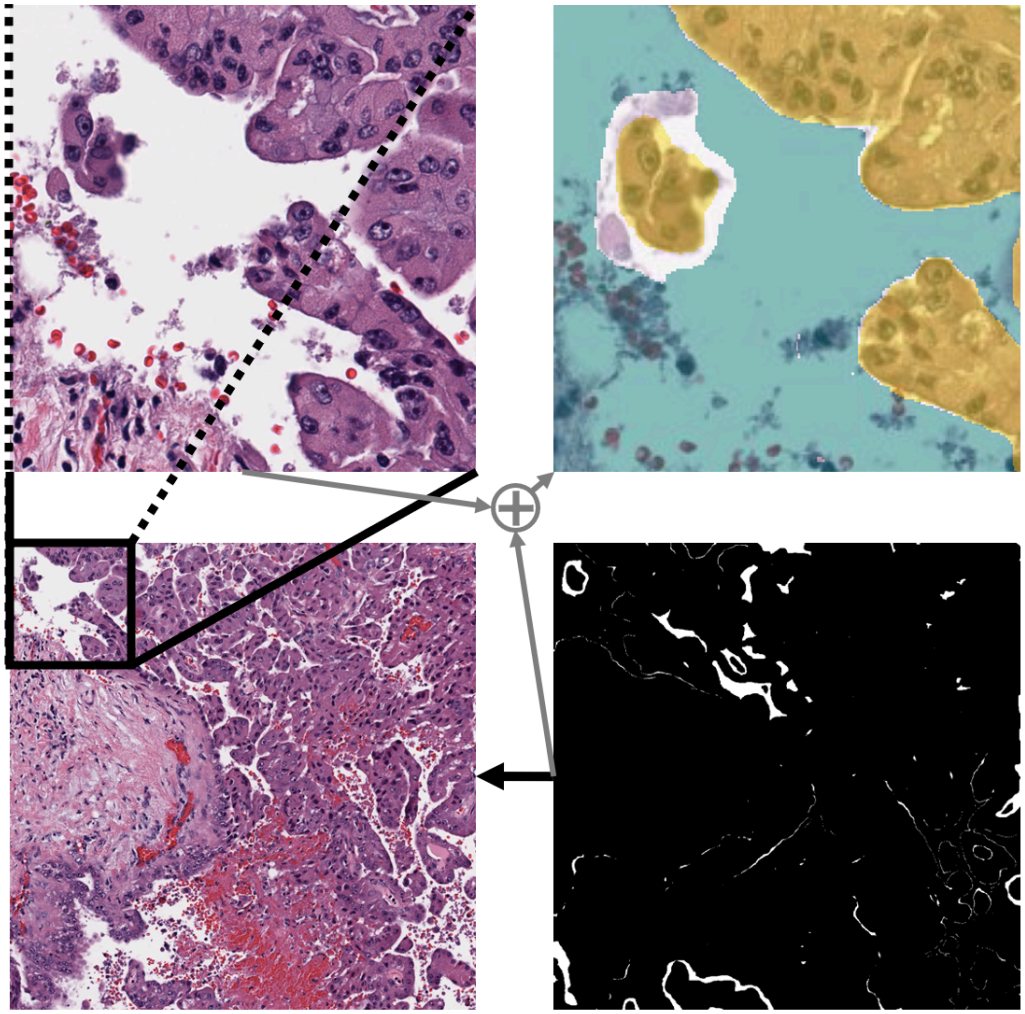
OCELOT 2023: Cell Detection from Cell-Tissue Interaction

Project Seminar Biomedical Image Analysis - Summer Semester 2023

Supervisor: PD Dr. Karl Rohr

Finn Henri Smidt, Luke Voß, Mika Mauro Rother
https://github.com/HenriSmidt/CellDetection

18.4.2023 - 25.7.2023



Abstract

The Ocelot Challenge aims to advance the understanding of cell-tissue relationships and enhance cell detection performance. The dataset consists of images containing cell patches alongside their corresponding surrounding tissues. Participants are tasked with leveraging the provided cell nuclei annotations and tissue segmentation to accurately detect and classify all cell nuclei present within the cell patch images. In this report, we propose a fusion network architecture that effectively processes the given multi-modal data by incorporating state-of-the-art methods in both object detection and segmentation tasks. Our approach employs a YOLOv8 object detection model for precise cell nucleus detection while incorporating a SegFormer model for improved class prediction through learned tissue segmentation. Notably, our experiments showcase high performance, securing a position within the top 5 models on the leaderboard, while being trained on a low-cost computational machine.

Performed tasks and written parts of the report

Since we collaborated closely throughout the entire project, all tasks, including literature review, model architecture, model implementation, writing the report, etc., were performed collectively and most of the time in person. Thus, attempting to dissect individual contributions would require significant effort and would fail to provide a comprehensive understanding for any reviewer.

Contents

Chapter 1

Introduction

Cell detection is one of the most important tasks in computational pathology (CPATH) (Abels et al. [2019]). To assist researchers in advancing the efficacy of their cell detection models, Whole Slide Images (WSIs) emerged as a valuable tool for analyzing digitized patient specimens (Pantanowitz et al. [2015]). WSIs can be envisioned as high-resolution images generated by scanning an entire microscopic slide and subsequently consolidating the data into a single digital file. This process ensures the preservation and storage of patient information in digital format.

In real-life scenarios, pathologists often rely on understanding the tissue context within a large Field of View (FoV) image before zooming in to identify individual cells. This contextual information is crucial for accurately classifying isolated cells, particularly when distinguishing tumor cells from background cells. To achieve this, pathologists utilize tissue segmentation to determine which regions are cancerous and which are not. By incorporating this knowledge, they can analyze smaller FoVs within cell images to identify potential tumor cells and differentiate them from background cells. A more detailed depiction of this behavior is presented in Figure 1.1. However, replicating this process for cell-detecting machine-learning models is challenging due to the limited availability of comprehensive datasets that encompass both cell and tissue images.

In our project, we make use of the OCELOT dataset (Ryu et al. [2023]), which consists of diverse Whole Slide Images (WSIs) depicting various tissue samples. OCELOT is specifically designed to support the training of cell detection and classification models by providing additional segmented tissue information. Chapters 2 and 4 provide comprehensive details about the OCELOT dataset, including a discussion on previous work and our implementation. The dataset consists of cell images with annotated cell nuclei locations and classes, along with accompanying images of the surrounding tissue and their corresponding segmentation masks.

However, it is important to acknowledge that, despite correct labeling, approximately only 93% of tumor cells (TC) are located within the segmented cancer area (CA) of the tissue patches, while approximately 85% of background cells (BC) are situated outside the CA, presenting a challenge for achieving accurate segmentation results. We selected the topic of overlapping cell and tissue annotations for our project as it aligns with the previously mentioned Ocelot challenge (OCELOT [2023]). Participating in this challenge offers distinct advantages that greatly support our project seminar's objectives.

1. Clear-defined research objective and performance metrics
2. Facilitated comparability of model performance with other participants

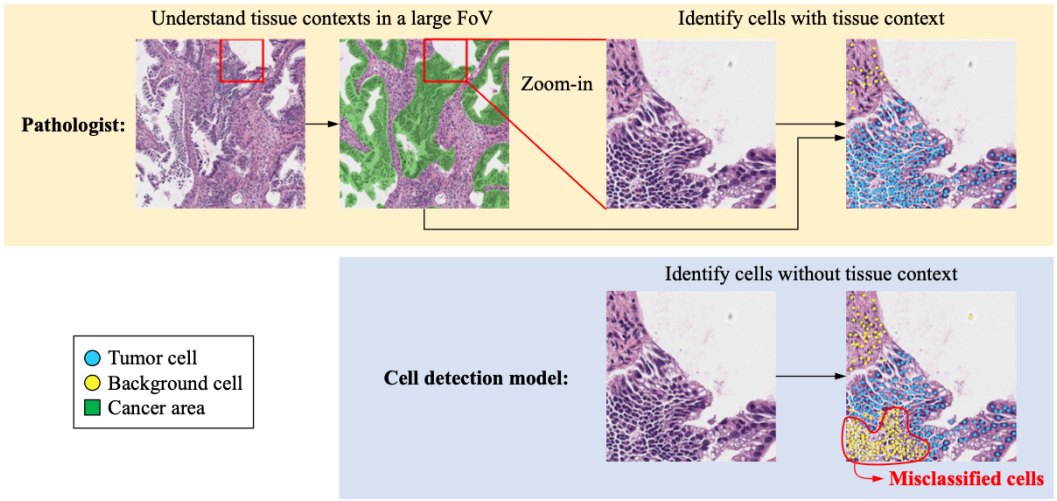


Figure 1.1.: Comparison of Pathologist Behavior and Cell Detection Models: Pathologists rely on tissue context to annotate cells accurately, while machine learning models without this context may fail and misclassify cells. (Figure taken from Ryu et al. [2023]).

