DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Biomedical Computing

Deep Learning Based Analysis of Tumor-infiltrating Lymphocytes in H&E Stained Histological Sections for Survival Prediction of Breast Cancer patients

Margaryta Olenchuk

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Biomedical Computing

Deep Learning Based Analysis of Tumor-infiltrating Lymphocytes in H&E Stained Histological Sections for Survival Prediction of Breast Cancer patients

Deep Learning basierte Analyse von tumorinfiltrierenden Lymphozyten in H&E gefärbten histologischen Schnitten zur Überlebensvorhersage von Brustkrebspatienten

Author:	Margaryta Olenchuk
Supervisor:	Prof. Dr. Peter Schüffler
Advisor:	Dr. Philipp Wortmann, Ansh Kapil
Submission Date:	15.12.2022

Contents

1 Data

1.1 Segmentation

The data comes from publicly released Tumor InfiltratinG lymphocytes in breast cancER (TiGER) challenge dataset containing digital pathology images of Her2 positive (Her2+) and Triple Negative (TNBC) breast cancer whole-slide images (WSIs), regions of interest (ROIs), and manual annotations. More specifically, the WSIROIS dataset was used for model training, validation, and testing (see Table 1.1). TiGER data, both at WSI and ROI level, was released at a spacing (pixel size) of approximately 0.5 µm/px, for more information please refer to the original challenge website[1]. The TiGER tissue annotations include eight labels that

Source	Tissue			TILs		
	#slides	#ROIs	median ROI size #pixels [k]	#slides	#ROIs	median ROI size #pixels [k]
TCGA-BRCA	151	151	4 983	124	1744	20
RUMC	26	81	1 312	26	81	1 312
JB	18	54	1 465	18	54	1 465
	195	286		168	1879	

Table 1.1: TiGER data overview. Sources: Cancer Genome Atlas Breast Invasive Carcinoma (TCGA-BRCA), Radboud University Medical Center (RUMC) and Jules Bordet Institute (JB). Tissue slides and ROIs refer to the segmentation images and annotations whereas TILs prefix specifies the data for TILs detection provided by the challenge.

were reduced to three (see Table 1.2). The training masks were generated using available XML files. In the provided mask images, in certain cases, regions not included in ROIs and non-annotated regions in ROIs where marked with the same label, which could not be directly used for training (see Ground truth in Figure 2.5).

While for tissue segmentation the images and their masks could be used as directly extracted from the dataset, the data for TILs segmentation required some preprocessing. The TiGER fixed-size bounding box annotation for lymphocytes and plasma cells (see Table 1.3) was adapted for segmentation by transforming each bounding box into an annotation of the center pixel with dilatation of three.

[1] `https://tiger.grand-challenge.org/Data/`

TiGER Tissue Label	Share	ID	new ID	new Tissue Label
Invasive tumor	0.283	1	1	Tumor
In-situ tumor	0.029	3	1	Tumor
Tumor-associated stroma	0.286	2	2	Stroma
Inflamed stroma	0.096	6	2	Stroma
Necrosis not in-situ	0.048	5	0	Rest
Healthy glands	0.0008	4	0	Rest
Background	0.231	0	0	Rest
Rest	0.026	7	0	Rest

Table 1.2: Reduction of labels provided in TiGER challenge dataset. Resulting labels include three classes: Tumor (1), Stroma (2) and Rest (0) with shares of 0.312, 0.382 and 0.306. Shares were calculated by dividing the number of pixels belonging to some label by the number of the pixel in the current image and averaged over all images.

Source				Number of cells per ROI		
	#slides	#ROIs	#cells	min	max	median
TCGA-BRCA	124	1 744	19 115	0 (44.3%)	206	1
RUMC	26	81	4 728	0 (7.4%)	657	19
JB	18	54	5 523	0 (7.4%)	608	51.5
	168	1 879	29 366			

Table 1.3: Data overview for TILs detection. Sources: Cancer Genome Atlas Breast Invasive Carcinoma (TCGA-BRCA), Radboud University Medical Center (RUMC) and Jules Bordet Institute (JB). Number of cells here refers to the number of bounding boxes that were assigned for lymphocytes and plasma cells, further named TILs.

1.2 Survival Analysis

TiGER challenge aims to assess the prognostic significance of computer-generated TILs scores for predicting survival by applying the Cox proportional hazards model. The survival analysis is done using a large independent test dataset that includes cases from both clinical routine and from a phase 3 clinical trial, which is not directly accessible by participants. The survival analysis within this thesis is done exclusively on publicly available TCGA-BRCA data. Where death (`vital_status` = 1) is considered as an event, and the time until the event or censoring is taken either from `days_to_death` (number of days to death from the first diagnosis) or `days_to_followup` (number of days to last follow-up from first diagnosis).

vital_status	#cases	median age at diagnosis [years]	median time to event [months]
Dead	146	62	37.8
Alive	919	58	26.3
	1065	58	28.7

Table 1.4: Survival data overview.

2 Results and Discussion

2.1 Tissue Segmentation

The three models were trained to segment the RGB input of H&E stained image at 20× magnification (resolution of 0.5 micron-per-pixel) into three prediction maps: tumor, stroma, and rest (not white space, but tissue that is neither tumor nor stroma, for details refer to Table 1.2). The data was separated on the patient level (or slide level, since there is one slide per patient present) into training, validation, and test with 80%, 10%, and 10% accordingly. In order to keep the distribution of the resulting patches numbers (see Table 2.2) and dataset sources fair, the patient separation in Table 2.1 was introduced.

	Train	Validation	Test
TCGA-BRCA	120	16	15
RUMC	20	3	3
JB	16	1	1
	156	20	19

Table 2.1: Split of patients across different medical sources into train, validation, and test sets for segmentation tasks.

The patches were then created using a sliding window approach with 256×256 sized patches and stride equals 128. The additional rotation augmentation was applied, by rotating each patch 5 times at 9 degrees each.

	slides	ROIs	patches	Number of patches that include					
				Tumor	Stroma	Rest	1 class	2 classes	3 classes
Train	156	228	220 567	154 734	172 046	107 506	63 919	99 577	57 071
				(70%)	(78%)	(49%)	(29%)	(45%)	(26%)
Validation	20	25	29 465	16 884	22 187	13 954	11 171	13 028	5 266
				(57%)	(75%)	(47%)	(38%)	(44%)	(18%)
Test	19	33	30 248	18 194	25 630	14 787	8 548	15 037	6 663
				(60%)	(85%)	(49%)	(28%)	(50%)	(22%)
	195	286	280 280						

Table 2.2: Overview of patches that were split into train, validation, and test sets for tissue segmentation. The percentages indicate the fraction of a specific conditioned group of patches to the number of all patches in the "patches" column.

