN architecture with a ResNet50 backbone was implemented, and various hyper-parameters were fine-tuned. The most notable enhancement over the default settings of Mask R-CNN within the MMDetection framework was achieved by expanding the range of anchor generator scales in the Region Proposal Network (RPN) head. This adjustment notably improved the recall of the predictions of the model. Additionally, experiments with different backbones of the Mask R-CNN framework were performed as well as the application of other more complex frameworks such as Cascade mask R-CNN and HTC.

The comparison between the best-performing model and IsletNet revealed that Mask R-CNN outperforms IsletNet with watershed transform in distinguishing individual islet instances. On the test set of 45 images and 2036 pancreatic islets in total, IsletNet with the watershed transform,

demonstrated a rate of incorrectly separated islets of 12.7% among the ground truth islets. In contrast, Mask R-CNN exhibited a lower rate of incorrectly separated islets of 7.5% among the ground truth islets, while maintaining a similar semantic segmentation quality, with an IoU score of 0.825 ± 0.065, slightly below the IoU score of 0.833 ± 0.085 of the IsletNet.

Nevertheless, the proposed model is not yet capable of entirely replacing the IsletNet model due to three limitations: a higher count of false negative islets (8.9% of the GT islets compared to IsletNet's 7%), less precise segmentation of islet boundaries, and the lack of training of the model to recognize exocrine tissue.

However, unexplored strategies not incorporated in this thesis could enhance the instance segmentation model. Improving mask precision might involve implementing a deeper and more complex mask head. Addressing the false negative issue might require modifications in the loss function, such as assigning more weight to smaller islets. Exploring more sophisticated frameworks and backbones could also assist in mitigating this problem. Additionally, training the model to segment exocrine tissue might involve experimenting with converting the mask for an instance segmentation task or implementing a panoptic segmentation approach. These approaches could potentially lead to a further advancements in refining the proposed model.

Bibliography

1. GREEN, Anders et al. Type 1 diabetes in 2017: global estimates of incident and prevalent cases in children and adults. *Diabetologia*. 2021, vol. 64, no. 12, pp. 2741–2750. ISSN 1432-0428. Available from DOI: 10.1007/s00125-021-05571-8.
2. DANEMAN, Denis. Type 1 diabetes. *The Lancet*. 2006, vol. 367, no. 9513, pp. 847–858. ISSN 0140-6736. Available from DOI: 10.1016/s0140-6736(06)68341-4.
3. RICORDI, Camillo. Lilly Lecture 2002. *Diabetes*. 2003, vol. 52, no. 7, pp. 1595–1603. ISSN 1939-327X. Available from DOI: 10.2337/diabetes.52.7.1595.
4. ŠVIHLÍK, Jan et al. Classification of microscopy images of Langerhans islets. In: OURSELIN, Sebastien; STYNER, Martin A. (eds.). *SPIE Proceedings*. SPIE, 2014. Available from DOI: 10.1117/12.2043621.
5. RICORDI, Camillo. Quantitative and Qualitative Standards for Islet Isolation Assessment in Humans and Large Mammals. *Pancreas*. 1991, vol. 6, no. 2, pp. 242–244. ISSN 0885-3177. Available from DOI: 10.1097/00006676-199103000-00018.
6. BALAMURUGAN, A.N. et al. Islet Product Characteristics and Factors Related to Successful Human Islet Transplantation From the Collaborative Islet Transplant Registry (CITR) 1999–2010. *American Journal of Transplantation*. 2014, vol. 14, no. 11, pp. 2595–2606. ISSN 1600-6135. Available from DOI: 10.1111/ajt.12872.
7. KISSLER, H.J. et al. Validation of methodologies for quantifying isolated human islets: an islet cell resources study. *Clinical Transplantation*. 2009, vol. 24, no. 2, pp. 236–242. Available from DOI: 10.1111/j.1399-0012.2009.01052.x.
8. WANG, Ling-Jia; KAUFMAN, Dixon B. Digital Image Analysis to Assess Quantity and Morphological Quality of Isolated Pancreatic Islets. *Cell Transplantation*. 2016, vol. 25, no. 7, pp. 1219–1225. Available from DOI: 10.3727/096368915x689947.
9. BUCHWALD, Peter et al. Fully Automated Islet Cell Counter (ICC) for the Assessment of Islet Mass, Purity, and Size Distribution by Digital Image Analysis. *Cell Transplantation*. 2016, vol. 25, no. 10, pp. 1747–1761. Available from DOI: 10.3727/096368916x691655.
10. HABART, David; BLAZEK, Adam; SAUDEK, Frantisek. IsletNet: Web service for automated analysis of islet graft images. In: *IPITA Congress 2017 Abstracts* [online]. IPITA, 2017, p. 219 [visited on 2023-12-05]. Available from: https://tts.org/component/tts/?view=presentation&id=185334.
11. WEIR, G C; BONNER-WEIR, S. Islets of Langerhans: the puzzle of intraislet interactions and their relevance to diabetes. *Journal of Clinical Investigation*. 1990, vol. 85, no. 4, pp. 983–987. ISSN 0021-9738. Available from DOI: 10.1172/jci114574.