3. Access to a freely available dataset, eliminating the need for labor-intensive data collection

4. Active contribution to state-of-the-art research

Firstly, we will offer a comprehensive overview of the previous work conducted in the field, highlighting the advancements made before the OCELOT challenge. Subsequently, we will provide an in-depth explanation of our methods and present the architecture of our model. In the implementation section, we will apply our model to the data obtained from the OCELOT dataset, providing insights into the specific hyperparameters and methodologies employed during training. Moving on to the results chapter, we will showcase the performance of our model by presenting the results of numerous experiments that validate its accurate detection and classification of cells with the assistance of tissue annotations. Lastly, in the discussion and conclusion chapter, we will summarize our findings and draw meaningful conclusions from our project.

Chapter 2

Previous Work

This chapter focuses on the OCELOT dataset and the overarching concept of the challenge. In section 2.1, we present a detailed description of previous research approaches that have addressed cell detection, tissue segmentation, large Field of Views (FoVs), and cell-tissue relationships. Additionally, we present the OCELOT challenge (OCELOT [2023]) and its accompanying published paper (Ryu et al. [2023]) in section **??**, and discuss the obtained results to provide readers with a clear understanding of the challenge's objectives.

2.1. Overlapped Cell on Tissue Dataset for Histopathology

The Ocelot dataset (OCELOT [2023]) introduces a new research direction by publishing the OCELOT dataset and focusing on cell detection using cell-tissue relationships. However, it is important to acknowledge the significant body of related work that has been conducted. In the following, we provide a summary of previous contributions and published datasets:

- Graham et al. [2019] and Naylor et al. [2019] have focused on cell detection within a single organ. The former utilizes a novel dataset of Haematoxylin & Eosin (H&E) stained colorectal adenocarcinoma image tiles, while the latter focuses on data from Triple Negative Breast Cancer (TNBC) patients.
- Other studies have explored cell detection across multiple organs. Gamper et al. [2020] expanded existing datasets to include 19 different tissue types, while Vu et al. [2018] utilized four different types of cancer for their model.
- Tissue segmentation has also been investigated. Karimi et al. [2020] developed a deep-learning-based method for grading prostate cancer (PCa), while Qian et al. [2022] employed a transformer model in a weakly supervised setting to perform tissue segmentation using H&E stained histopathology images of colon cancer.
- Da et al. [2022] employed a dataset consisting of subsets for both cell detection and tissue segmentation. However, the absence of overlapping structures within their data should be noted, as cell and tissue annotations were collected independently from different patients.
- The Tiger dataset TIGER [2022] includes annotations for both cells and tissue in the study of breast cancer WSIs. However, the data does not explicitly utilize the overlapping information for cell-tissue interaction, despite having cell-annotated areas within the tissue-annotated regions.

2.2. Introduction to the OCELOT Challange

The OCELOT challenge was launched in March 2023 to develop a cell detection model that leverages a deeper understanding of the surrounding tissue context based on the OCELOT dataset. The primary evaluation metric for the challenge, as shown in Table 2.1, is the F1 score. The challenge consists of multiple phases, including training, validation, and testing, with the corresponding data releases occurring gradually over several months. In addition to the opportunity for scientific advancement, participants in the OCELOT challenge have the chance to win prize money. The first-place winner will receive a reward of $1000, the second-place winner will receive $500, and the third-place winner will be awarded $200. The following sections aim to yield a comprehensive overview of the provided dataset and introduce the methods and results of the respective paper.

Dataset

In contrast to previous works, the OCELOT dataset significantly expands the amount of cell and tissue annotations compared to the previously published TIGER dataset (TIGER [2022]). OCELOT encompasses 306 Whole Slide Images (WSIs) sourced from six different organs, namely the kidney, head-and-neck, prostate, stomach, endometrium, and bladder. The dataset's design specifically aims to capture the hierarchical relationship between cells and tissues, particularly within the tumor environment. To enhance the understanding of the dataset's composition, let's examine the mathematical representation of the dataset:

$$D = \{(x_s, y_s^c, x_l, y_l^t, c_x, c_y)_i\}_{i=1}^{N}. \tag{2.1}$$

Here x_s, x_l are the small and large Field of View (FoV) patches extracted from the WSI, y_c^s, y_l^t refer to the corresponding cell and tissue annotations, c_x, c_y are the relative coordinates of the center of x_s within x_l, and N is the total number of image pairs. As depicted in Figure 2.1, the dataset includes a sample that effectively emulates the cell detection process performed by real pathologists. The visualization provides a clear representation, highlighting the intentional design of the dataset.

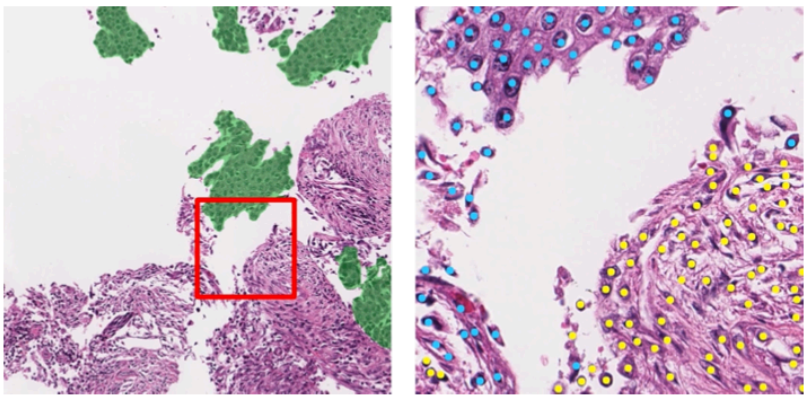


Figure 2.1.: Large FoV of the segmented tissue structure (left) and zoomed-in FoV of the annotated cells (right). (Figure taken from Ryu et al. [2023]).

Methodology

The authors employed three distinct approaches to integrate surrounding tissue segmentation information into the network architecture. Subsequently, we will provide a concise summary of each method.

Tissue-label leaking model

The initial approach involved incorporating the accurate tissue annotation as an additional input for training the cell detection model. In this approach, the model utilizes the cell path x_s as one input, and the corresponding cropped, upsampled, and channel-wise concatenated tissue label y_s^t as the second input. Figure 2.2 provides a visual depiction of the model described. Building upon the successful tissue-label leaking model, the OCELOT team sought to further investigate additional cell-tissue relationships that could be leveraged for training a cell detection model. To achieve this, they adopted a bi-directional information-sharing approach employing two models: a tissue segmentation model and a cell detection model. This architecture facilitated feature sharing in both directions, enabling information flow from tissue to cell and vice versa. In this dual-branch architecture, the cell detection model was designed as a segmentation task. Thus the same architecture for both the cell detection and tissue segmentation models could be used, streamlining the overall workflow.

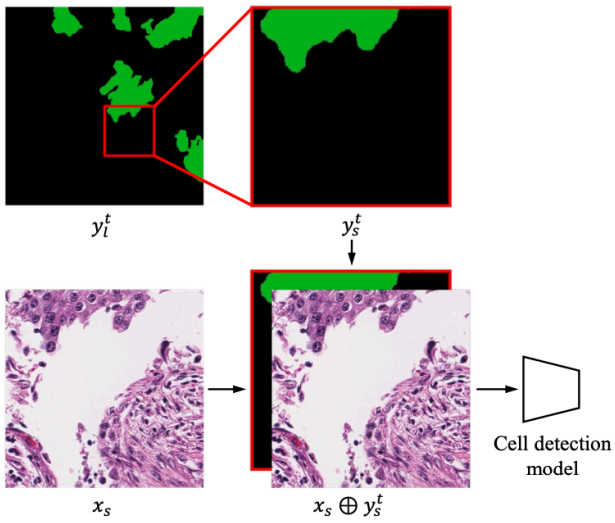


Figure 2.2.: Tissue-label leaking model. (Figure taken from Ryu et al. [2023]).

