UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Data Science Curriculum

Arnel Pällo

Measuring Testis Tubule Wall Thickness in Histopathology Images

Master's Thesis (15 ECTS)

Supervisor: Dmytro Fishman, PhD

Tartu 2023

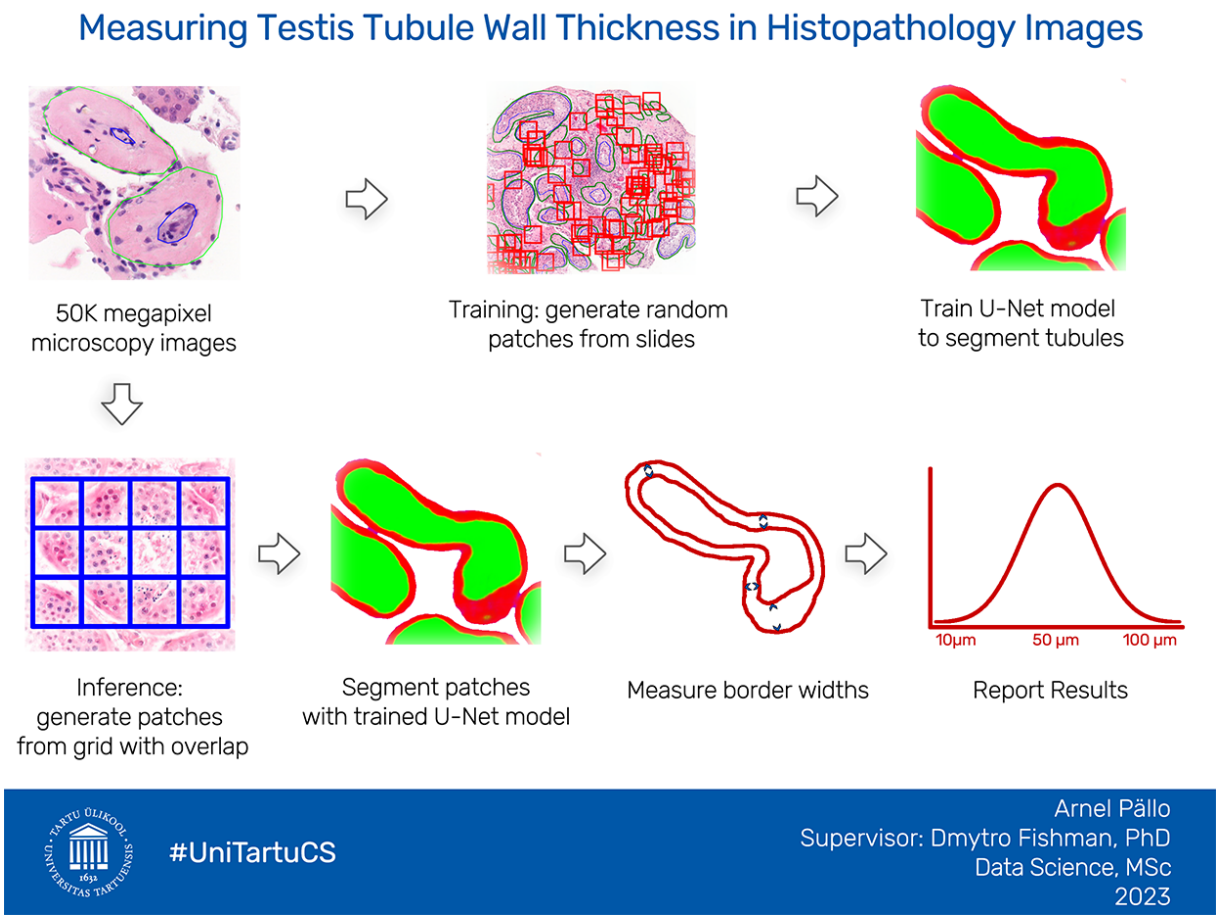
Measuring Testis Tubule Wall Thickness in Histopathology Images

Abstract

One of many causes of infertility is too thick tubule walls in male testis, locking in the sperm cells. In this thesis we have developed a machine-learning-powered software pipeline for analysing testis histopathology images. The software identifies the tubules and measures their wall thicknesses, allowing medical professionals to draw conclusions and/or perform additional follow-up analysis as needed. Our value proposition is in a clear focus on practical application. The software is designed and trained for usage on large-format (50 000 megapixels) testis tissue samples, measuring specific abnormalities. It is the author's desire that the software pipeline could be used by medical facilities in Estonia on real patients, providing real value, actually helping people and making a difference.

Keywords: Deep learning, Medical image segmentation, Computer Vision, Image processing

CERCS: P170 Computer science, numerical analysis, systems, control; P176 - Artificial Intelligence; T111 - Imaging, image processing; T115 Medical technology



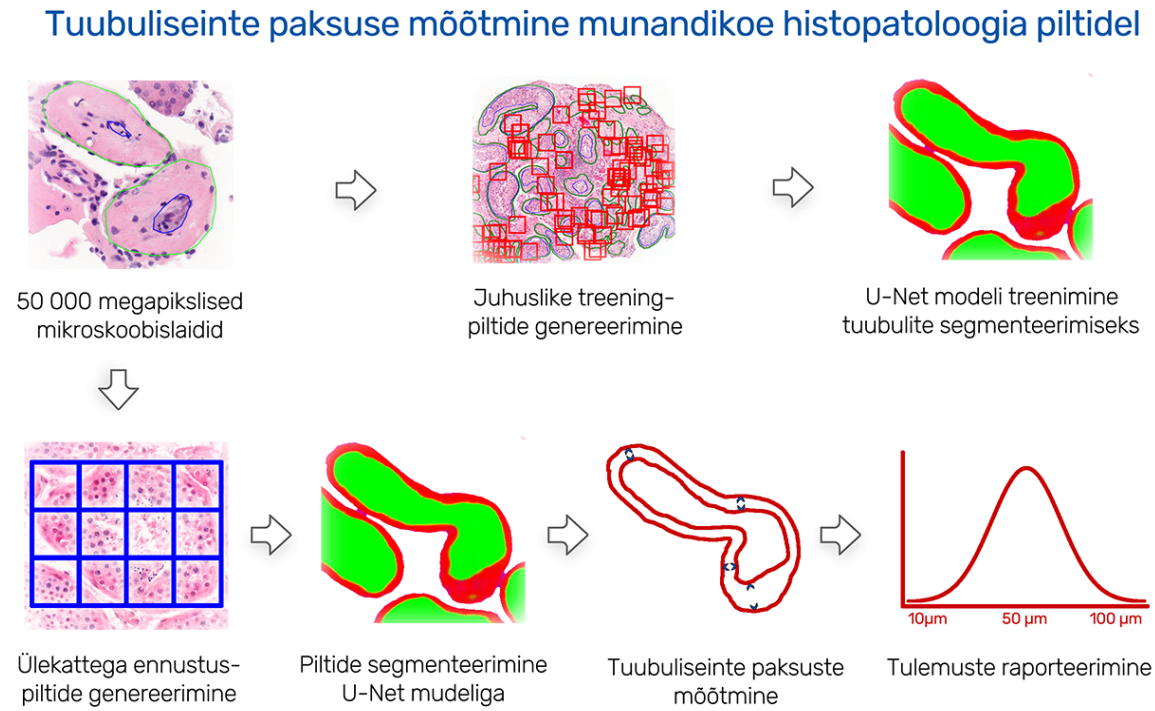
Tuubuliseinte paksuse mõõtmine munandikoe histopatoloogia piltidel

Lühikokkuvõte

Üheks meeste viljatuse põhjuseks on munandites paiknevate tuubulite liiga paksud seinad, mis ei lase seemnerakkudel sealt väljuda. Käesolevas magistritöös arendasime välja masinõppel baseeruva tarkvarapaketi munandite histopatoloogia piltide analüüsimiseks. Tarkvara tuvastab piltidelt tuubulid ja mõõdab nende seinte paksused, hõlbustades meditsiinipersonalil otsuste langetamist ja vajadusel täiendavate uuringute läbiviimist. Töö põhiliseks väärtuseks on selge fookus praktilisel rakendatavusel. Tarkvara on disainitud ja treenitud töötama suureformaadiliste (50 000 megapikslit) koepiltide peal, otsides sealt kitsalt määratletud spetsiifilisi anomaaliaid. Autori sooviks on, et loodud tarkvara on võimalik juurutada Eesti meditsiiniasutustes, tuues patsientidele reaalset kasu.