12. KATSAROU, Anastasia et al. Type 1 diabetes mellitus. *Nature Reviews Disease Primers*. 2017, vol. 3, no. 1. ISSN 2056-676X. Available from DOI: 10.1038/nrdp.2017.16.

13. VIGNESH, JP; MOHAN, V. Hypoglycaemia unawareness. *The Journal of the Association of Physicians of India*. 2004, vol. 52, pp. 727–32.

14. GRAVELING, Alex J.; FRIER, Brian M. Hypoglycaemia: An overview. *Primary Care Diabetes*. 2009, vol. 3, no. 3, pp. 131–139. ISSN 1751-9918. Available from DOI: 10.1016/j.pcd.2009.08.007.

15. CERIELLO, Antonio; MONNIER, Louis; OWENS, David. Glycaemic variability in diabetes: clinical and therapeutic implications. *The Lancet Diabetes; Endocrinology*. 2019, vol. 7, no. 3, pp. 221–230. ISSN 2213-8587. Available from DOI: 10.1016/s2213-8587(18)30136-0.

16. SHAPIRO, A. M. James; POKRYWCZYNSKA, Marta; RICORDI, Camillo. Clinical pancreatic islet transplantation. *Nature Reviews Endocrinology*. 2016, vol. 13, no. 5, pp. 268–277. ISSN 1759-5037. Available from DOI: 10.1038/nrendo.2016.178.

17. RICORDI, Camillo et al. Automated Method for Isolation of Human Pancreatic Islets. *Diabetes*. 1988, vol. 37, no. 4, pp. 413–420. ISSN 1939-327X. Available from DOI: 10.2337/diab.37.4.413.

18. PAPAS, Klearchos K; SUSZYNSKI, Thomas M; COLTON, Clark K. Islet assessment for transplantation. *Current Opinion in Organ Transplantation*. 2009, vol. 14, no. 6, pp. 674–682. ISSN 1087-2418. Available from DOI: 10.1097/mot.0b013e328332a489.

19. DVOŘÁK, Jiří; ŠVIHLÍK, Jan; HABART, David; KYBIC, Jan. Comparison of volume estimation methods for pancreatic islet cells. In: GIMI, Barjor; KROL, Andrzej (eds.). *SPIE Proceedings*. SPIE, 2016. Available from DOI: 10.1117/12.2216783.

20. *Qualitative and Quantitative Assessment of Human Islets Using Dithizone (DTZ)* [online]. Powers and Brissova Research Group, Vanderbilt University, 2020 [visited on 2023-12-11]. Available from: https://hpap.pmacs.upenn.edu/explore/workflow/islet-physiology-studies?protocol=7.

21. HABART, David et al. Automated Analysis of Microscopic Images of Isolated Pancreatic Islets. *Cell Transplantation*. 2016, vol. 25, no. 12, pp. 2145–2156. Available from DOI: 10.3727/096368916x692005.

22. NOVAK, Roman et al. *Sensitivity and Generalization in Neural Networks: an Empirical Study*. arXiv, 2018. Available from DOI: 10.48550/ARXIV.1802.08760.

23. YU, Xiaoyu et al. A Smartphone-Fluidic Digital Imaging Analysis System for Pancreatic Islet Mass Quantification. *Frontiers in Bioengineering and Biotechnology*. 2021, vol. 9. Available from DOI: 10.3389/fbioe.2021.692686.