Tissue prediction injection models

Compared to the tissue-label leaking model, the tissue prediction injection approach utilizes tissue segmentation solely as predictions instead of relying on ground truth labels as inputs. The OCELOT team developed four distinct models by incorporating tissue predictions at four different injection points:

(a) before the encoder as second input (*Pred-to-input*),

(b) after the encoder (*Pred-to-inter-1*),

(c) after the ASPP[1] module (*Pred-to-inter-2*)

(d) after the decoder (*Pred-to-output*).

[1] Atrous Spatial Pyramid Pooling (ASPP) is a semantic segmentation module for resampling a given feature layer at multiple rates before convolution (Chen et al. [2017]).

Figure 2.3 gives an overview of these four different tissue prediction models.

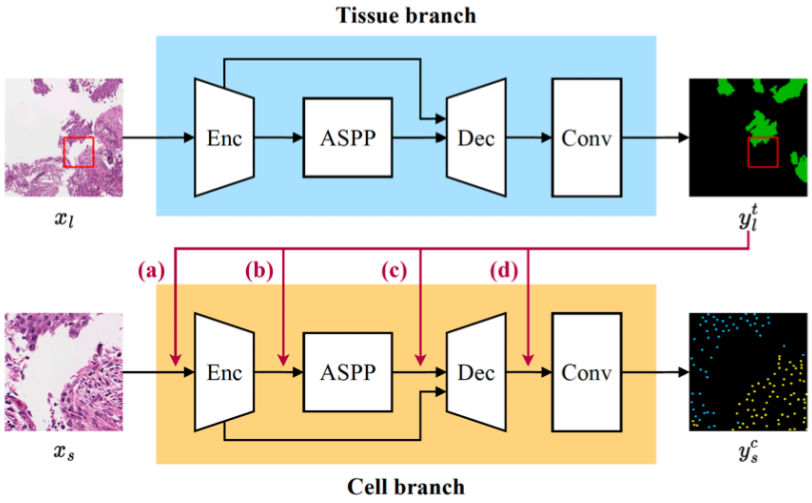


Figure 2.3.: Tissue prediction injection models. (Figure taken from Ryu et al. [2023]).

Cell-Tissue feature sharing model

To introduce greater diversity and flexibility in the flow of cell-tissue information, the OCELOT team developed an additional set of models for cell-tissue feature sharing. These models enable information exchange between the cell detection model and the tissue segmentation model at various feature representation levels in each network. This approach allows for fluid and dynamic information sharing between the two models. The resulting best model is illustrated in Figure 2.4.

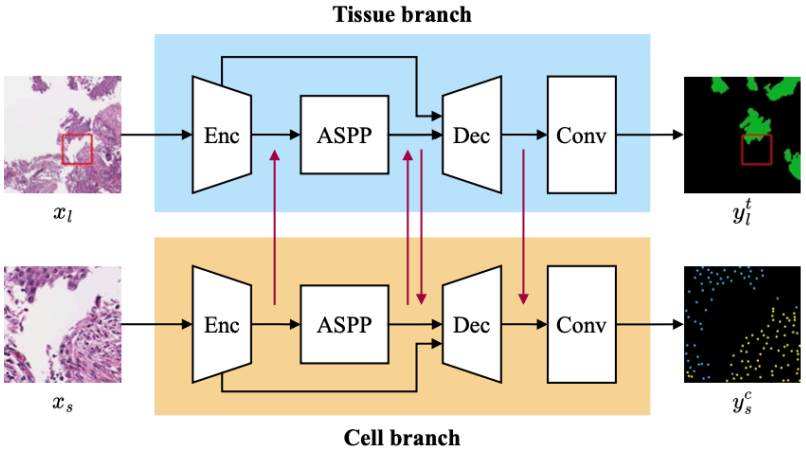


Figure 2.4.: Cell-Tissue feature sharing model. (Figure taken from Ryu et al. [2023]).

Results

All six models described above were compared to each other using the OCELOT, TIGER, and CARP datasets. The DeepLabV3+ architecture, with a ResNet-34 encoder, was employed for all models (Chen et al. [2018]; He et al. [2015]). The results of these comparisons are presented in Table 2.1. The findings demonstrate a substantial improvement when utilizing overlapping cell and tissue images for training a cell detection model.

Method	OCELOT		TIGER		CARP	
	Val	Test	Val	Test	Val	Test
Cell-only	68.87±1.76	64.44±1.82	63.89±1.39	53.82±1.23	78.48±0.69	70.96±1.47
Pred-to-input	**73.36±0.59**	69.65±3.93	66.00±2.00	53.29±1.30	79.46±0.79	72.98±0.82
Pred-to-inter-1	72.74±0.50	70.54±2.20	**66.19±1.02**	**55.87±1.78**	79.74±0.80	73.05±0.69
Pred-to-inter-2	72.68±1.58	**71.23±0.96**	65.43±1.14	54.75±2.25	**79.87±0.78**	**73.14±1.53**
Pred-to-output	66.85±5.62	65.05±3.72	63.02±0.16	53.32±0.42	78.92±0.60	72.61±0.95
Feature-sharing	72.30±0.73	68.91±2.52	65.64±1.07	55.10±2.18	79.38±0.74	73.00±0.33
Tissue-label leaking	76.56±0.80	74.20±0.91	69.71±0.61	61.66±1.16	80.13±1.04	72.97±0.49

Table 2.1.: Cell-Detection mean F1 scores per model. Except for the Tissue-label leaking model, the highest score is written in bold and the second highest score is underlined (Table and caption taken from Ryu et al. [2023]).

Chapter 3

Methods

In this chapter, we present the state-of-the-art YOLO object detection model and SegFormer model for segmentation, which serve as the primary backbones of our model architecture. Additionally, we introduce the weighted boxes fusion technique, which we utilized in our implementation to consolidate predictions from multiple detection models into a single expert predictor.

3.1. YOLO Object Detection

YOLO (You Only Look Once) has emerged as a pivotal real-time object detection system, revolutionizing applications in robotics, driverless cars, and video monitoring (Terven and Cordova-Esparza [2023]). First introduced by Redmon et al. [2016], YOLO offers a new approach to object detection. Traditionally, object detection involved repurposing classifiers for detection tasks. In contrast, YOLO tackles object detection as a regression problem by predicting spatially separated bounding boxes and associated class probabilities. By employing a single neural network, YOLO directly predicts bounding boxes and class probabilities from complete images in a single evaluation. This unified architecture enables direct end-to-end optimization, leading to exceptional detection performance. The base YOLO model operates in real-time, processing images at an impressive rate of 45 frames per second. Moreover, a smaller variant of the network known as Fast YOLO achieves an astounding 155 frames per second while maintaining double the mean Average Precision (mAP) compared to other real-time detectors (Redmon et al. [2016]).

By combining the benefits of speed, accuracy, and unified architecture, YOLO has established itself as a fundamental framework in various domains, offering remarkable capabilities for real-time object detection.

Anchorbox free design

The YOLOv8 detection model (Jocher et al. [2023]) represents a recent advancement in object detection algorithms, characterized by its anchor-free approach. Unlike previous versions of YOLO, which relied on predicting the offset from predefined anchor boxes, YOLOv8 directly predicts the center coordinates of objects. This shift in methodology alleviates the challenges associated with anchor boxes, which often fail to accurately represent the distribution of objects in custom datasets (ROBOFLOW [2023]).

By adopting an anchor-free detection strategy, YOLOv8 significantly reduces the number of box predictions, leading to notable improvements in computational efficiency. One particular

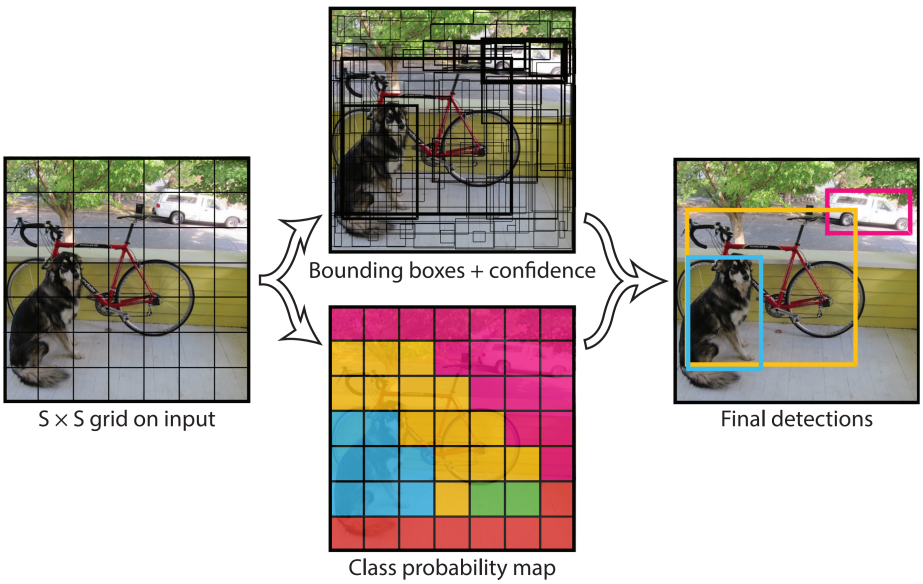


Figure 3.1.: The image is first divided into an $S \times S$ grid. Then for each grid cell B bounding boxes, confidence for those boxes and C class probabilities are being predicted, resulting in an $S \times S \times (B * 5 + C)$ tensor. (Figure taken from Redmon et al. [2016]).

step that benefits from this approach is the Non-Maximum Suppression (NMS) stage, a complex post-processing step that filters out redundant detections. The reduced number of box predictions accelerates the NMS process, enhancing the overall inference speed of the model (ROBOFLOW [2023]).