Märksõnad: Tehisnärvivõrgud, meditsiini piltide segmenteerimine, masinnägemine, pilditöötlus

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria) ; P176 - Tehisintellekt; T111 - Pilditehnika; T115 Meditsiinitehnika



Contents

1 Introduction

Infertility affects 8–12% of couples worldwide, of which 40-50% is due to the “male factor” (Kumar and Singh, 2015). It has many causes, such as low sperm concentration, poor sperm motility, or abnormal morphology. In case of low sperm concentration, one of the next steps is taking a tissue sample of the patient’s testis to check for abnormalities. Medical imaging such as histopathology plays a central role in this diagnosing process.

However, analysing the images has been done manually by expert humans, mostly pathologists. Due to the size of the slides and the amount of information, a pathologist only investigates a small portion of the whole image slide. This introduces subjectivity and loss of precision. Even with these shortcuts it is a long laborious process.

Due to the high workload and limited resources, a large number of images cannot be thoroughly analysed. One must carefully assess the necessity and cost of such a procedure beforehand. This can lead to long waiting times and missed diagnosis. It also prevents widespread screening of patients for early signs of diseases. There is also an opportunity cost: the time spent on analysing images is not spent on other necessary medical tasks (and vice versa).

1.1 Contribution

In this thesis we have developed a machine-learning powered software pipeline for automatic processing of testis tissue histopathology images. Our focus is limited to searching for and identifying one particular abnormality, namely too thick tubule walls.

The software, once trained, does not require additional work to operate. Compared to the manual labour by the expert, it only takes minutes to analyse one huge slide and can be scaled up much more easily as it can be run 24/7 with additional computing resources. Ideally it could be integrated into automatic workflows, where every testis slide-scan is processed by the proposed software. This is in stark contrast with the state-of-the-art which requires manual human labour, where it takes 30 minutes for one human to analyse a small portion of one slide, in addition to time spent training the medical professionals.

Using machine learning in medical imaging is not new. There are many existing models suitable for analysing various medical images (such as computed tomography scans, X-rays, etc.). Although the thesis does not introduce any new machine learning models nor architectures, it presents a thorough pipeline for applying deep learning in solving a particular real world problem.

The value proposition is in a clear focus on practical application. It is the author’s desire that the software pipeline could be used by medical facilities in Estonia on real patients, providing real value, actually helping people and making a difference.

At the time of writing the author is not aware of comparable fully automated solutions that analyse testis tissue being deployed to real-world usage.

The software is made available in a public GitHub repository and can be found at https://github.com/arnelism/testis_thesis/.

1.2 Outline

Chapter 2 (Background) gives a short overview of the histopathology process, introduces existing computer-aided solutions for analysing testis tissue and defines the scope and motivation of our work.

Chapter 3 (Methods and Data) describes the whole software pipeline from pre-processing to post-processing in high detail.

Chapter 4 (Experiments and results) describes the hyperparameter space we explored in tuning the model, explains which configurations worked best and presents the wall thickness measurement results of the whole pipeline.

Chapter 5 (Discussion and Conclusions) assesses the success of the software pipeline, admits the shortcomings of current work and proposes follow-up work that should be undertaken in order to deploy the software to real-world usage.

2 Background

According to our clinical collaborators, testis tissue contains tubules where sperm cells are "born". If the tubules' walls are too thick (5.3 +/- 1 micrometres is normal), sperm cells cannot exit them, thus reducing their count and leading to infertility (Figure 1). The goal of a pathologist is determining how many tubules are "unhealthy" in that regard.

In the process of histopathology, tissue samples are collected and processed in several steps into glass slides. They are then analysed under a microscope, scanned and converted into large digital images. In the samples made available to us, one slide is over over 50 000 megapixels, (i.e. 350K x 150K pixels in resolution), contains several thin extracts of the tissue and includes hundreds of tubules. The pathologist identifies them, measures their wall thicknesses and derives statistical conclusions. Every single slide requires about 30 minutes to investigate according to our clinical collaborators.

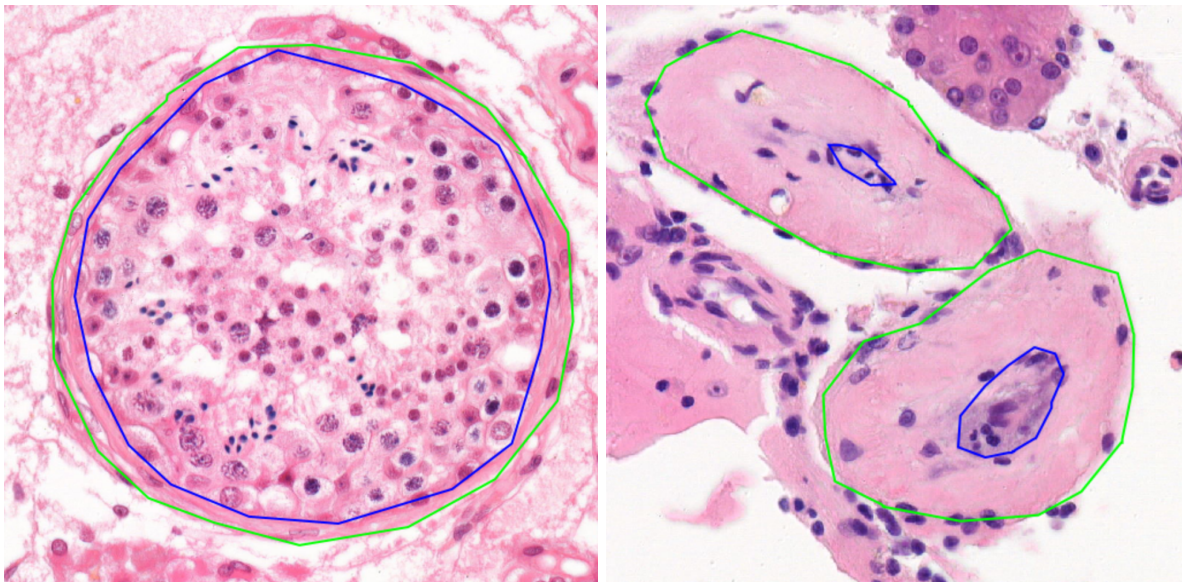


Figure 1: visualisation of different kinds of tubules. Left: healthy tubule with a moderately thin wall Right: thick tubule walls, hence the sperm cells cannot exit..

If the tubule walls' thickness is the sole cause of infertility, then there is hope - healthy sperm cells can be extracted and used in artificial fertilisation (Godart and Turek, 2020). Conversely - if the tubules are healthy, it might be a bigger problem for a patient: for an infertile patient this indicates that the diagnosis is something else and potentially incurable, such as an inability to create viable sperm cells.