24. NICLAUSS, Nadja et al. Computer-Assisted Digital Image Analysis to Quantify the Mass and Purity of Isolated Human Islets Before Transplantation. *Transplantation*. 2008, vol. 86, no. 11, pp. 1603–1609. Available from DOI: 10.1097/tp.0b013e31818f671a.

25. STEGEMANN, Jan P; O'NEIL, John J; NICHOLSON, Don T; MULLON, Claudy J.-P. Improved assessment of isolated islet tissue volume using digital image analysis. *Cell Transplantation*. 1998, vol. 7, no. 5, pp. 469–478. ISSN 0963-6897. Available from DOI: https://doi.org/10.1016/S0963-6897(98)00017-7.

26. FRIBERG, Andrew S. et al. Quantification of the Islet Product: Presentation of a Standardized Current Good Manufacturing Practices Compliant System With Minimal Variability. *Transplantation*. 2011, vol. 91, no. 6, pp. 677–683. Available from DOI: 10.1097/tp.0b013e31820ae48e.

27. WANG, Ling-Jia et al. Application of Digital Image Analysis to Determine Pancreatic Islet Mass and Purity in Clinical Islet Isolation and Transplantation. *Cell Transplantation*. 2015, vol. 24, no. 7, pp. 1195–1204. Available from DOI: 10.3727/096368914x681612.

28. GMYR, Valery et al. Automated Digital Image Analysis of Islet Cell Mass Using Nikon's Inverted Eclipse Ti Microscope and Software to Improve Engraftment may Help to Advance the Therapeutic Efficacy and Accessibility of Islet Transplantation across Centers. *Cell Transplantation*. 2015, vol. 24, no. 1, pp. 1–9. Available from DOI: 10.3727/096368913x667493.

29. ŠVIHLÍK, Jan; KYBIC, Jan; HABART, David. Color normalization for robust evaluation of microscopy images. In: TESCHER, Andrew G. (ed.). *Applications of Digital Image Processing XXXVIII*. SPIE, 2015. Available from DOI: 10.1117/12.2188236.

30. RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv, 2015. Available from DOI: 10.48550/ARXIV.1505.04597.

31. ŠVIHLÍK, Jan; KYBIC, Jan; HABART, David. Automated separation of merged Langerhans islets. In: STYNER, Martin A.; ANGELINI, Elsa D. (eds.). *Medical Imaging 2016: Image Processing*. SPIE, 2016. ISSN 0277-786X. Available from DOI: 10.1117/12.2216798.

32. BREU, H.; GIL, J.; KIRKPATRICK, D.; WERMAN, M. Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1995, vol. 17, no. 5, pp. 529–533. ISSN 0162-8828. Available from DOI: 10.1109/34.391389.

33. ROERDINK, Jos B.T.M.; MEIJSTER, Arnold. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*. 2000, vol. 41, no. 1, 2, pp. 187–228. ISSN 0169-2968. Available from DOI: 10.3233/fi-2000-411207.

34. LIN, Tsung-Yi et al. *Microsoft COCO: Common Objects in Context* [https://cocodataset.org/]. 2017.

35. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. *Deep Residual Learning for Image Recognition*. 2015. Available from arXiv: 1512.03385 [cs.CV].

36. XIE, Saining; GIRSHICK, Ross; DOLLÁR, Piotr; TU, Zhuowen; HE, Kaiming. *Aggregated Residual Transformations for Deep Neural Networks*. 2017. Available from arXiv: 1611.05431 [cs.CV].

37. VASWANI, Ashish et al. *Attention Is All You Need*. arXiv, 2017. Available from DOI: 10.48550/ARXIV.1706.03762.

38. HE, Kelei et al. Transformers in medical image analysis. *Intelligent Medicine*. 2023, vol. 3, no. 1, pp. 59–78. ISSN 2667-1026. Available from DOI: 10.1016/j.imed.2022.07.002.

39. DOSOVITSKIY, Alexey et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv, 2020. Available from DOI: 10.48550/ARXIV.2010.11929.