The models were trained with the mmsegmentation toolbox [1]. All models were trained for 160K iterations with Cross Entropy Loss and standard data augmentation techniques including resizing at a random sample scale in the range of (0.5, 2.0), cropping with the maximum 0.75 of a single category present, flipping with 0.5 probability, and application of photometric distortion which includes 0.5 probability for each of the following transformations: random brightness, contrast, saturation, hue and color adjustments. The DeepLabv3+ model was taken as a baseline and trained with ResNet50 backbone, Adam optimizer, learning rate equals 0.0001 and batch size of 64. Whereas the transformer-based SegFormer-B5 (further referred to as SegFormer) architecture was trained once with the same setup of Adam optimizer, 0.0001 learning rate and batch size of 64, and additionally, as in original paper [2], using AdamW optimizer, the learning rate set to an initial value of 0.00006 and then used a poly learning rate schedule with factor 1.0 by default.

Model	FLOPs	Params	Iterations	Runtime	mDice			
					Overall	Tumor	Stroma	Rest
DeepLabv3+	44.16	43.58 M	160 K	1d 12h 2m	**85.25**	85.13	88.07	83.66
SegFormer	12.96	81.97 M	160 K	3d 4h 30m	83.44	83.83	86.32	80.16
SegFormer, AdamW	12.96	81.97 M	160 K	3d 4h 25m	84.93	85.40	87.40	82.00

Table 2.3: Overview of the trained tissue segmentation models. The runtime is given for a training on one GPU NVIDIA A100 SXM4.

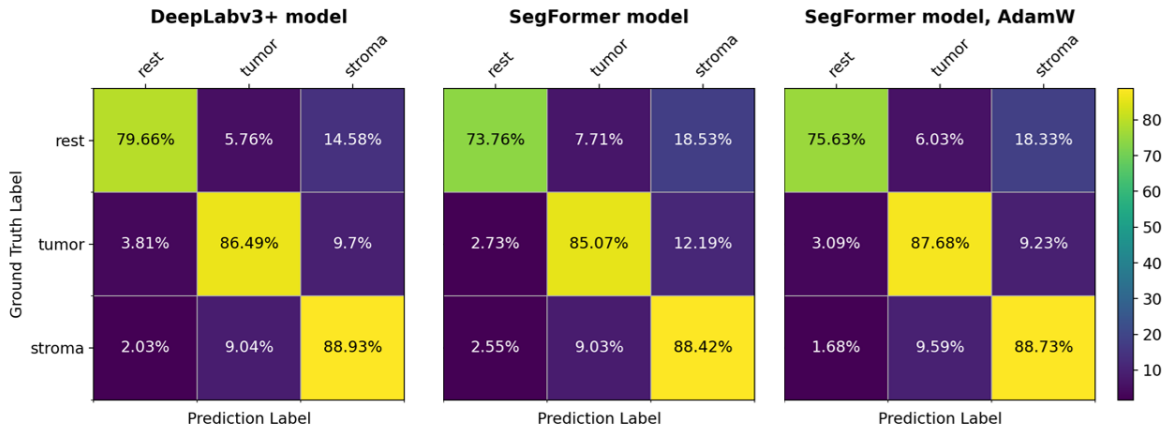

Figure 2.1: Confusion matrices on pixel level for DeepLabv3+, SegFormer and SegFormer with AdamW optimizer based on test set of 32 ROIs.

During the close investigation of test data, one slide (with the prefix TCGA-OL-A5RW-01Z-00-DX1) was excluded from the test set due to an obvious image-mask mismatch. The overall performance of the models, the number of parameters, and the resulting dice score after testing on 32 test images can be found in Table 2.3.

The first thing that catches the eye is the severe runtime difference of SegFormer-based methods compared to the DeepLabv3+ accompanied by doubled number of parameters. None of the SegFormer approaches outperform the DeepLabv3+, but the performance is comparable, which can be also observed in the confusion matrices in Figure 2.1.

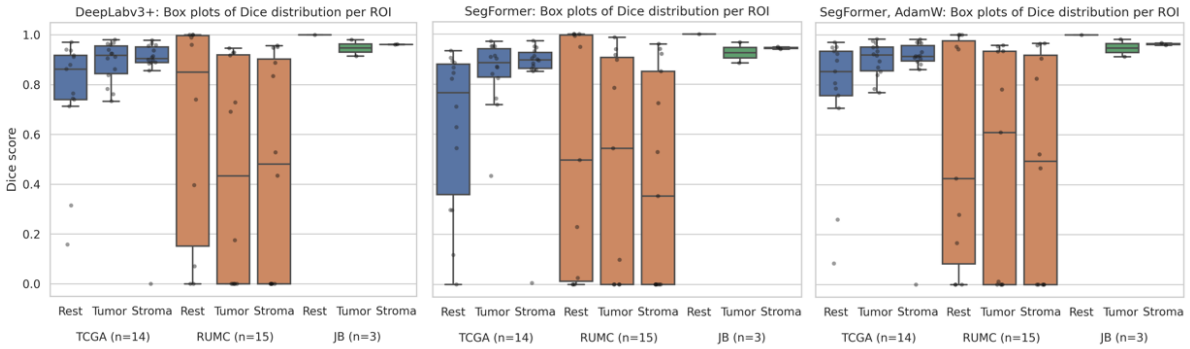

Figure 2.2: Boxplots of pixel wise calculated dice score across three datastes (TCGA-BRCA, RUMC, JB) and three segmentation labels.

Both SegFormer-based methods show difficulty correctly segmenting rest regions, whereas SegFormer AdamW slightly outperforms DeepLabv3+ in the number of true positive detected tumor pixels. As previously mentioned, the dataset originates from three medical institutions which make it reasonable to characterize the performance separately. The boxplots in Figure 2.2 indicate that the performance of the DeepLabv3+ and SegFormer AdamW models in TCGA-BRCA and JB groups are fairly invariant. Whereas the RUMC group accounts not only for the lower performance of SegFormer-based methods in segmenting rest regions but also for an improvement of tumor region segmentation by SegFormer AdamW model both observed in Figure 2.1.