Pathologists are measuring more than just tubule wall thicknesses in those images. According to our clinical collaborators, they are also interested in the size and number of the tubules as well as the amount of sperm cells inside them. Our pipeline can be adapted to measure those

characteristics as well but is not a focus of this thesis (although it is a great candidate for further research and development).

There is a wide body of research on using machine learning for medical image analysis. In particular, U-Net (Ronneberger, Fischer and Brox, 2015) is a well-performing model created specifically for medical image segmentation tasks that has been successfully used in a range of biomedical applications from microscopy image segmentation (Siddique *et al.*, 2021), to cancer detection in computed tomography scans (Saood and Hatem, 2021). Similarly to autoencoders, it has an encoder-decoder architecture. The encoder portion of the network reduces the input's dimensionality, while the decoder portion upscales it and constructs a segmentation mask. However, unlike autoencoders, U-Net also has multiple skip connections between the encoder and decoder (concatenated together), allowing the network to retain high-resolution information (Figure 2).

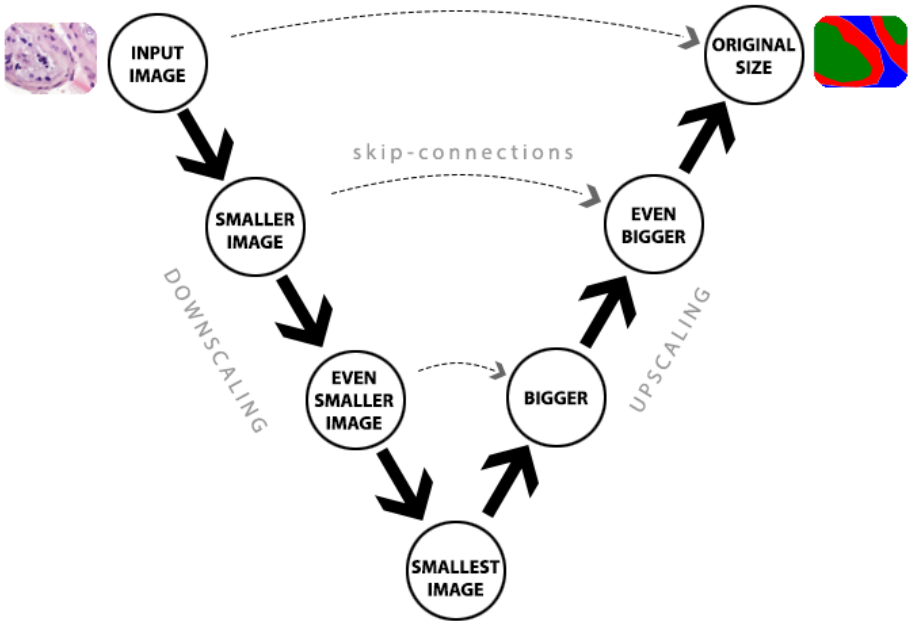


Figure 2: U-Net Architecture

In (Fakhrzadeh *et al.*, 2023) the authors describe a ResNet based alternative to U-Net for segmenting the testis tissue in sexually mature minks. However, their paper is focused only on the proposed model and not in end-to-end application. Also, the fidelity of their images was orders of magnitude lower: according to the article, image dimensions were at 1200×1600 pixels and 0.4 mm per pixel. In contrast, our slides measure 150 000 x 450 000 pixels, and 0.0001 mm per pixel.

Alternatively, in (Sziva *et al.*, 2022) the authors describe a quantitative mathematical methodology for the analysis of testicular tissue, without using deep learning (but the authors

concluded that their technique is apt to be subjected to further automation with machine learning and artificial intelligence).

We did not find any existing literature focused on measuring tubule wall thicknesses in segmentation maps of testis tissue. Since our proposed algorithm (which works on the segmentation masks) is relatively simple and straightforward (less than 100 lines of code), we theorise that such algorithms are an implicit part of similar software pipelines and not separately published.

Our focus is on building an end-to-end solution which receives full-resolution (50K megapixels) microscopy images as input and reports tubule walls' thicknesses as output. Thus, in our work we have developed a software pipeline to automate that process. The software uses U-net to identify the tubule structures, measures their wall thicknesses and reports the measurements.

3 Methods and Data

In this chapter we are describing all the steps of the software pipeline in great detail and discuss the design tradeoffs where applicable. We are putting a particular emphasis on pre-processing of the images because in our opinion it is the most critical component.

The software pipeline proposed in this work consists of multiple steps. First, pre-processing converts the huge image sources into a form suitable for further analysis. Secondly, a neural network creates a segmentation map from the pre-processed images. This is followed by post-processing and cleanup of the model output. Finally, tubules' walls are measured on the clean segmentation and the final results are reported.

3.1 Dataset

The inputs to the software are digital microscopy images from clinical collaborators of East Tallinn Central Hospital (figure 3). They consist of digital scans in MIRAX Virtual Slide File (.mrxs) format and ground truth annotations in geojson files. An .mrxs file is an image file created by MIRAX-compatible microscope digital slide scanners ('MIRAX Virtual Slide File', 2020). It consists of an index and a thumbnail stored in .mrxs file, accompanied by the raw data in numerous .dat files. It stores images of specimen samples from glass slides on a digitised virtual slide. It allows extracting images in 11 different zoom levels (level 0: the original image, typically thousands of megapixels; level10: tiny thumbnail). The difference between each adjacent zoom level is 2x in both dimensions.

We are using QuPath software for working with the .mrxs files and annotations. QuPath is an open-source software for digital pathology image analysis (Bankhead *et al.*, 2017). It allows exploration (viewing, zooming, panning) and annotation of the digital slides. We are also using QuPath for defining training, validation and test regions of the slides. In addition to providing annotation capabilities, the usage of the software is necessary because standard operating system tools do not support .mrxs files nor the size of the images in question (one slide is over 50 000 megapixels, i.e. 350K x 150K pixels in resolution).