40. LIN, Tianyang; WANG, Yuxin; LIU, Xiangyang; QIU, Xipeng. A survey of transformers. *AI Open*. 2022, vol. 3, pp. 111–132. ISSN 2666-6510. Available from DOI: 10.1016/j.aiopen.2022.10.001.

41. LIU, Ze et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. arXiv, 2021. Available from DOI: 10.48550/ARXIV.2103.14030.

42. ASGARI TAGHANAKI, Saeid et al. Deep semantic segmentation of natural and medical images: a review. *Artificial Intelligence Review*. 2020, vol. 54, no. 1, pp. 137–178. ISSN 1573-7462. Available from DOI: 10.1007/s10462-020-09854-1.

43. SIAM, Mennatullah; ELKERDAWY, Sara; JAGERSAND, Martin; YOGAMANI, Senthil. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017. Available from DOI: 10.1109/itsc.2017.8317714.

44. LUC, Pauline et al. Predicting Deeper Into the Future of Semantic Segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.
45. MÜLLER, Dominik; SOTO-REY, Iñaki; KRAMER, Frank. Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes*. 2022, vol. 15, no. 1. ISSN 1756-0500. Available from DOI: 10.1186/s13104-022-06096-y.
46. KARIMI, Davood; WARFIELD, Simon K.; GHOLIPOUR, Ali. Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations. *Artificial Intelligence in Medicine*. 2021, vol. 116, p. 102078. ISSN 0933-3657. Available from DOI: 10.1016/j.artmed.2021.102078.
47. HE, Kaiming; GKIOXARI, Georgia; DOLLÁR, Piotr; GIRSHICK, Ross. *Mask R-CNN*. arXiv, 2017. Available from DOI: 10.48550/ARXIV.1703.06870.
48. REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross; SUN, Jian. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. Available from arXiv: 1506.01497 [cs.CV].
49. GIRSHICK, Ross. *Fast R-CNN*. arXiv, 2015. Available from DOI: 10.48550/ARXIV.1504.08083.
50. CAI, Zhaowei; VASCONCELOS, Nuno. *Cascade R-CNN: High Quality Object Detection and Instance Segmentation*. arXiv, 2019. Available from DOI: 10.48550/ARXIV.1906.09756.
51. CHEN, Kai et al. *Hybrid Task Cascade for Instance Segmentation*. arXiv, 2019. Available from DOI: 10.48550/ARXIV.1901.07518.
52. HENDERSON, Paul; FERRARI, Vittorio. End-to-End Training of Object Class Detectors for Mean Average Precision. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2017, pp. 198–213. ISBN 9783319541938. ISSN 1611-3349. Available from DOI: 10.1007/978-3-319-54193-8_13.
53. LIN, Tsung-Yi et al. *Microsoft COCO: Common Objects in Context* [https://cocodataset.org/]. 2017. Accessed: Jan 6, 2024.
54. OPENCV CONTRIBUTORS. *OpenCV: Open Source Computer Vision Library* [https://opencv.org]. 2023.
55. CHEN, Kai et al. *MMDetection: Open MMLab Detection Toolbox and Benchmark* [https://github.com/open-mmlab/mmdetection]. 2021.
56. PASZKE, Adam et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library* [https://pytorch.org/]. 2019.
57. METACENTRUM. *MetaCentrum (MetaVo): Academic Computing Infrastructure* [https://metavo.metacentrum.cz/en/index.html]. [N.d.].
58. DENG, Jia et al. ImageNet: A Large-Scale Hierarchical Image Database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

Concents of the media attachment

- `README.md.txt` ..... description of the contents and how to run the code
- `data_preparation` ..... directory with scripts for image preprocessing
- `evaluation` ..... Jupyter notebooks for model evaluation
- `model_predictions` ..... directory with the predictions all models
- `thesis` ..... folder containing the written thesis
- `training`
  - `augmentations` ..... directory with implemented data augmentations
  - `configs` ..... folder containing configurations of all models
  - `local_scripts` ..... scripts for running the training locally
  - `metacentrum_scripts` ..... scripts for running the training on MetaCentrum
- `utils` ..... directory with helper functions and constants
- `visualization` ..... scripts for visualization of the instance segmentation results
- `work_dirs` ..... directory with saved trained models