The additional specialty that Figure 2.2 brings to light is the considerable number of RUMC ROIs that have been evaluated with dice scores close to zero across all models. Due to the nature of the Dice score, those can be originated from significant numbers of either false positives, false negatives, or both. According to the boxplots of precision and recall in Figure 2.4 the precision across all models has more close zero values that indicate more frequent false positives. There are clear examples, such as Figure 2.3, where the ground truth includes exclusively rest but all trained models provide multiple class predicitons. Even though there are also opposite examples of regions that were solely annotated as rest and predicted as such (which then lead to occasional dice, precision, and recall equal to one), the issue of false positives, in this case, might be solved only by finding a better dataset split.

A close look at the prediction also revealed that at some cases dice scores might suffer due to some inaccuracies in the annotations. Figure 2.5 showcases that all models were penalized for detecting a rest region inside of the tumor, which was probably learned with some dependency to the presence of white space, which also present in the same slide (the bubles in the lower part of the image) and was annotated as rest.

Nonetheless, there are positive segmentation results present, such as JB ROIs depicted in Figure 2.6 and 2.7 where the performance of SegFormer AdamW is either very close or slightly better then DeepLabv3+ or like in Figure 2.8 where SegFormer AdamW is significantly more accurate. Yet while the SegFormer AdamW model remains promising, due to overall better performance and runtime, this thesis will use the DeepLabv3+ model for further inferences and analysis. For model inference, the model with the best Dice score on the validation set was used.

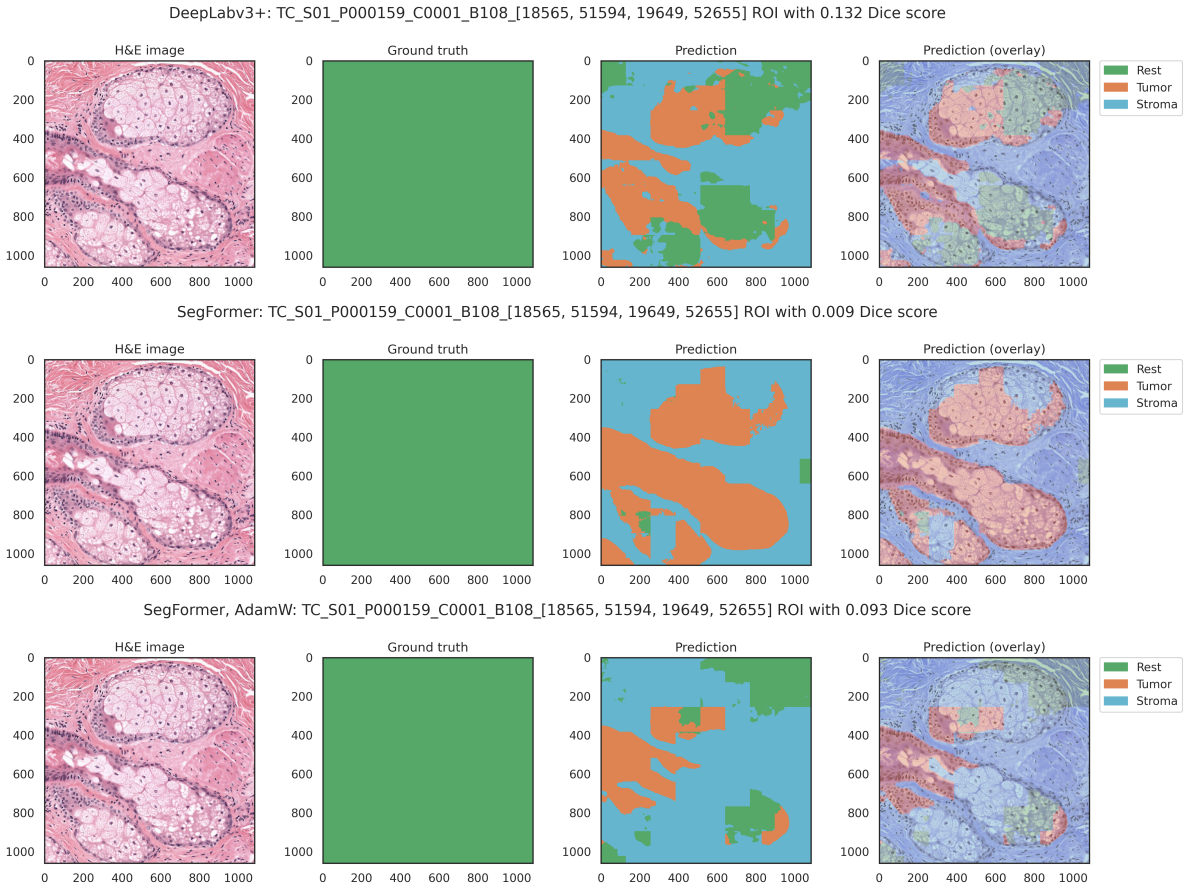

Figure 2.3: Example of rich false positive segmentation RUMC ROI that contributes to the cases of close zero dice scores.