Architectural modifications

In terms of architectural modifications, the YOLOv8 model introduces several key changes. First, the initial 6x6 convolutional layer in the stem is replaced by a 3x3 convolutional layer. This alteration aims to adapt the model's initial feature extraction process. Additionally, the primary building block within the architecture undergoes modification, and the substitution of C2f for C3 occurs. In this context, C2f refers to the concatenation of all outputs from the bottleneck, which consists of two 3x3 convolutions with residual connections. On the other hand, C3 solely utilizes the output of the last bottleneck (ROBOFLOW [2023]). It is worth noting that the bottleneck structure employed in YOLOv8 remains consistent with that of YOLOv5, but a notable change is made to the kernel size of the first convolution, transitioning from 1x1 to 3x3. This adjustment aligns YOLOv8 with the ResNet block initially introduced in 2015, signifying a reversion to this earlier architecture (ROBOFLOW [2023]).

In the neck module of YOLOv8, features are directly concatenated without enforcing uniform channel dimensions. This design choice effectively reduces the parameter count and overall tensor size, which positively impacts the model's efficiency and memory requirements (ROBOFLOW [2023]).

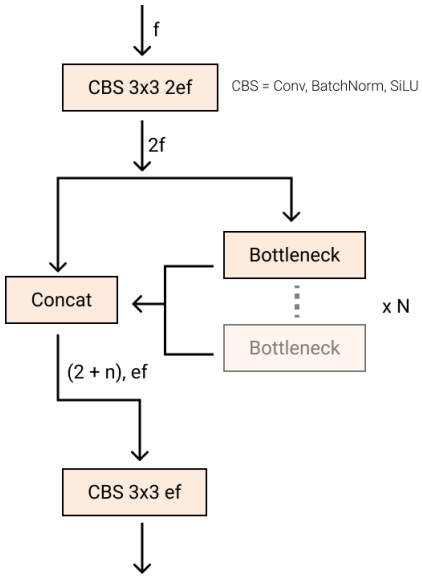


Figure 3.2.: The C2f module takes the outputs from all bottleneck layers and concatenates them. This means that instead of discarding intermediate features and only considering the output of the last bottleneck layer, the C2f module retains and combines the feature maps from all bottleneck layers. (Figure taken from ROBOFLOW [2023]).

Summary

In summary, YOLOv8 represents a novel anchor-free detection model that predicts object centers directly, eliminating the reliance on anchor boxes. The model's architectural enhancements, such as modifications to the stem and the use of Bottleneck structures, demonstrate its progression towards a ResNet-inspired design. Furthermore, the concatenation of features in the neck module without enforcing uniform channel dimensions contributes to improved parameter efficiency and reduced tensor size.

3.2. SegFormer Segmentation

SegFormer (Xie et al. [2021]) is a state-of-the-art semantic segmentation model that utilizes the transformer architecture, originally introduced for natural language processing tasks, in the field of computer vision. SegFormer incorporates the power of self-attention mechanisms to capture long-range dependencies and effectively model spatial relationships within images for accurate and efficient semantic segmentation. It leverages the transformer's self-attention mechanism to capture global and context-rich information from the input image. By attending to different regions of the image, SegFormer can effectively model relationships between pixels and encode spatial dependencies. This enables more accurate and robust semantic segmentation compared to traditional convolutional neural network (CNN)-based approaches.

Semantic Segmentation with Object Detection

SegFormer extends the transformer architecture to handle both object detection and semantic segmentation simultaneously. By combining the capabilities of object detection and semantic

segmentation within a unified model, SegFormer benefits from the complementary strengths of both tasks. In SegFormer, the detection and segmentation branches share the same backbone network, enabling efficient feature extraction. This shared backbone enhances feature representation and facilitates the integration of high-level context into the segmentation process (see Figure 3.3).

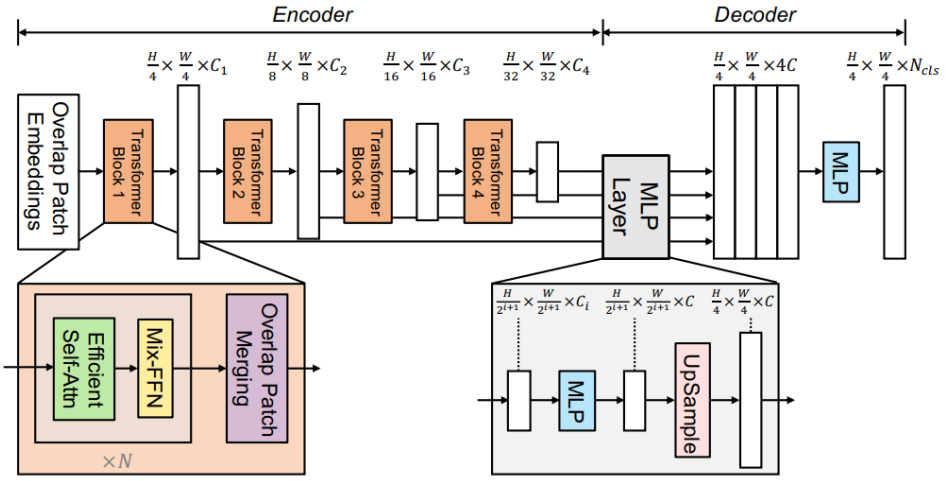


Figure 3.3.: SegFormer framework consists of two main modules: A hierarchical transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse these multi-level features and predict the semantic segmentation mask. "FFN" indicates a feed-forward network. (Figure and caption taken from Xie et al. [2021]).

Performance and Efficiency

To address the computational complexity of the transformer architecture, which typically requires substantial computational resources, SegFormer incorporates various optimizations for efficient training and inference. These optimizations include patch-based processing, efficient attention mechanisms, and progressive upsampling techniques.

By dividing the input image into smaller patches, SegFormer reduces memory requirements during training and inference. Additionally, efficient attention mechanisms, such as sparse self-attention or local window-based attention, are employed to limit computational overhead while maintaining model performance. Moreover, progressive upsampling techniques allow for efficient upsampling of feature maps, reducing the computational burden of dense predictions. Hence, SegFormer has achieved remarkable performance on several benchmark datasets for semantic segmentation tasks. It has demonstrated state-of-the-art results on challenging datasets, including Cityscapes, ADE20K, and COCO-Stuff (Xie et al. [2021]) (see Figure 3.4).

Summary

In summary, SegFormer is a transformer-based semantic segmentation model that harnesses the power of the transformer architecture to achieve outstanding performance in capturing spatial dependencies and context-rich information. With its ability to simultaneously handle

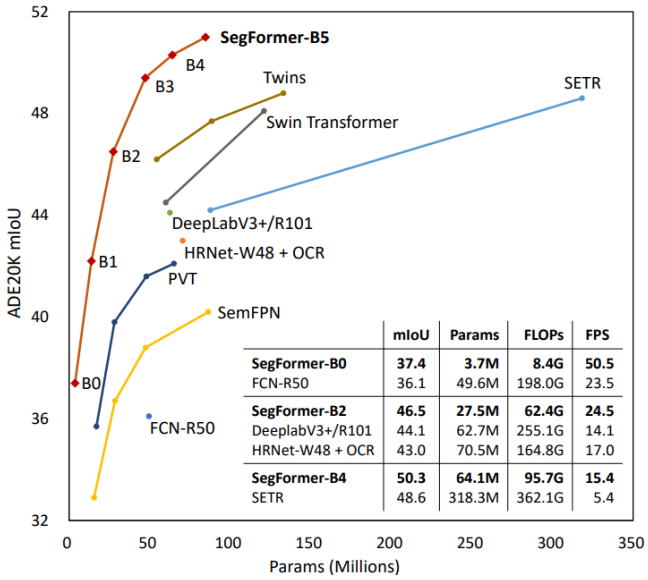


Figure 3.4.: Performance vs. model efficiency for different state-of-the-art segmentation models on the ADE20K dataset. (Figure taken from Xie et al. [2021]).

object detection and semantic segmentation, SegFormer represents a significant advancement in computer vision and offers promising applications in a wide range of fields.