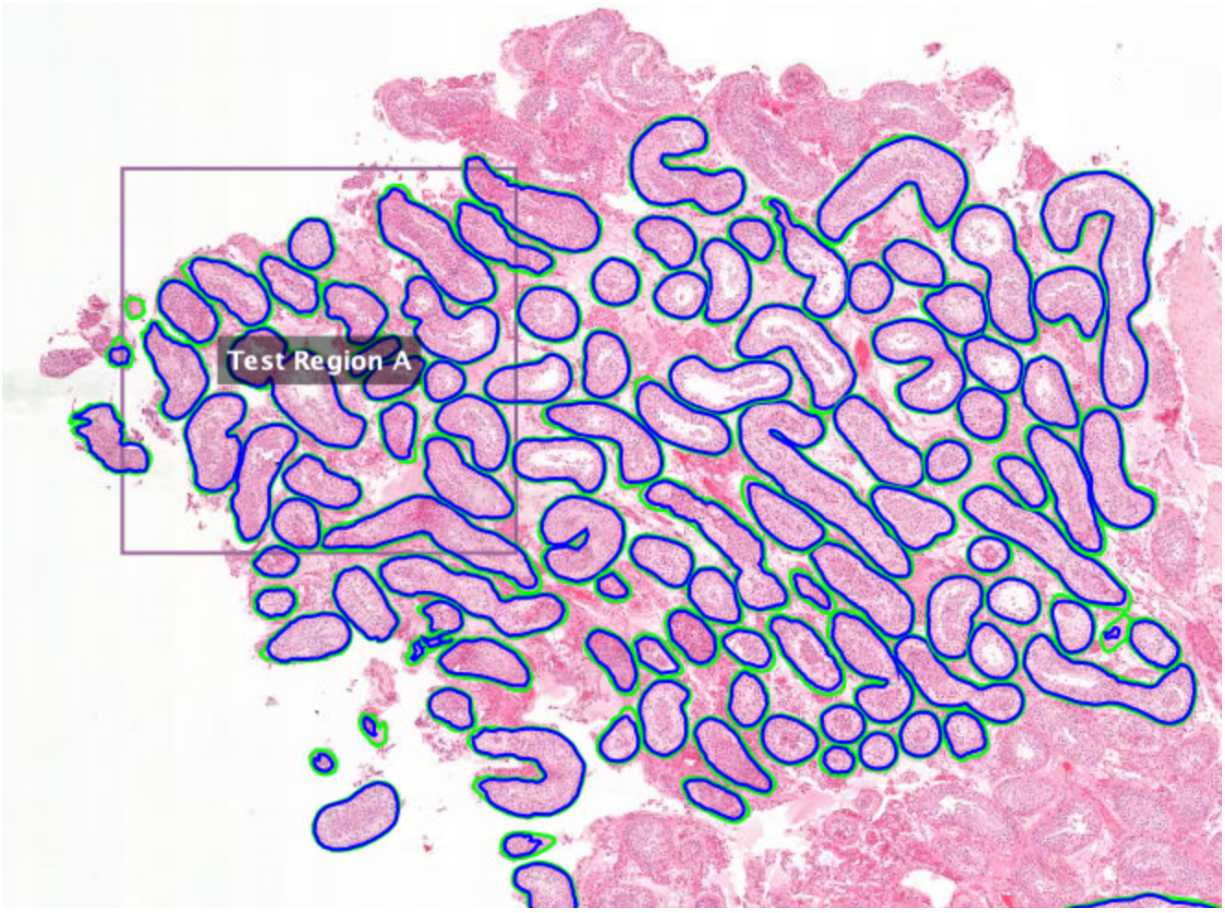


Figure 3: Screenshot of an annotated tubule slide in QuPath. Tubule bodies (blue) and borders(green) are annotated by clinical collaborators. “Test region A” is an area defined by the author for model validation. Tubules outside the test region are used as training data.

3.2 Pre-processing

Pre-processing was the main emphasis of our efforts because we believed that is what determines the success of the project. An important part of the pipeline is image segmentation performed by a neural network. Machine learning models consume small images (from 256px to 512px) as input. However, the images we needed to analyse were somewhat larger: around 50 000 megapixels (150 000 x 350 000 px). It is infeasible to alter the U-Net model to accept inputs of that dimensionality. Instead, we implemented a pre-processing pipeline which allows us to bypass that limit and process the slides.

We were using “openslide” Python library (*OpenSlide*, 2015) for reading and extracting images from the MIRAX virtual slides in our software code. Since geojson libraries are in JSON format, Python standard library is sufficient for working with them.

Since the original images are too big to be processed directly by the neural networks, we split them into thousands of patches (512x512px) which are then used as an input for the

segmentation model. The model is trained to segment these small regions. In order to increase the quantity and variety of training data, input patches are generated randomly from the slides (as opposed to a regular grid) (Figure 5). In addition to providing a larger amount of training data, it ensures that the model does not learn to depend on any particular slicing of the source image (any tubule structure can appear at any part of an image). Both training and validation data was sampled the same way, but from distinctly separated regions.

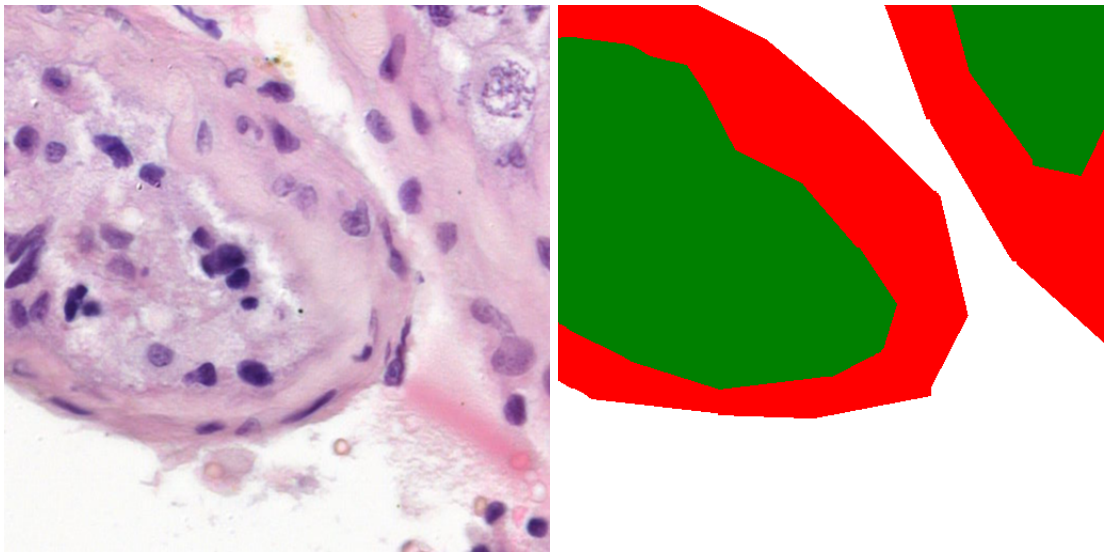


Figure 4: Original histopathology image (left) and the corresponding ground truth segmentation map (right), where green regions indicate tubules, the red pixels correspond to the tubule walls (and white means background).

Additionally, several augmentations are applied to the training patches such as horizontal and vertical flips and 90, 180 or 270 degree rotations. The augmentations reduce overfitting by increasing the amount and variety of training data available. Without using them it is common to experience low training but large validation errors during model fitting.

While there are numerous image augmentation options available, one must carefully pick which ones are suitable. For example, flipping and rotating images result in pictures that are similar to the original training data. However, augmentations such as zooming in and out (cells have relatively stable size for a given zoom level), adjusting brightness (scans have very uniform and well specified lighting) or object occlusion (model is consuming a 2d slice of a specimen sample, not a photographic image) would not be useful (likely even harmful) when working with histopathology slides, because they introduce non-realistic variety to the training data which is not found in actual images.

Obtaining annotated training data was one of the largest challenges. We received three tissue slides (henceforth referred to as alpha, beta and gamma). Two of them were fully annotated by

the clinical collaborators and one only partially. We continued annotating that one by ourselves and after 12 hours of effort the slide was still less than 50% complete.

Thus, an additional pre-processing strategy was introduced: avoid generating training patches from non-annotated regions, because that would confuse the model (ground truth would indicate that there is no tubule where in reality there actually was). This was confirmed by initial training results. At first, when the slides were more sparsely annotated, generating patches with higher minimum annotation overlap (every random patch must be at least 60% tubule by area) yielded lower validation loss while later models trained with more thoroughly annotated slides performed better with 10% minimum tubule area.

Training slides from three separate histopathology slides were collected into a single training set of 12 000 patches (4000 images from each, with augmentations). Validation set was collected similarly and consists of 1200 patches.