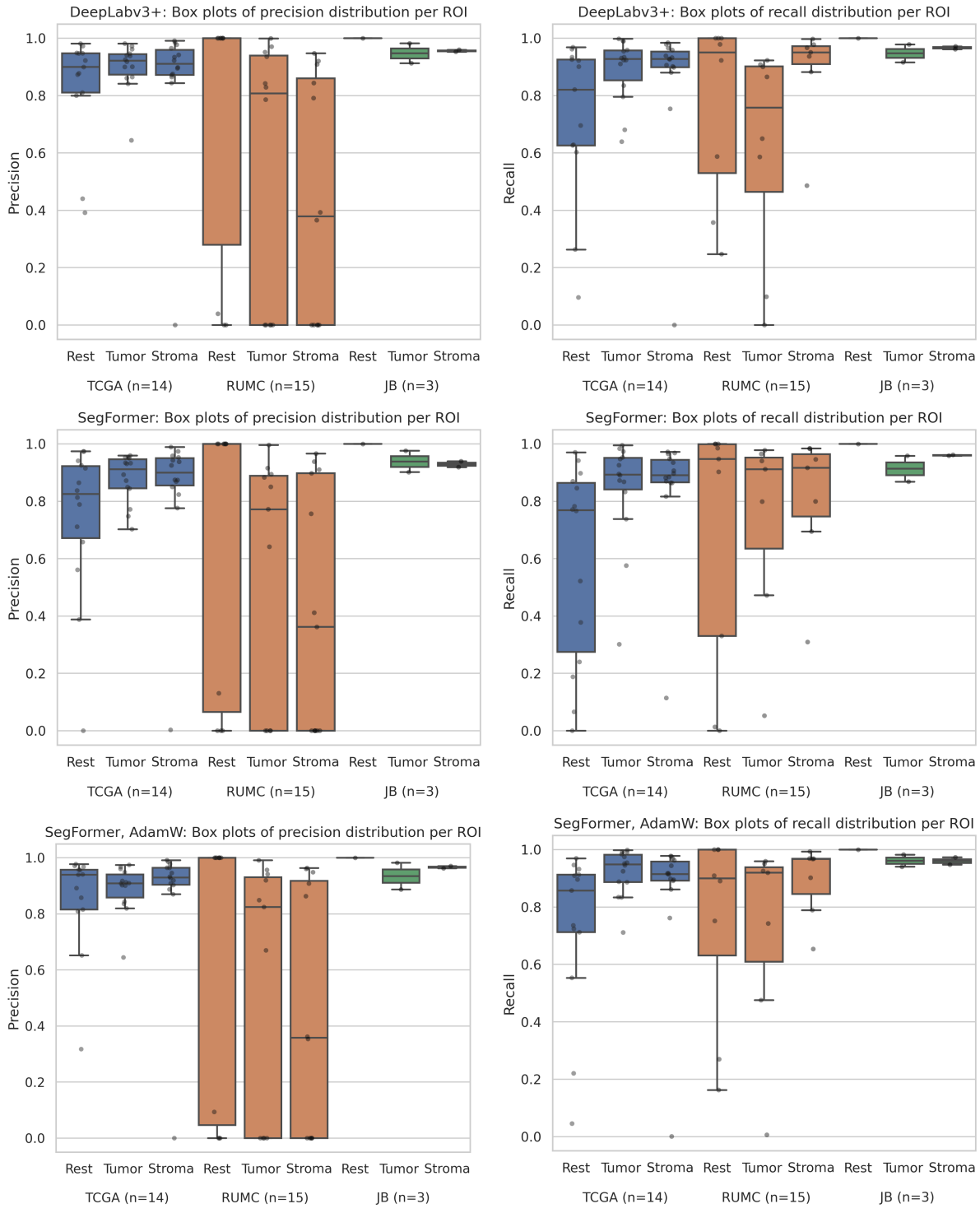

Figure 2.4: Boxplots of pixel wise calculated precision and recall across three datastes (TCGA-BRCA, RUMC, JB) and three segmentation labels.

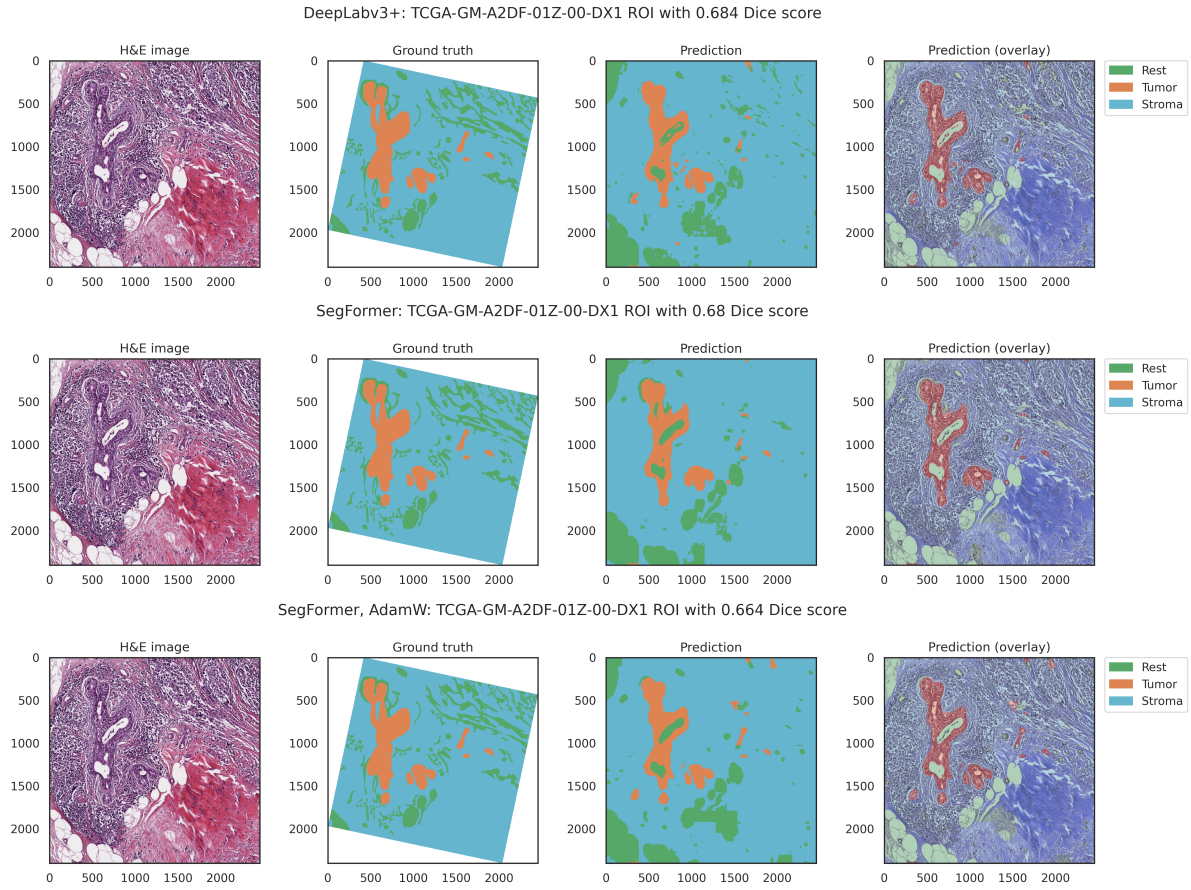

Figure 2.5: Example of a slightly devalued dice score due to some annotation inaccuracies.