3.3. Weighted Boxes Fusion

The Weighted Boxes Fusion (WBF) algorithm is a method for fusing bounding box predictions from multiple models or augmented versions of the same image introduced by Solovyev et al. Solovyev et al. [2021]. It combines predicted boxes by grouping them into clusters based on an Intersection over Union (IoU) threshold. The new coordinates of the fused box are calculated by performing weighted averaging of the coordinates of the constituent boxes within each cluster. The weights used for averaging are determined by the confidence scores of the corresponding boxes and custom model weights if given. This weighting scheme ensures that boxes with higher confidence scores contribute more to the final fused box's position. Boxes with higher confidence scores have a greater influence on the fused box's coordinates, while boxes with lower confidence scores have a relatively smaller impact.

The confidence of the fused bounding box is calculated by averaging the confidence of the boxes in the cluster and then multiplying by the number of boxes in the cluster and dividing by the number of models or test time augmentations. This rescaling step helps adjust the confidence scores in the fused box list, ensuring that clusters with a low number of boxes, potentially indicating less agreement among models, have decreased confidence scores. On the other hand, clusters with a higher number of boxes, suggesting more consensus among models, receive increased confidence scores. By taking into account both the position and confidence of the constituent boxes, WBF aims to generate a fused box that represents the collective information from multiple predictions more accurately, separating it from traditional methods like non-maximum suppression.

Chapter 4

Implementation

This chapter presents the applied methodology for addressing the task at hand. We begin by introducing our preprocessing techniques, which enabled us to leverage pre-trained object detection models effectively. Additionally, we elaborate on our crucial data augmentation strategies, given the constraints of working with a small dataset. In the subsequent section, we elaborate on our proposed network architecture that utilizes the YOLO model backbone. We then proceed to outline our approach for label correction based on segmentation, which involves the application of the SegFormer model. Lastly, we provide an overview of our implementation of hyperparameter tuning to optimize the performance of our model.

4.1. Preprocessing

In order to effectively work with the given dataset and enhance the performance and computational efficiency of various pre-trained object detection models, it was necessary to preprocess the dataset. The original format of the dataset was modified to align with the requirements of state-of-the-art pre-trained models. Specifically, the center locations of the cells were transformed into squared bounding boxes with a height and width of 30 pixels, centered around the original locations (see Figure 4.1).

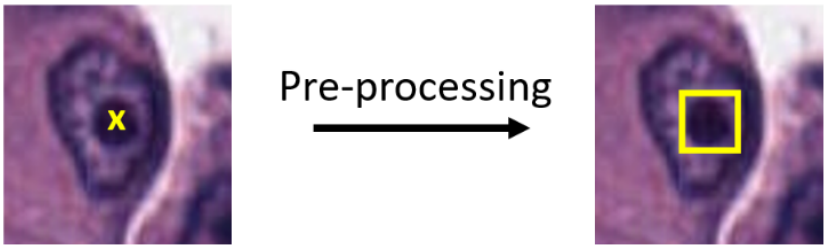


Figure 4.1.: To use a variety of pre-trained object detection models, the center coordinates annotation (left, denoted in Equation (4.1)) was changed into a bounding box format(right, denoted in Equation (4.2))

The dimensions of the bounding boxes were chosen based on the evaluation criteria of the challenge, which involved verifying whether the model correctly identified the cell nucleus within a distance of 15 pixels from the ground truth center location (see Figure 4.2). The dataset annotations were originally provided in a .csv file format, containing the following structure:

$$x_{center},\ y_{center},\ class. \tag{4.1}$$

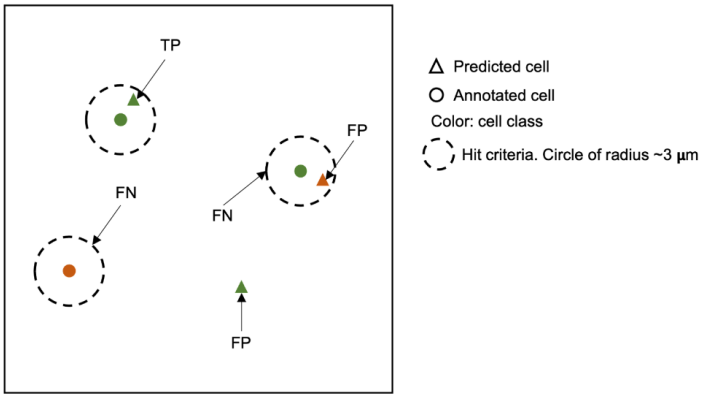


Figure 4.2.: During the model evaluation the cell predictions are checked whether any ground-truth cell is within a valid distance ($\sim$ 15 pixels, $\sim$ 3µm). (Figure taken from OCELOT [2023])

x_{center} and y_{center} denote the x and y coordinates of the nuclei center about the image, and *class* stores the information of the corresponding class (either background or tumor cell). However, certain pre-trained models, such as YOLOv8, which was utilized for obtaining preliminary results, required annotations in a .txt file format with the following structure for each label:

$$class,\ ||x_{center}||,\ ||y_{center}||,\ ||width||,\ ||height||. \tag{4.2}$$

$||\,\cdot\,||$ denotes normalization from 0 to 1. To accommodate this requirement, the x and y coordinates, as well as the width and height of 30 pixels, were normalized concerning the image size. The dataset was then split using a Holdout method, with 80% of the samples allocated for training and 20% for validation. Additional validation and test datasets are scheduled to be released by the authors of the Ocelot challenge on the 11th of August, at which point we will use these as intended and the prior selected validation set as additional training input. We made the decision not to manually correct incorrect annotations in our dataset, as our proposed model will undergo validation and testing using similar data annotations. By preserving the original annotations, we ensure that our model is evaluated under consistent conditions and can generalize to real-world scenarios with similar annotations. This approach avoids introducing potential biases or discrepancies that may arise from manual corrections.

4.2. Data Augmentation

Due to the limited size of the training dataset, employing refined data augmentation techniques becomes crucial for improving the performance and robustness of the object detection models. Most pre-trained models include default data augmentation strategies, which can be tailored to suit the specific use case. In this task, the orientation of the image is not a critical factor, allowing for a wide range of augmentation possibilities. Currently, we employ various dynamic augmentations, including color distortion, rotations, flips, and more advanced methods. These augmentations are applied probabilistically to the training images during each epoch, enabling the model to train on diverse augmented data, thereby expanding the effective training dataset.

Our current data augmentation strategies for both the YOLO and SegFormer models are listed in Table 4.1.

Augmentation	Value YOLO	Value SegFormer
HSV-Hue augmentation (fraction)	0.015	0.05 (image only)
HSV-Saturation augmentation (fraction)	0.7	0.05 (image only)
HSV-Value augmentation (fraction)	0.4	0.05 (image only)
Rotation (+/- degree)	180.0	0
Translation (+/- fraction)	0.1	0
Scale (+/- gain)	0.5	0.5
Flip up-down (probability)	0.5	0.5
Flip left-right (probability)	0.5	0.5
Mosaic (probability)	1.0	0

Table 4.1.: Different used data augmentations and their corresponding values/probabilities. For the SegFormer model these augmentations are applied for both the tissue patch and the corresponding segmentation mask unless specified otherwise

4.3. Segmentation-based label correction

To address the challenge of class ambiguity between tumor and background cells, our approach utilizes the provided tissue patches. We fine-tuned a pre-trained SegFormer model to segment the cancerous and background areas. The segmentation results are then integrated into the prediction process using the following approach: If a prediction has a confidence score below 0.5, its class label is updated based on the corresponding position in the segmentation mask. This means that cells with relatively low confidence are assigned the class label of the segmented tissue at their respective cell centers. By incorporating this segmentation information, we aim to leverage the tissue context to enhance the accuracy of class assignments for ambiguous cells.