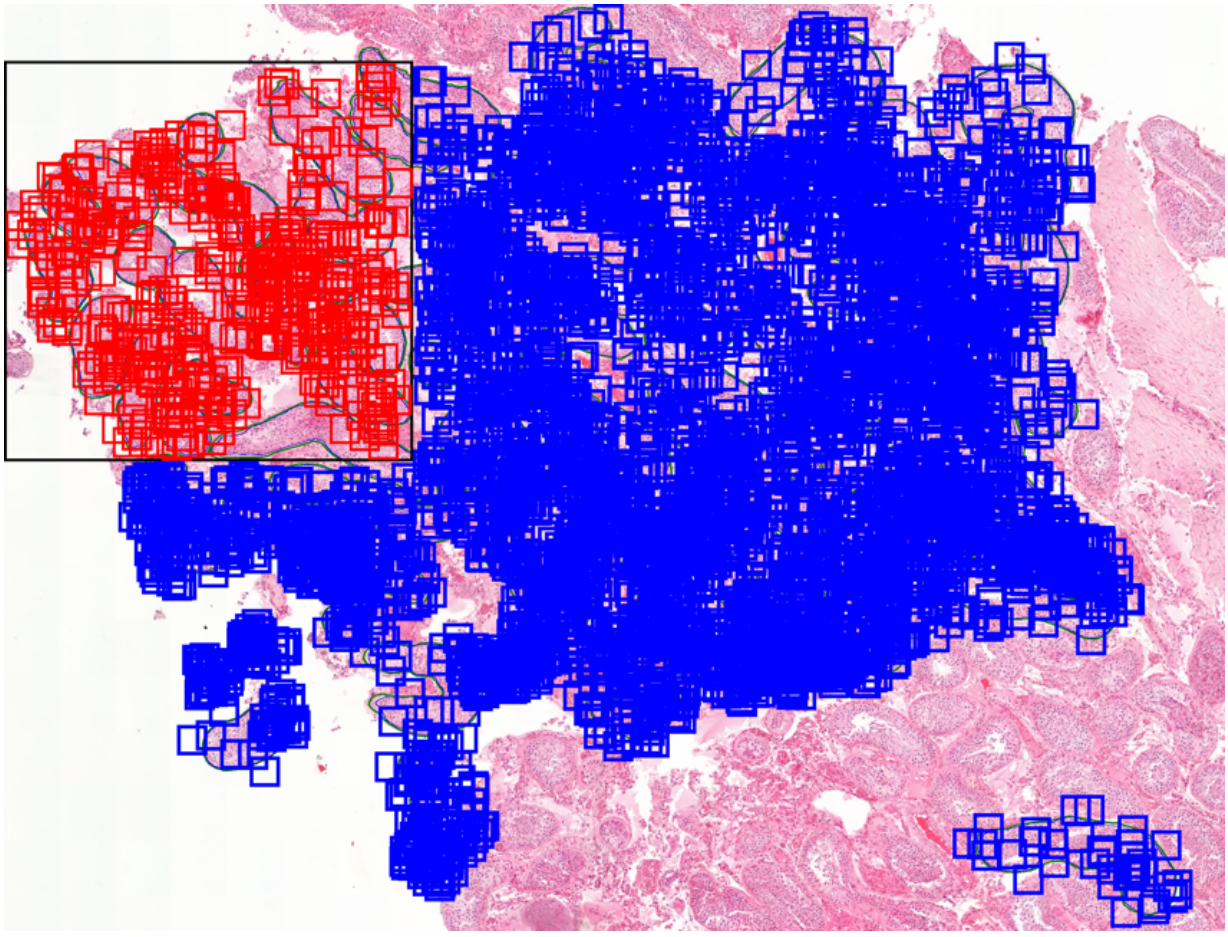


Figure 5: Random sampling from one of the slides. Validation patches are drawn in red colour, training ones in blue. Data sampled only from annotated areas, strict separation between training and test data. Due to the randomness, the same tubule can appear in multiple distinct patches, increasing the model's performance.

Random-slicing described above is not used during inference. For that purpose, the patches are generated from a regular grid placement because the model's outputs are joined back together to

make one large image. Patches used during inference have been extracted with a certain adjustable overlap (Figure 6). Predictions from only the middle part are used during reconstruction and the edges are discarded. The assumption was that predictions near the edges are less precise. The whole patch is used by the model in order to saturate its perceptive region, because predictions at the edges have lower quality. Using overlap adds a buffer around each patch, thus moving the edge further away. Similar strategy for improving segmentation performance was described in the U-Net paper (Ronneberger, Fischer and Brox, 2015). In that paper, the authors cloned a buffer around the edges of the input image which improved segmentation accuracy at the (original) edges.

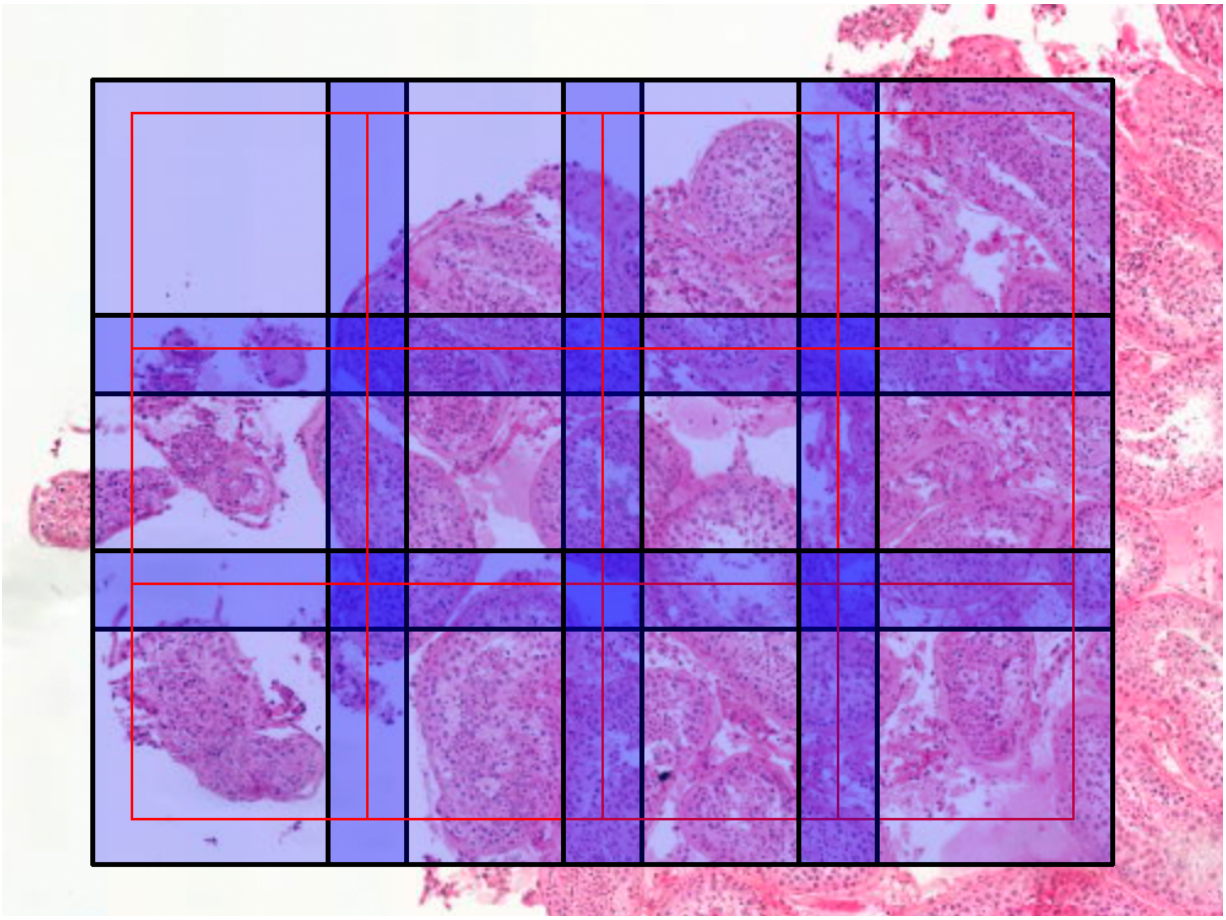


Figure 6: Grid placement of inference patches.. They are denoted in light blue with black borders. Usable region from each patch is marked by the red border. Two adjacent images have 25% overlap, but their used regions touch precisely. Patch sizes are not to scale with regards to the pathology image below.