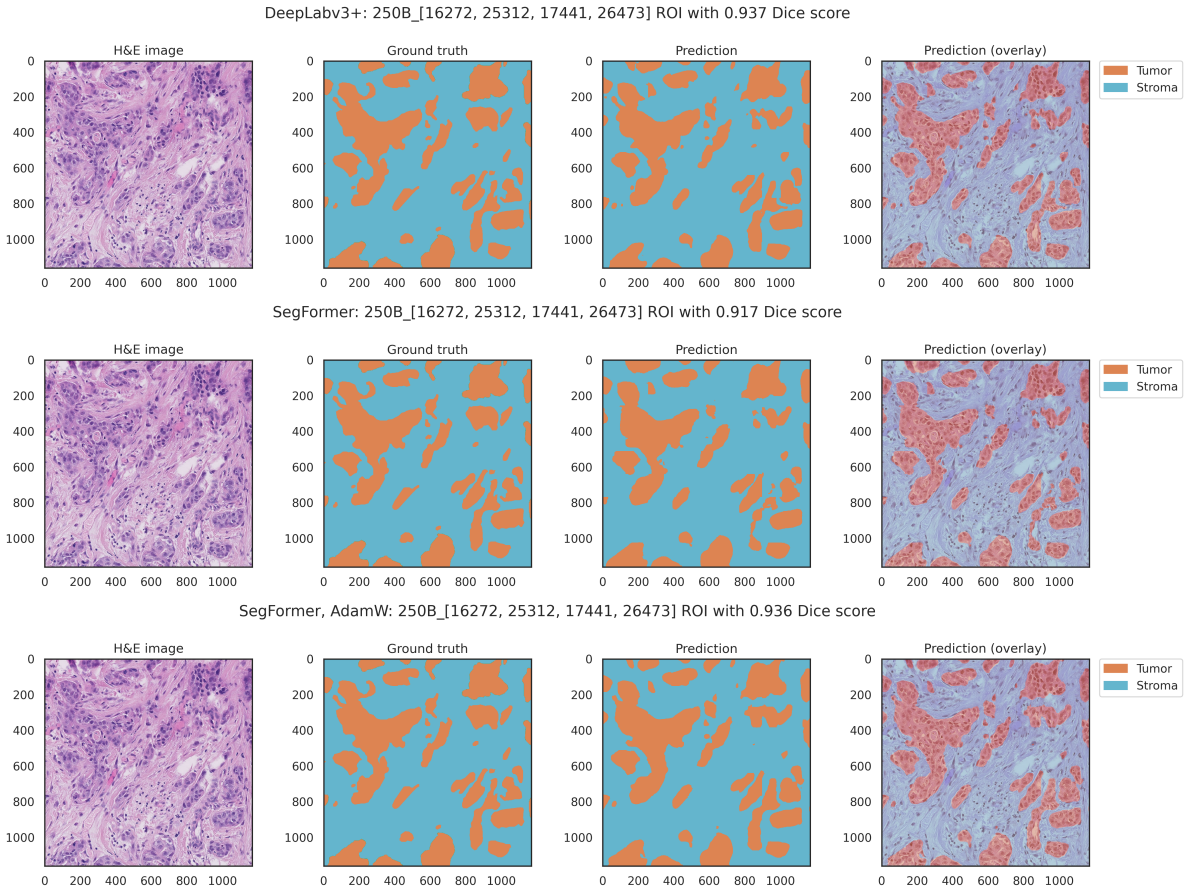

Figure 2.6: JB S_250B ROI segmentation result.

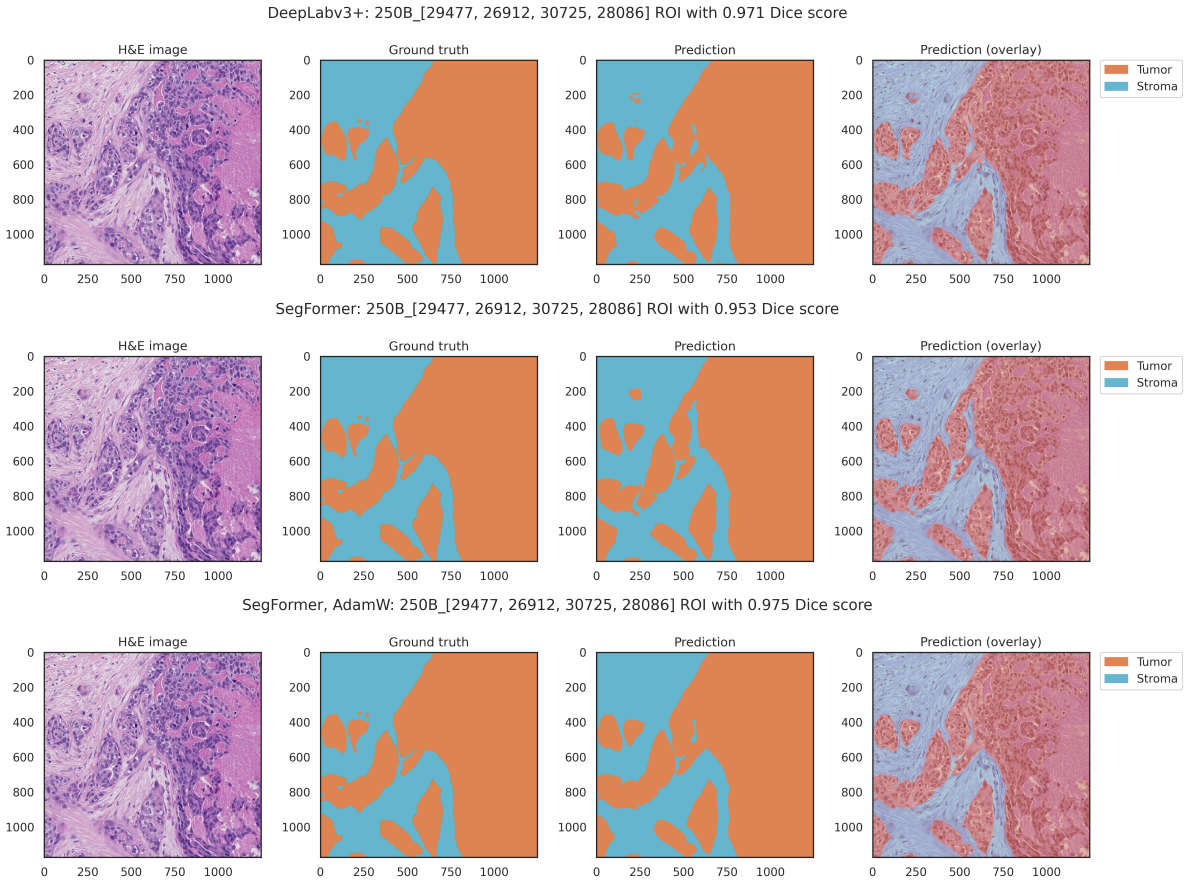

Figure 2.7: JB S_250B ROI segmentation result.