4.4. Network Architecture

The proposed method follows a well-organized network architecture comprising the following steps (see Figure 4.3). First, three YOLOv8 models are employed to detect the cell nuclei within the input image. Next, the class predictions undergo a correction process based on the segmentation of the tissue image. To generate the final results, the corrected predictions of each model are fused using weighted boxes fusion. The three models and weights assigned to them in the weighted boxes fusion were determined through evaluation, where different configurations were tested and the best values were selected. Additionally, the optimal IoU threshold was determined through experimentation, ensuring the fusion process achieves the highest performance and accuracy.

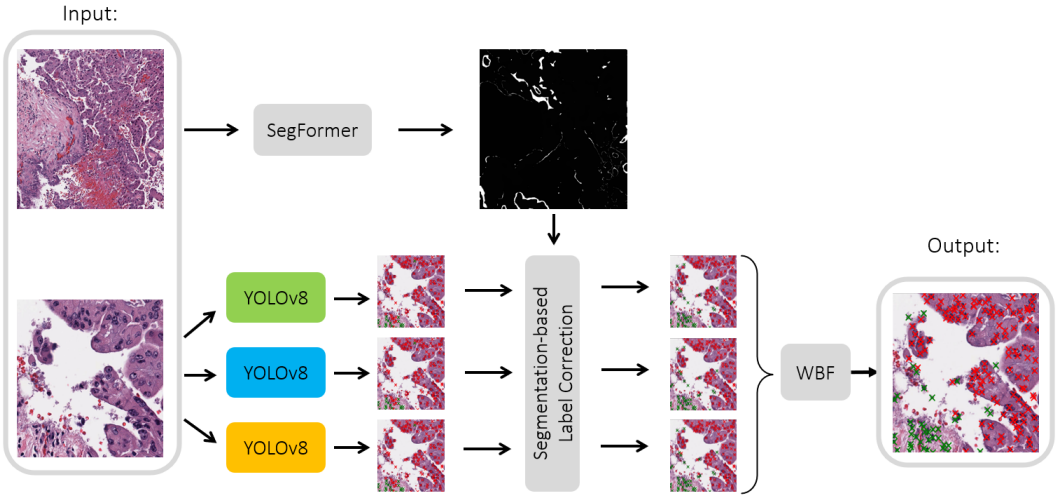


Figure 4.3.: The model architecture comprises the fusion of segmentation predictions and detection results as well as weighted boxes fusion.

4.5. Hyperparameter Optimization

In deep learning, hyperparameters are parameters that are not learned from the data but are set by the user before the training process begins. Proper tuning of hyperparameters is crucial for achieving optimal model performance and generalization. In this section, it is described, how the hyperparameters for the individual models, as well as the ensemble, have been set.

Model Hyperparameter Optimization

To enhance the performance of the YOLOv8, hyperparameter sweeps were employed. We utilized weights and biases (WANDB) to explore various combinations of essential training hyperparameters. The objective was to identify the optimal configuration that maximizes the model's performance. Each configuration of the hyperparameters was set randomly. Random search has the benefit that it does not require testing every configuration to discover an almost optimal solution. This is practical when dealing with a large search space and limited computational resources, making exhaustive methods like grid search impractical. The evaluation was performed in two steps, to further break down the search space and reduce computation time.

First Sweep

The first hyperparameters taken into account are the beginning and closing learning rates for the cosine learning rate schedule. This technique dynamically adjusts the learning rate throughout training to enhance convergence and generalization. Through the exploration of different values for the starting and ending learning rate, we assessed the impact on the model's training progress and final accuracy. Furthermore, the image size was evaluated in the first step.

Second Sweep

In this subsequent hyperparameter sweep, the optimal configurations identified in the initial step are used. At this stage, sweeps were conducted for the class loss, box loss, and DFL (Dual Focal Loss) loss. These losses serve as foundational components within our object detection model. By varying the loss values, their influence on the model's ability to accurately classify objects and determine the precise localization of their bounding boxes was evaluated. The aim was to identify the optimal balance between these losses, thereby leading to the best overall performance.

Additionally, the effects of altering the number of epochs and incorporating rotation augmentations were analyzed. The primary intention behind the augmentations was to increase the model's robustness and resilience. Furthermore, the image size was tested. Lastly, it was evaluated whether or not mosaic augmentations should be deactivated during the last epochs.

4.6. Ensemble Hyperparameter Optimization

Ensemble hyperparameter optimization plays a crucial role in improving the performance of ensemble models, which are composed of multiple individual models combined to make predictions. The optimal combination of models, weights for each model in the ensemble, and the IoU threshold are important hyperparameters that greatly impact the ensemble's effectiveness. Once the predictions of the models have been extracted computing the ensemble results is not computationally expensive, which allowed us to use grid search to find the optimal aforementioned hyperparameters.

Chapter 5

Experimental Results

In this chapter, the results of the methods employed are shown. First, the hyperparameter optimization results are presented, followed by the results of the segmentation-based label correction. Lastly, the results of the weighted boxes fusion are shown. The evaluation metric is the mean F1 (mF1) score. It is commonly used to take precision as well as sensitivity under consideration. For each different cell class, the F1 score is computed by

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.1}$$

using Precision = TP/(TP+FP), and Recall = TP/(TP+FN), where TP, FP, and FN denote True Positive, False Positive, and False Negative detections, respectively. To determine TP, FP, and FN, the hit criterion for the cell detection is illustrated in Figure 4.2.

5.1. YOLOv8 Hyperparameter Optimization

As described in Section 4.5, first, a random search was used to find a nearly optimal set of hyperparameters for the individual models. This hyperparameter optimization has been performed on the pre-trained YOLOv8l consisting of 43.7M parameters. Figures 5.1 and 5.2 present the results of the respective random search steps mentioned in Section 4.5.

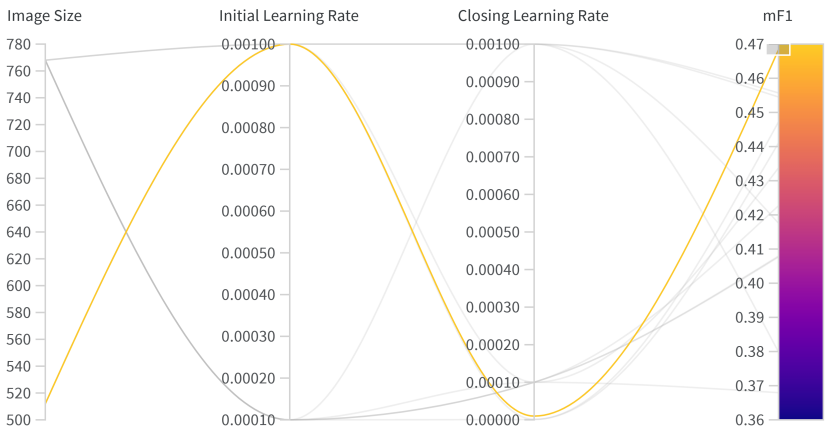


Figure 5.1.: The first hyperparameter search indicates, that an image size of 512, an initial learning rate of 0.001, and a closing learning rate of 0.00001 yield the best results.

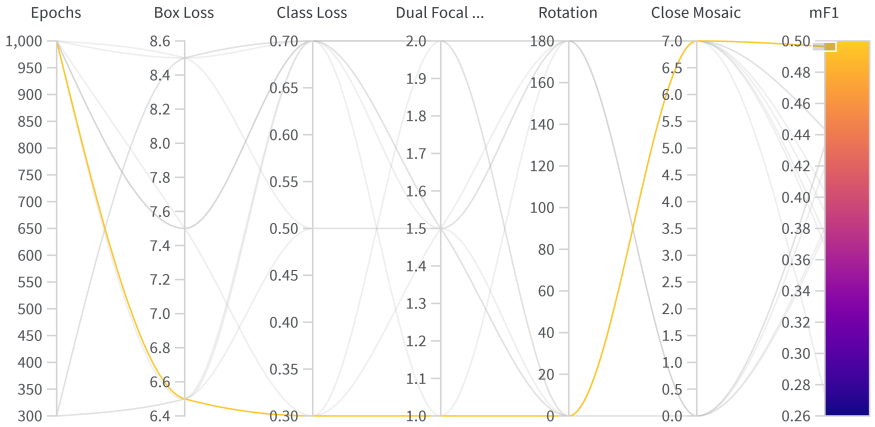


Figure 5.2.: The results of the second stage are best for a model trained with 1000 epochs, a box loss of 6.5, a class Loss of 0.3, a dual focal Loss of 1.0, no rotation augmentation and closing mosaic augmentation for the last 7 epochs.