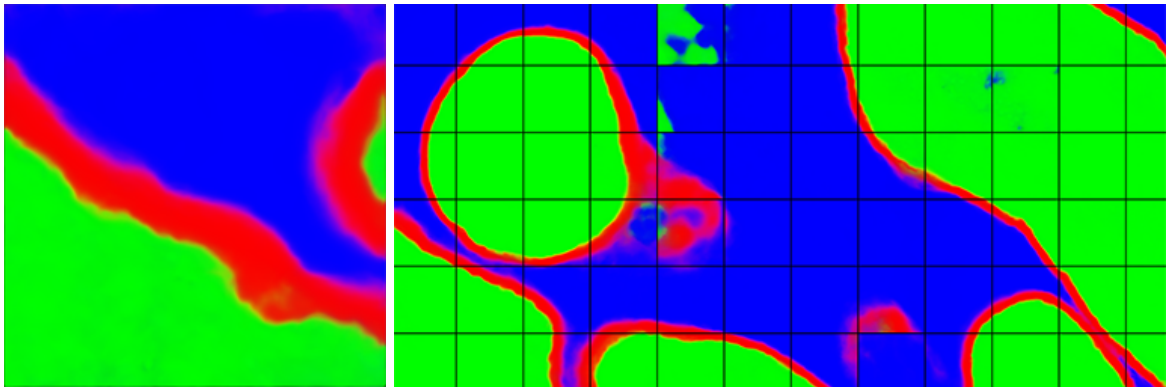


Figure 7: Single inference output image (left). Raw model output joined back together (right). Black lines added for reference. Even with overlap the predictions are not perfect and the tiling effect is visible in some regions. Tubule bodies are drawn in green colour and their borders in red.

2.4 Segmentation model

Pre-processed images are consumed by a multiclass-segmentation model (tubule body, tubule wall and background). We chose the U-Net architecture for performing image segmentation.

Alternatively, one could use different models such as U-Net++ (Zhou *et al.*, 2018), Fully Convolutional Networks (FCN) (Roth *et al.*, 2018), Feature Pyramid Network (FPN) (Lin *et al.*, 2016) or LinkNet (Chaurasia and Culurciello, 2017) . We are using the Segmentation Models library (Iakubovskii, 2019) written in Python programming language which makes U-Net's usage trivially simple. There may be benefits to using one of the mentioned architectures over U-Net, though the author thinks the difference will be very minor, if at all. Still, testing that assumption is a good candidate for follow-up work. However, in our opinion it is the preprocessing, data volume and augmentation, not model selection, that leads to better results.

While encoder sub-network (backbone) could consist of a simple collection of convolutional layers, more advanced backbones (such as ResNet without the last layer) are often used instead. In our model we chose Resnet-34 as the encoding backbone because it is relatively lightweight while still being powerful enough. Deeper variants (such as ResNet-101) are not expected to improve results while being much slower to train and require more memory (Wang, Li and Xu, 2022). In the author's opinion even ResNet-18 might have been sufficient (this assumption was not tested because this occurred to us after training the models).

ResNet is a convolutional neural network architecture that has been shown to work well on many image recognition tasks (He *et al.*, 2015), (Minaee *et al.*, 2020). Their main idea is adding skip-connections between non-adjacent layers, skipping 2 or 3 layers at a time. Without skip-connections the accuracy of deeper networks rapidly drops (even in training). The shortcut-connections counteract this, which allows for training deeper neural networks than before. It was introduced in 2015, winning the Imagenet classification competition..

Therefore, the chosen U-Net model had two separate groups of skip-connections: within the encoder (inputs are summed) and also between encoder and decoder (inputs are concatenated).

Segmentation model's performance is measured by intersection-over-union (IoU, also known as the Jaccard Index). It is a metric which compares the predicted and ground-truth areas in images, dividing their overlap size by their union size. It results in a score between 0 and 1. In case of a perfect match (both areas align flawlessly), the intersection and union are exactly the same and the score equals 1. In all other cases the overlap is smaller than the union, leading to a lower score (Figure 8). The method does not differentiate between false positives (model predicts a tubule where there is none) and false negatives. It penalises both equally.

In binary classification the calculation is straightforward. In multi-class segmentation the IoU may be calculated separately for every class and results are averaged. There are multiple averaging strategies available (*Sklearn.Metrics.Jaccard_score*, no date).

- Calculate IoU globally by counting the total true positives, false negatives and false positives.
- Calculate Iou separately for each class and use either weighted (accounting for class imbalances) or unweighted (discarding class imbalances) averaging

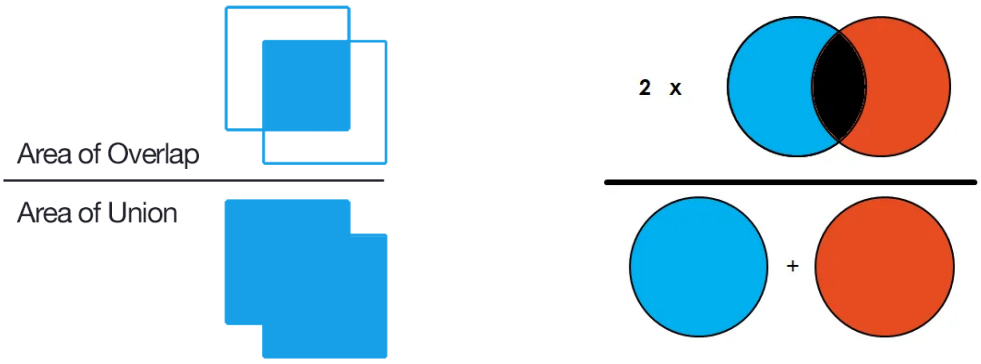


Figure 8: Quality metrics of segmentation maps. IoU (left) and F1 score (right) calculations (Tiu, 2019) both compare the areas of predictions and ground truths.

The model's performance could also be measured by F1 score (Dice Coefficient). It is very similar to IoU (Eelbode *et al.*, 2020) but calculated slightly differently. While IoU divides overlap by the union, F1 divides overlap (multiplied by two) by the sum of the areas. Both metrics work well and behave similarly. Either one is preferable over simple pixel accuracy because they work well with class imbalance. For example, pixel accuracy would report an artificially good score for a model that classifies everything as tubule body or background since tubule borders have a much lower area.

U-Net with resnet-34 backbone was chosen for segmentation. Categorical cross-entropy loss with Adam optimizer were used for training. Training progress was measured with IoU (intersection over union) and F1 scores.

In software, the Tensorflow-backed U-net from Segmentation-Models library was used, which made setting up the model relatively easy:

```
model = Unet(
    'resnet34', encoder_weights=None, input_shape=(512, 512, num_channels),
     classes=3
)
model.compile('Adam', loss=categorical_crossentropy, metrics=[iou_score,
f1_score])
```

Pixel values in both source images and ground truth segmentation maps were normalised to 0…1 range. Since there are exactly 3 classes, it was convenient to visualise the segmentation maps as

RGB images, where red channel is used for tubule borders, green for tubule bodies and blue for background. This made debugging and correctness verification much easier. Image loader converts input images and segmentation maps to tensors with shape (512,512,3) which are loaded into the model during training.

3.4 Post-processing

The raw model outputs were pieced back together, resulting in a single large image (6979 by 6817 pixels for test slide Alpha). However, the raw result was unconfident and needed post-processing (figure 9). Since the model is seeing small patches of a single huge image, a single patch sometimes contains elements from all three classes but sometimes only one class (tubule body or background). There are also instances of fully hyalinized tubules (just the border, tubule body is missing) which the model has not learnt very well. Moreover, differentiating between tubule body and border is often subtle and difficult even for a human annotator with full context (which the model does not have). This led to the presence of undesirable artefacts on the output which need to be removed. We are leveraging Otsu thresholding and image labelling, as well as morphological dilation, -erosion, -opening and -closing.