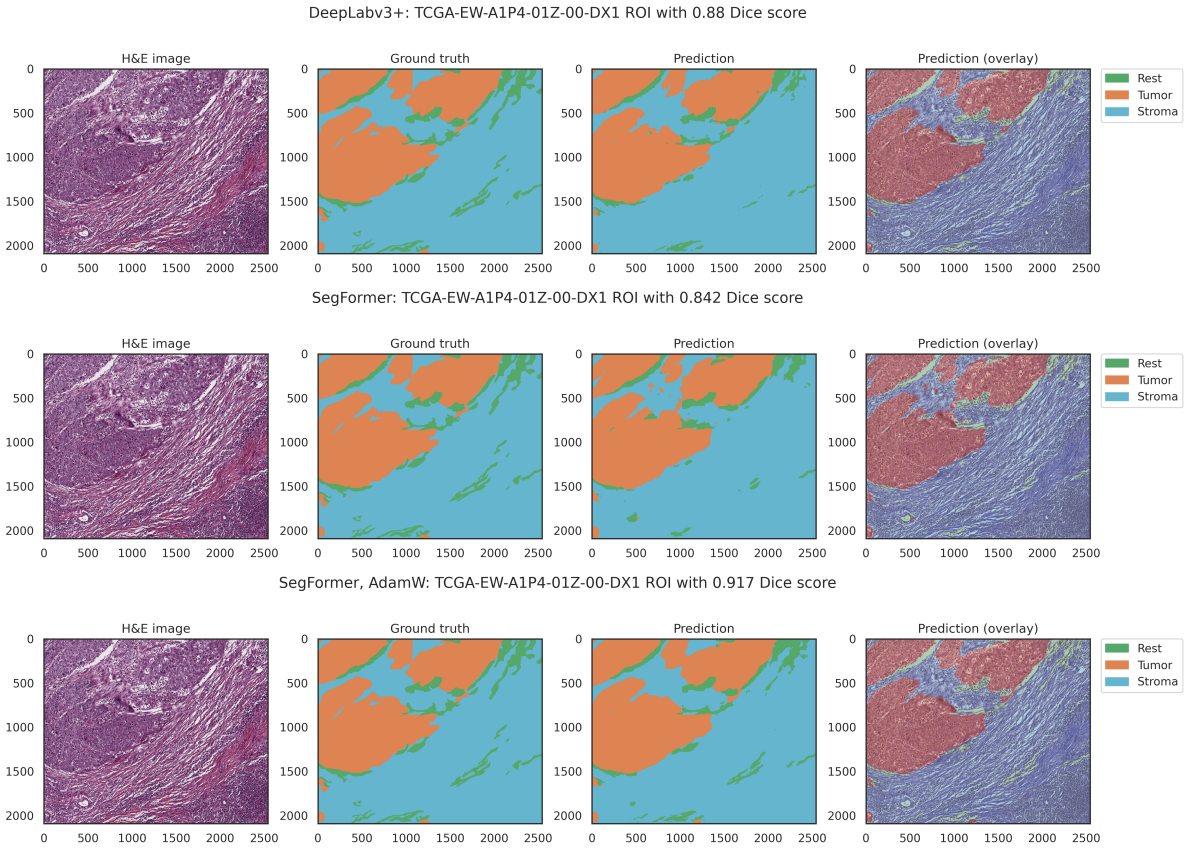

Figure 2.8: TCGA-EW-A1P4-01Z-00-DX1 ROI segmentation result.

2.2 TILs Segmentation

TILs segmentation task aimed to segment the RGB input of H&E stained image at 20× magnification (resolution of 0.5 micron-per-pixel) into two prediction maps: TILs and rest. The split of the patients was used similarly to the previously described in Tabel 2.1. Even though the same patients are present in this data set, the annotations originate from a different study which results in a different ROI and patch statistics shown in Table 2.4. The patches were then created using a sliding window approach with 128×128 sized patches and stride equals 100. The ROIs that were smaller than 128×128 were padded. The additional rotation augmentation was applied, by rotating each patch 5 times at 9 degrees each. The column "Number of patches that include rest" in Table 2.4 seems excessive, but was still added for better data comprehension: the ROIs for TILs segmentation are either completely annotated as rest or include occasional TILs masks.

The model architectures and their parameters were used as described in 2.1: DeepLabv3+, SegFormer with Adam optimizer, and SegFormer with AdamW. An important change is an increased batch size of 128, which was allowed due to smaller patch sizes of 128×128. In the model overview in Table 2.5 the number of parameters and iterations remain the same.

	slides	ROIs	patches	Number of patches that include			
				TILs	Rest	1 class	2 classes
Train	156	1 552	106 974	35 433 (33%)	106 974 (100%)	71 541 (67%)	35 433 (33%)
Validation	20	154	15 518	14 164 (27%)	15 518 (100%)	5 638 (60%)	3 734 (40%)
Test	19	173	9 372	3 734 (40%)	9 372 (100%)	11 354 (73%)	4 164 (27%)
	195	1 879	131 864				

Table 2.4: Overview of patches that were split into train, validation, and test sets for TILs segmentation. The percentages indicate the fraction of a specific conditioned group of patches to the number of all patches in the "patches" column.

FLOPs values are dependent on input shape, therefore it was expected to drop since the input is smaller. It is aslo noticed that for TILs segmentation the runtime between all three models becomes comparable in contrast to tissue segmentation (Table 2.3).

Model	FLOPs	Params	Iterations	Runtime	F1 score	Precision	Recall
DeepLabv3+	11.04	43.58 M	160 K	2d 8h 36m	0.49	0.58	0.43
SegFormer	3.24	81.97 M	160 K	2d 11h 36m	0.62	0.62	0.33
SegFormer, AdamW	3.24	81.97 M	160 K	2d 15h 19m	**0.66**	0.64	0.69

Table 2.5: Overview of the trained TILs segmentation models. The runtime is given for a training on one GPU NVIDIA A100 SXM4.

For proper evaluation the predicted TILs segmentation needed to be reduced to TILs centers (one pixel) that can be then further matched to ground truth. To get the centers of predicted TILs, non-maximum suppression was applied on posterior images that were clipped between 0 and 255. The search for the best-fitted parameter of kappa (threshold) and kernel size was executed on the validation set. As pictured on Figure 2.9 there were multiple experiments performed with kernel sizes in [1, 3, 5, 7, 9, 11, 13] and multiple kappas. The highest value of kappa was the median value over all posteriors. The consecutive values were the two power fractions of the median. Figure 2.9 includes two images for DeepLabv3+ (first in the first row and first in the second row) to provide a zoomed look of the tighter range. As a result, the best parameters on the validation set were chosen as kappa=21, kernel size=9 for DeepLabv3+ and kappa=64, kernel size=5 for SegFormer-based methods. The resulting centers can be then matched to the ground truth by applying the Hungarian algorithm that finds the best assignment to match ground truth TILs with predicted ones. The allowed maximum distance for a match of predicted with ground truth TILs was set to 5 µm.