Each column contains the different possible values of the respective hyperparameter. Each line indicates a configuration. The yellow line indicates the configuration with the highest mean F1 score. The best five models from the hyperparameter search were saved for the ensemble evaluation. Meanwhile, the individual model with the highest score (mF1: 0.4964) has been established as the baseline for further comparison and evaluation. Figure 5.3 showcases a noteworthy illustration of the single-class model's prediction.

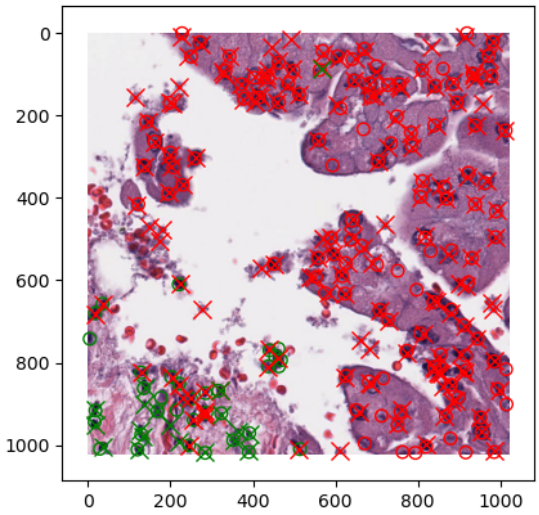


Figure 5.3.: The YOLOv8 model predicts the cell centers well, however, it does not differentiate the classes satisfactorily.

Within the diagram, the ground truth is denoted by circles, while the predictions are represented by $\times$ marks. The color scheme assigns red to cancer cells and green to background

cells. For a prediction to qualify as a true positive, the center of the × mark must coincide with the circle and correspond to the same class, i.e., to the same color.

5.2. Segmenation-Based Label Correction

The SegFromer segmentation model reaches an average precision score of 0.9905. Example segmentations can be seen in Figure 5.4. There one can see that SegFormer performs very well on the tissue dataset.

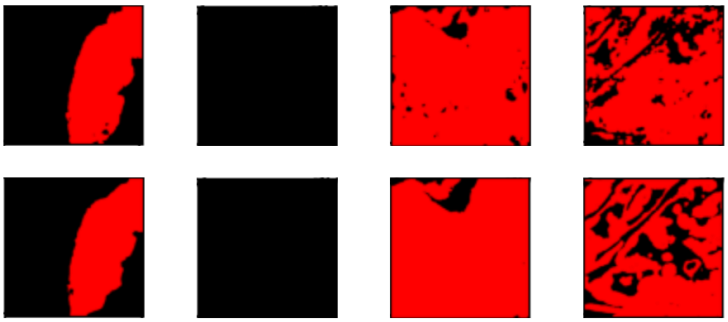


Figure 5.4.: On the first row one can see the segmentation predictions and on the second row the reference labels.

Modifying the classes to correspond to the segmented area, except for those with confidence higher than 0.5, increased the mean F1 score. The segmentation-based label-corrected model achieves now an mF1 score of 0.6489, which is an improvement of 0.1527 compared to the baseline cell detection model. Figure 5.5 shows, that the model incorporating the segmentation predictions distinguishes well between the two areas.

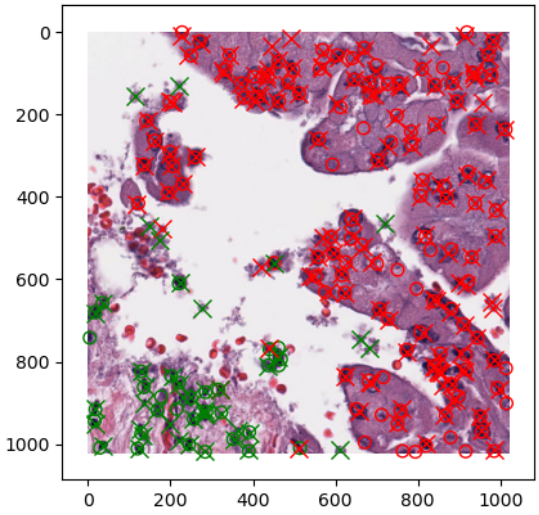


Figure 5.5.: The model incorporating the segmentation results differentiates well between the two classes.

Compared to Figure 5.3, where we consider the baseline cell detection model, we get especially a higher precision for the background cells, since before a lot of background cells were wrongly annotated as tumor cells.

5.3. Weighted Boxes Fusion

The random search was conducted to find the best ensemble comprising three segmentation-based corrected models with individual weights and the grid search finding the ideal IoU threshold stated an ideal IoU of 0.35 (see Figure 5.6). We applied this IoU hyperparameter to our weighted boxes fusion algorithm, described in Section 3.3, and added it to the previous segmentation-based, feature-corrected model to produce our final model, which achieves an mF1 value of 0.6991, a further improvement of 0.0502.

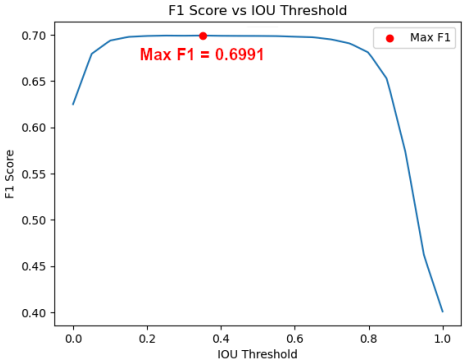


Figure 5.6.: The ideal IoU threshold is 0.35.

Figure 5.7 shows the results of our final model. It is evident that the utilization of weighted boxes fusion has significantly reduced the number of predictions in densely predicted areas. Previously there were always some clusters of predictions with a lot of false positive predictions. Moreover, other false predictions were individually eliminated as they were only detected by one or two models of the ensemble but were correctly identified as false positives by the weighted boxes fusion algorithm.

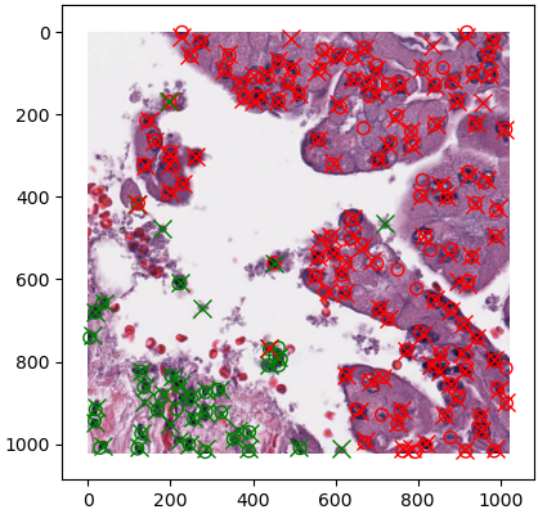


Figure 5.7.: The ensemble helped to reduce the number of false positives in areas of dense predictions.

5.4. Results Overview

Let us finish this chapter by summarizing all our intermediate and final results. In Table 5.1 we compare all three different models with each other and take a closer look at the evaluation results. Our mF1 score increases with each additional method applied, just like the F1 scores of background and tumor cells. The precision of both cell types also increases significantly with each additional method applied. A similar effect can be observed for the recall value of the tumor cells. The only surprising thing is that the recall value of the background cells gets worse when using segmentation-based label correction. This is because the more false negatives there are, the lower the recall value. And since segmentation-based label correction correctly removes many false positives, false negatives often remain (compare the lower left area of the Figures 5.3 and 5.5 with each other).

Model	Pre/BC	Rec/BC	Pre/TC	Rec/TC	F1/BC	F1/TC	mF1
detection only	0.3274	0.5429	0.6512	0.5301	0.4084	0.5844	0.4964
+ seg-based corr.	0.6577	0.4336	0.771	0.7793	0.5226	0.7752	0.6489
+ weighted boxes	0.6805	0.4744	0.8315	0.8471	0.5591	0.8392	0.6991

Table 5.1.: Pre: Precision, Rec: Recall, TC: Tumor Cell, BC: Background Cell.

Chapter 6

Discussion and Outlook

By successfully completing the challenge and achieving a score within the top 6 teams of the OCELOT challenge (as of the 18th of July 2023), we have reason to consider our results as satisfactory. The visual outcomes are particularly impressive, considering the complexity of detecting a large number of targets within each image and the inherent challenges posed by distinguishing tumor cells from background cells. Additionally, it is evident how each applied method has contributed to the final results in a meaningful way. Notably, the segmentation-based label correction technique has effectively utilized tissue information to enhance class prediction. At the same time, the weighted boxes fusion approach has played a crucial role in minimizing false predictions and improving the precision of cell center detection.