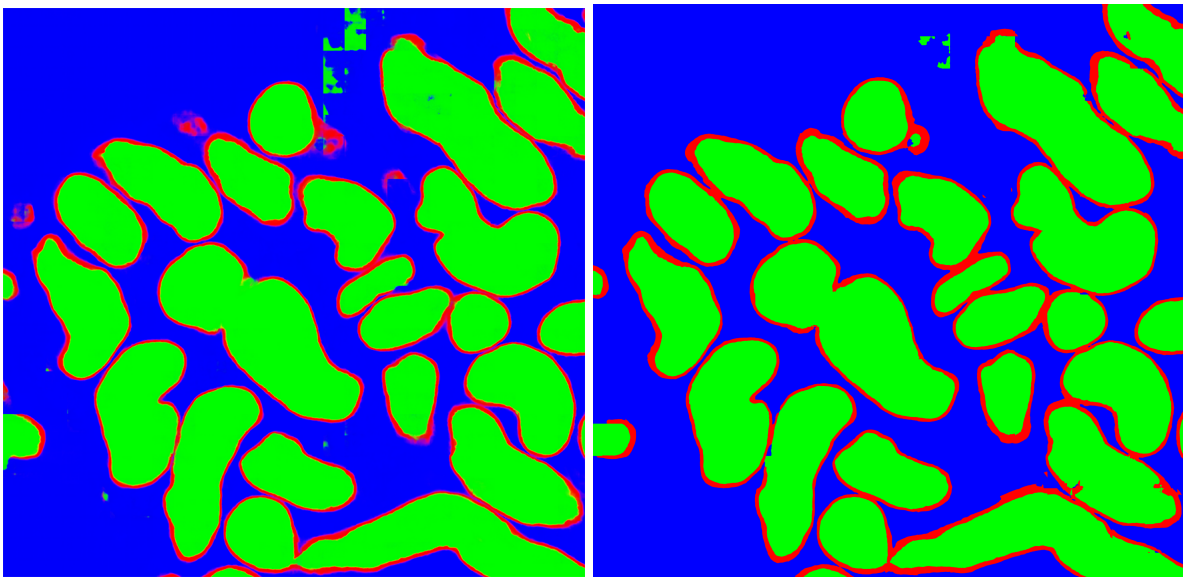


Figure 9: Raw model output of slide alpha (left) versus final post-processing result after thresholding, morphological opening & closing, labelling, removal of small tubules and borders (right)

3.4.1 Thresholding

First, we are making the raw output more confident. The three channels in the model's emitted segmentation map are not mutually exclusive and more than one channel may contain non-zero values, particularly around regions where changes occur (figure 10).

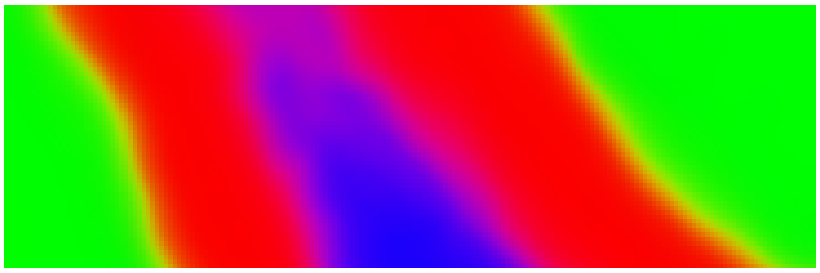


Figure 10: Segmentation map with multiple channels having non-zero values at the same time

We devised two strategies for handling this. One option is treating each class independently and accepting the model's low confidence in some areas. We used Otsu thresholding to split every channel into a binary map. Alternatively, we also used argmax to derive a clean and confident image, where for every pixel, the channel with the highest value was set to 255 and others to 0. Both approaches result in a binary map in each channel, but the former can have more than one class being activated for the same pixel.

Otsu thresholding (Otsu, 1979) is a simple technique used to separate the pixels of a grayscale (one-channel) image into foreground and background classes based on their intensity. The method determines a single threshold value which separates pixels into two classes in such a way that variances (of intensity) are minimised within the same class and maximised between the classes. Using this method, thresholding can be done automatically, without needing to pre-determine a suitable intensity value.

3.4.2 Morphological opening and closing

After thresholding we processed images using combinations of morphological dilation and erosion techniques (opening and closing). This allowed us to "fill in" the holes of tubule bodies and borders.

Morphological dilation is an image processing technique which sets every pixel to the maximum value in its neighbourhood (determined by kernel size). In binary images, if at least one pixel in the neighbourhood is 1 (white), the pixel is set to 1 (irregardless of its initial value). Morphological erosion is the opposite, setting every pixel to a minimum value in its neighbourhood (Figure 11).

By combining the two methods it is possible to perform morphological opening and -closing. Opening of an image is defined as an erosion followed by a dilation. It can remove small bright spots and connect small dark cracks. Closing is the opposite: dilation followed by erosion, allowing to remove dark spots and connect bright areas.

3.4.3 Labelling

We further processed the image by labelling and removing labels with small areas. Labelling assigns every distinct contiguous region of an image to a separate class (Figure 12). This allows for detecting tubules from the segmentation map and also removal of small false-positive ones which are likely to be noise. We used the tubule body class (without tubule border) for this task

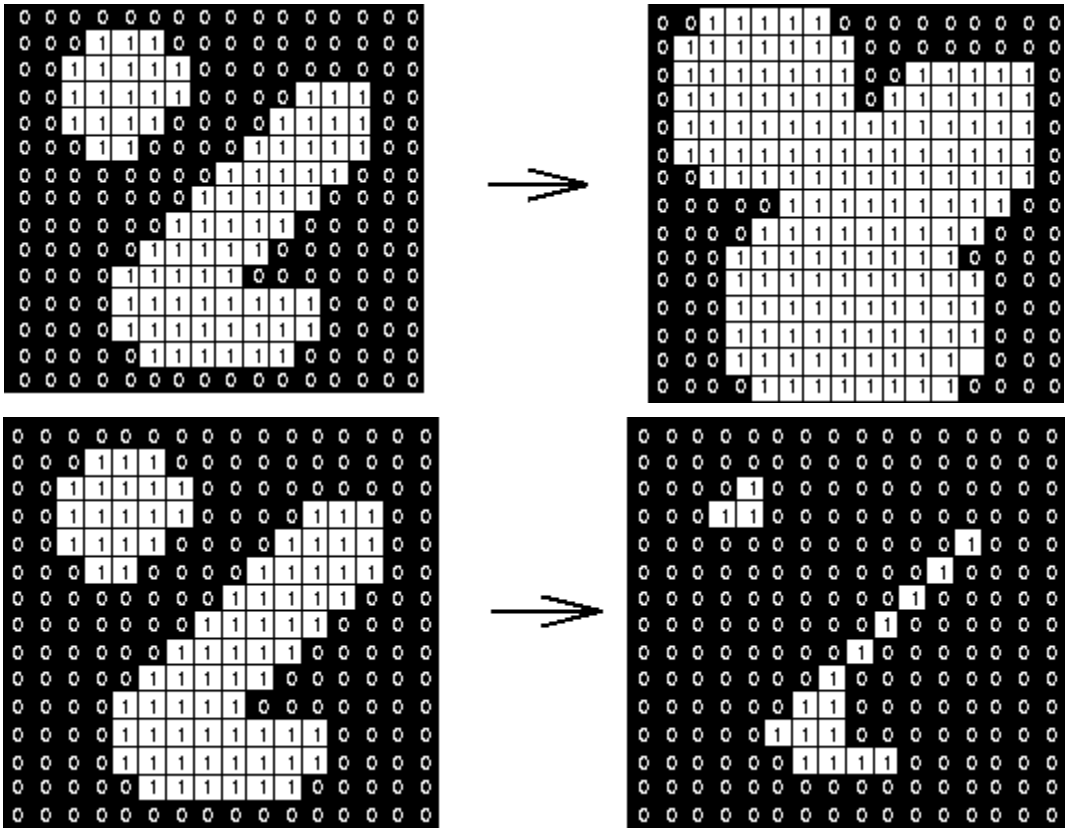


Figure 11: morphological dilation (above) and erosion (below) (Robert Fisher, Simon Perkins, Ashley Walker and Erik Wolfart)