The final results represented in Table 2.5 show that SegFormer model with AdamW optimizer strongly outperforms DeepLabv3+ and simple SegFormer. Furthermore, while

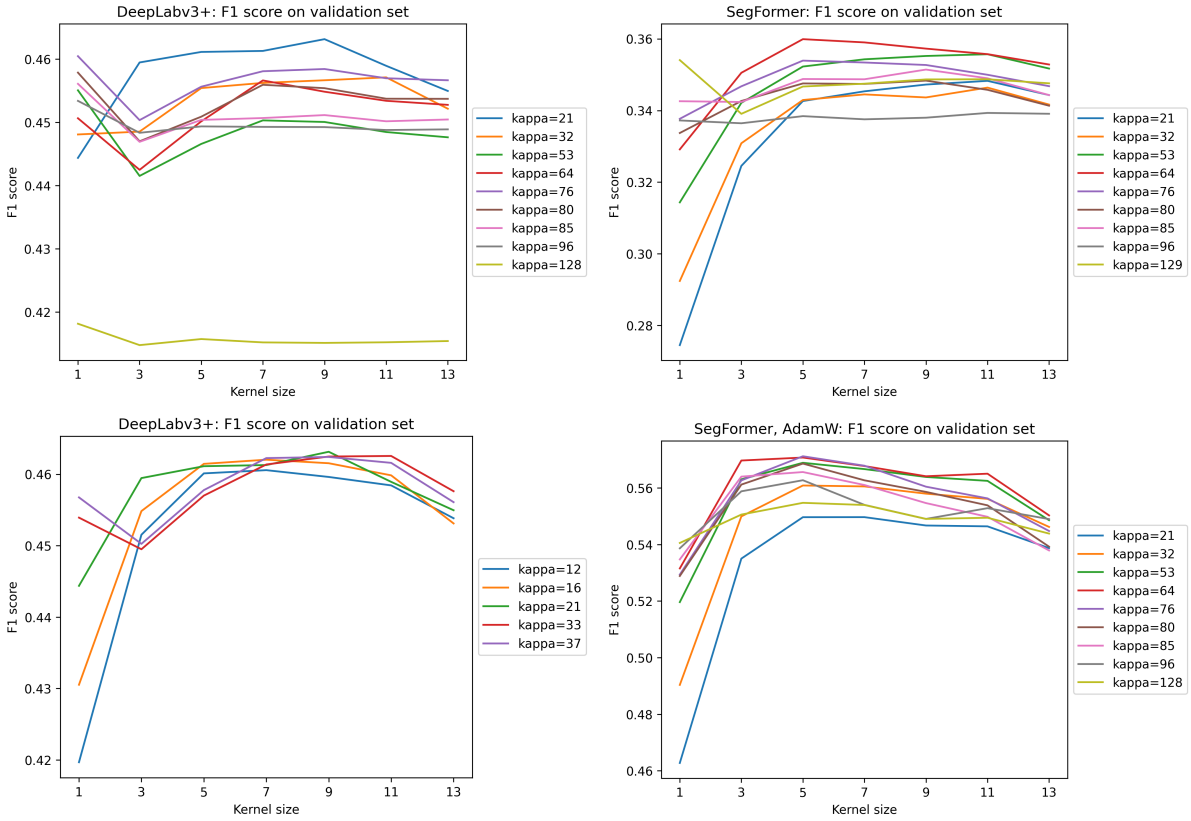

Figure 2.9: Determination process for best kappa and kernel size for DeepLabv3+ (first in the first row and first in the second row), SegFormer (second in the first row), and SegFormer with AdamW (second in the second row).

distinguishing the F1 scores between ROIs originating from different medical centers in Figure 2.10, SegFormer AdamW results show significantly better results in all subgroups.

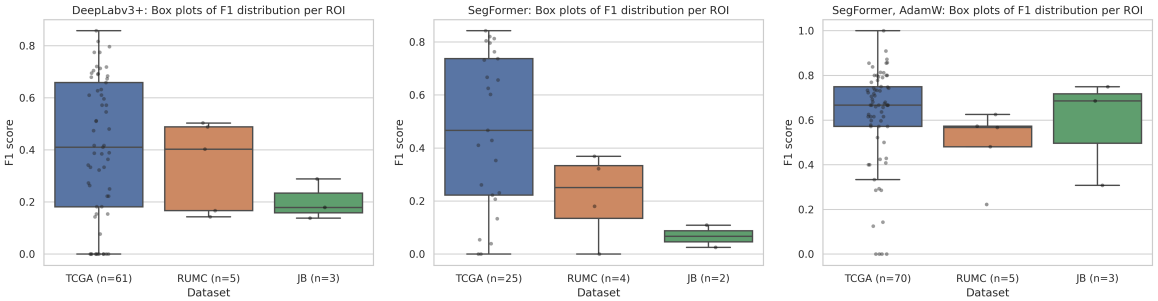

Figure 2.10: Boxplots of dice score across three datastes (TCGA-BRCA, RUMC, JB) with maximum allowed distance between ground truth and prediction equals 10 pixels (5 µm).

Interestingly the precision boxplots in Figure 2.11 do not show such an unequivocal superiority, where in the JB subgroup simple SegFormer manages to predict fewer false positives and (n=2) indicates that the model managed to predict empty ROI as a complete rest region, which is not the case for any other model. In more detailed Figure 2.12 one can see the intermediate steps of how the posteriors are simplified into point segmentations and later the misted, falsely annotated and correct TILs can be compared. Even on the level of posteriors (overlayed with H&E image), it is visible that SegFormer AdamW manages to detect more regions, especially closer to the border of the image. Taking into account all discussions above and the overall better performance, the SegFromer AdamW was considered the best model, and the model with the highest pixel-wise dice score on the validation set was taken to the next step.