Nevertheless, there is still ample opportunity for improvement. Notably, when comparing the mF1 score of our model to that of Ryu et al. [2023]'s various models (as shown in Table 2.1), it becomes apparent that theirs surpasses ours. However, it is worth mentioning that none of the other participants of the OCELOT challenge have achieved higher mF1 scores than those presented by Ryu et al. [2023] so far.

There are still aspects that can be fine-tuned in our model to further enhance its performance. Currently, our YOLOv8 backbone architecture solely utilizes the cell images as input data, with segmentation-based correction applied in post-processing. To push the boundaries further, our following attempt would involve incorporating tissue segmentation as an additional input during training. Additionally, exploring deeper and more complex models for both detection and segmentation holds promise. Regrettably, due to time constraints and limited computational resources, we were unable to conduct extensive GPU-demanding training. However, it's worth noting that the final submission for the challenge is still a few weeks away, granting us an opportunity to continue refining our project. Therefore, we should not consider our work finished at this stage.

In conclusion, our fusion network architecture achieved satisfactory results in the OCELOT Challenge, placing us among the top 6 teams. We accomplished this using models and methods that are computationally inexpensive and well-suited to our limited resources. Tissue information and weighted boxes fusion have been efficiently utilized to improve the predictions. While there is room for improvement compared to a specific competitor's model, no other participant has surpassed their performance. Future work involves incorporating tissue segmentation during detection and exploring deeper models. Our participation contributes to advancing cell-tissue understanding and detection techniques in a computationally efficient manner.

Bibliography

E. Abels, L. Pantanowitz, F. Aeffner, M. D. Zarella, J. van der Laak, M. M. Bui, V. N. Vemuri, A. V. Parwani, J. Gibbs, E. Agosto-Arroyo, A. H. Beck, and C. Kozlowski. Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the digital pathology association. *The Journal of Pathology*, 249(3):286–294, 2019. doi: https://doi.org/10.1002/path.5331. URL https://pathsocjournals.onlinelibrary.wiley.com/doi/abs/10.1002/path.5331.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

Q. Da, X. Huang, Z. Li, Y. Zuo, C. Zhang, J. Liu, and et al. Digestpath: A benchmark dataset with challenge review for the pathological detection and segmentation of digestive-system, 2022. URL https://www.sciencedirect.com/science/article/pii/S1361841522001323.

J. Gamper, N. A. Koohbanani, K. Benes, S. Graham, M. Jahanifar, S. A. Khurram, A. Azam, K. Hewitt, and N. Rajpoot. Pannuke dataset extension, insights and baselines, 2020.

S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

G. Jocher, A. Chaurasia, and J. Qiu. Yolo by ultralytics, 10 2023. URL https://github.com/ultralytics/ultralytics.

D. Karimi, G. Nir, L. Fazli, P. C. Black, L. Goldenberg, and S. E. Salcudean. Deep learning-based gleason grading of prostate cancer from histopathology images—role of multiscale decision aggregation and data augmentation, 2020.

P. Naylor, M. Laé, F. Reyal, and T. Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map, 2019.

OCELOT. Ocelot: Grand challenge, 03 2023. URL https://ocelot2023.grand-challenge.org.

L. Pantanowitz, N. Farahani, and A. Parwani. Whole side imaging in pathology: Advantage, limitations, and emerging perspectives, 06 2015.

Z. Qian, K. Li, M. Lai, E. I.-C. Chang, B. Wei, Y. Fan, and Y. Xu. Transformer based multiple instance learning for weakly supervised histopathology image segmentation, 2022.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016.

ROBOFLOW. What's new in yolov8, 06 2023. URL https://blog.roboflow.com/whats-new-in-yolov8/.

J. Ryu, A. V. Puche, J. Shin, S. Park, B. Brattoli, J. Lee, W. Jung, S. I. Cho, K. Paeng, C.-Y. Ock, D. Yoo, and S. Pereira. Ocelot: Overlapped cell on tissue dataset for histopathology, 2023.

R. Solovyev, W. Wang, and T. Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107, 2021. doi: https://doi.org/10.1016/j.imavis.2021.104117. URL https://www.sciencedirect.com/science/article/pii/S0262885621000226.

J. Terven and D. Cordova-Esparza. A comprehensive review of yolo: From yolov1 and beyond, 2023.

TIGER. Tiger: Grand challenge, 11 2022. URL https://tiger.grand-challenge.org.

Q. D. Vu, S. Graham, M. N. N. To, M. Shaban, T. Qaiser, N. A. Koohbanani, S. A. Khurram, T. Kurc, K. Farahani, T. Zhao, R. Gupta, J. T. Kwak, N. Rajpoot, and J. Saltz. Methods for segmentation and classification of digital microscopy tissue images, 2018.

WANDB. Weights and biases. https://wandb.ai. Accessed: 2023-07-17.

E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers, 10 2021. URL http://arxiv.org/abs/2105.15203.

Appendix A

Time plan

Initial Time Plan

The deadlines for each task are indicated by the dates provided at the beginning of each item.

- 25.04: Conduct the first team meeting to discuss potential topics and methodologies for our project.
- 02.05: Each team member presents one or two ideas for our project.
- 09.05: Unanimously decide to participate in the OCELOT challenge and choose to focus our project on cell detection from cell-tissue interaction. Additionally, plan to conduct literature research on related work.
- 16.05: Present an introductory overview, familiarize ourselves with the data, and assign tasks for the upcoming weeks.
- 01.06: Establish the data structure and set up a shared GitHub repository. Explore the dataset further and complete the literature research.
- 06.06: Identify an image analysis method and develop a model to obtain initial results. Incorporate both tissue and cellular information into the model and create a basic training and validation pipeline.
- 13.06: Present mid-term progress, finalize the image analysis method, and demonstrate a functional software system that performs reasonably well (though with room for improvement).
- 27.06: Submit the draft report, emphasizing the biological relevance of our work. The draft should include an introduction, a review of previous and related work, and an explanation of the data structure and implementation.
- 11.07: Complete the software development and conduct final experiments.
- 18.07: Present the final software product and submit the final report.

Real Time Plan

Below is the updated schedule of our actual time plan. The dates now indicate when we finished the respective tasks. Please note that we list the dates weekly, rather than the exact date, to minimize the number of different points in the schedule below. The changes made in comparison to the initial time plan are highlighted in blue.

- 25.04: We conduct the first meeting with all team members to discuss potential topics and methodologies we may want to apply.
- 02.05: Each team member presents one or two ideas for our project.
- 09.05: We decided to join the OCELOT challenge and focus our project on cell detection from cell-tissue interaction. Additionally, we had already planned to conduct literature research on related work.
- 16.05: We deliver the introductory presentation, obtaining our first data overview and distributing tasks for the upcoming weeks.
- 01.06: We set up the data structure and the shared GitHub repository. We also further explore the dataset.
- 06.06: We find an image analysis method and build a model that provides initial results. Up to this point, we have only utilized cellular information for training. We create a basic training and validation pipeline.
- 13.06: We have mid-term presentations. Our system is functional; however, it performs sub-optimally as it exclusively relies on cellular information for training. We present the intermediate results.
- 27.06: We submit the draft report. The report should include an introduction, previous and related work, and an explanation of the data structure and implementation. We have completed the literature research and incorporated the findings into the report. Additionally, we have included tissue annotations in the post-processing of our model, resulting in noticeable improvements.
- 11.07: We finalize the software and conduct the final experiments.
- 18.07: We present the final software product and submit the final report. We added weighted boxes fusion to our model to improve the performance.

Discussion on Changes

The majority of the changes we made were driven by time constraints and adjustments within our team. Initially, we began as a team of five individuals, but due to scheduling conflicts with other courses, two members chose to withdraw from the project. Consequently, only the three of us remained to handle the model development, programming, literature review, and writing tasks. Let's now briefly discuss all the changes we implemented: The completion of our literature research was delayed as we had limited time available to simultaneously construct the model and obtain intermediate results. Additionally, we needed additional literature to enhance the YOLOv8 model. Due to time constraints, we were unable to implement tissue annotations earlier. As a result, our model's progress by the desired date of June 13th was not as substantial as we had initially anticipated. During the process, we identified certain areas within the preprocessing stage that could be improved. We aim to further explore these aspects and incorporate the enhancements into our model.