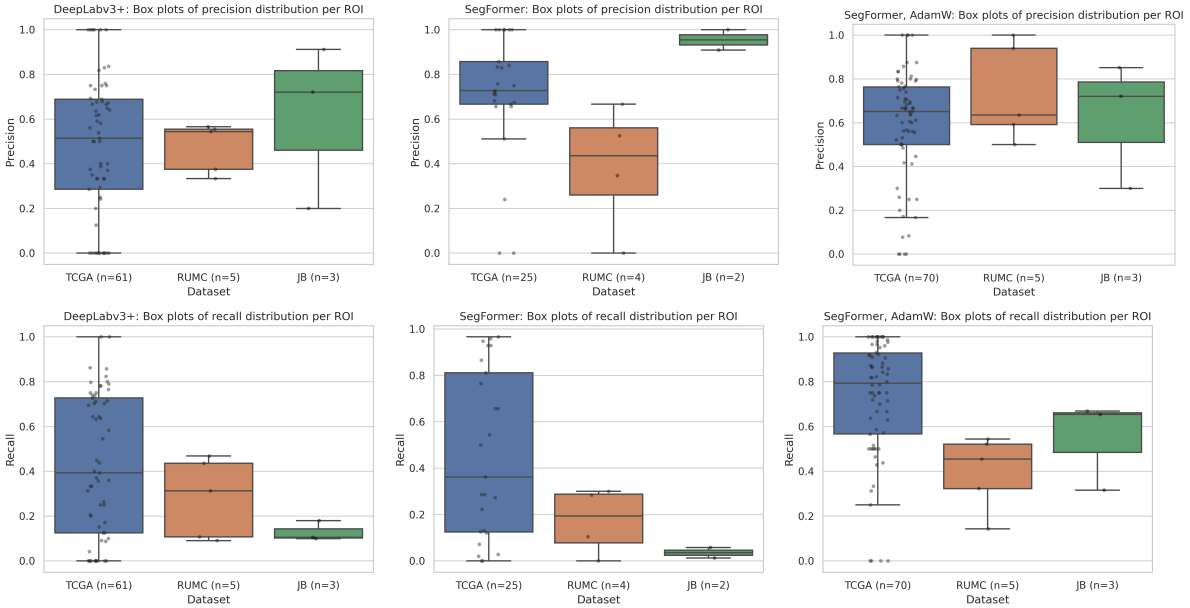

Figure 2.11: Boxplots of precision and recall across three datastes (TCGA-BRCA, RUMC, JB) with maximum allowed distance between ground truth and prediction equals 10 pixels (5 µm).

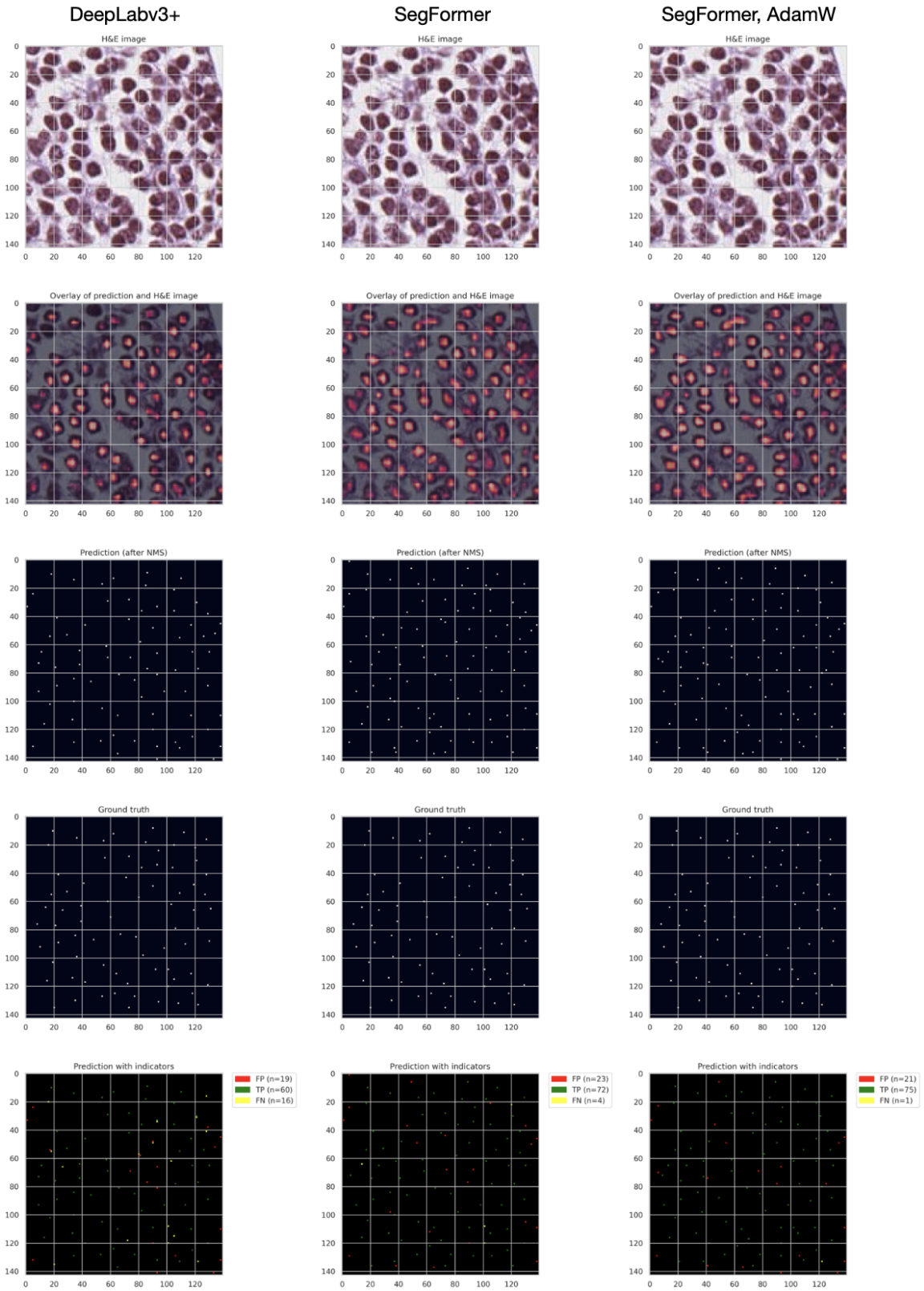

Figure 2.12: TCGA-D8-A142-01Z-00-DX12 TILs prediction with DeepLabv3+, SegFormer and SegFormer, AdamW. With 0.774, 0.842 and 0.872 F1 scores accordingly.

List of Figures

List of Tables

Bibliography

[1] M. Contributors. *MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark*. `https://github.com/open-mmlab/mmsegmentation`. 2020.

[2] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.