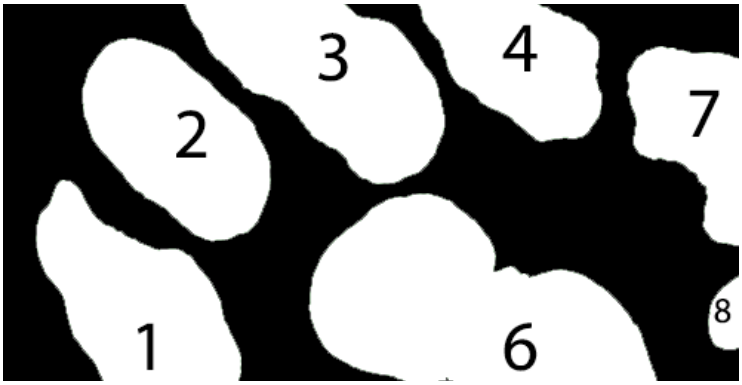


Figure 12: Labelling of contiguous areas

3.5 Measuring wall thickness

Measuring the tubule wall's thickness was relatively straightforward on the clean segmentation maps. First we tried using the "distance-maps" library, which worked but extremely slowly, taking over 10 seconds for a small 400x400 px image. It became clear that it will not scale up to images which are tens of thousands pixels wide and tall.

Thus we developed our own algorithm. As a first step, we detected border edges by applying 1px morphological dilation to the background and tubule bodies. We then subtracted the original background and tubule body areas from the result, leaving us with only the pixels altered by the dilation. The net result was two distinct lines: the edges where background and tubule bodies became tubule borders.

We then proceeded to measure the distances between these lines. For every point along the inner line, the algorithm measured a distance to the closest point on the outer one (Figure 13). We accomplished that by fitting a circle centred on the inner line that would touch or overlap the outer one. The radius of that circle is the distance at that point. The algorithm also displayed a useful characteristic: it is possible to get more precise results by increasing the circle's radius by 1 px at a time, or speed up the measurement process by increasing it in bigger steps (at the cost of accuracy). In our code, we chose to increase the radius in 5px steps, which dramatically increased the speed while still remaining relatively accurate. The algorithm spent 8 seconds on measuring an image sized 7000 by 7000 pixels, which constitutes a significant improvement over the algorithm implemented in the "distance-maps" library.

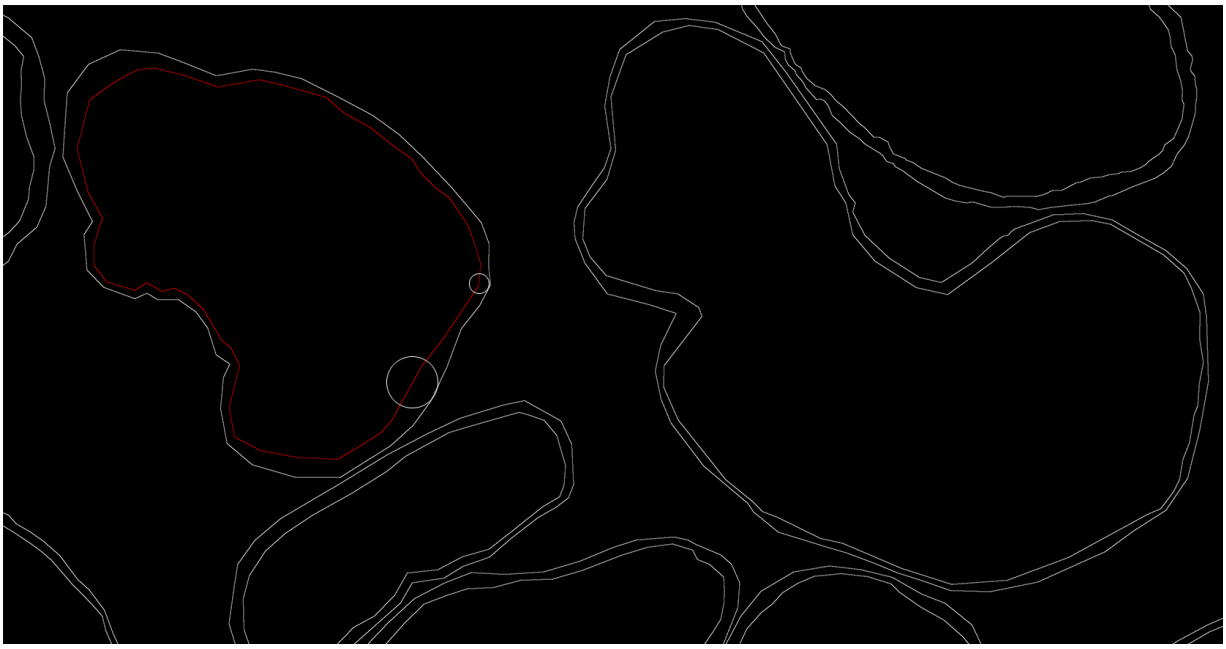


Figure 13: Calculating distances between the lines by fitting a circle centred at the inner line that touches the outer line. This was performed at every point along the inner lines.

4 Experiments and results

To find the best strategy for preprocessing images and training the model, we performed a grid search over the hyperparameter values. Each candidate setup was trained for 100 epochs which took roughly 4 hours for grayscale models and 8-12 hours for RGB ones.

Below are the hyperparameters that we have explored in attempting to improve our model performance:

- Minimum tubule area on each slide (1%, 10%, 30%)
- RGB or grayscale source images
- Slide zoom level (level 1 or level 2)

This resulted in 12 different possible combinations. In order to measure statistical significance of the results, we trained each configuration three times, leading to a total of 36 models being evaluated.

4.1 Model IoU Scores

We measured the models' performance after post-processing using an IoU score. The differences between models are far smaller than the score differences between different slides (Table 1). This indicates that the amount of data and quality of annotations are much more important than tuning hyperparameter values.

Slide	Best Model IoU	Worst Model IoU	Std Deviation
alpha	0.950	0.938	0.004
beta	0.770	0.636	0.047
gamma	0.913	0.900	0.004

Table 1: IoU scores between the best and worst performing configurations are relatively similar.

To further test that hypothesis, we assessed the statistical significance of different hyperparameters with Welch's t-test. It is a variant of independent Two Sample t-test which does not assume that both variants share the same variance. Since the score differences between slides were greater than between models, we are evaluating each hypothesis slide-by-slide.

When we compared scores by colour vs grayscale input patches, the former performed better (Table 2). These results matched our expectations because RGB images contain more information than grayscale pictures. On the other hand, surprisingly, the differences were much less dramatic than initially anticipated and not always statistically significant. We explain this by

the nature of the biomedical images: even in grayscale the images are clear enough to detect patterns. Also noteworthy is that grayscale models took roughly 3 times less time to train.

Slide	IoU Colour Images	IoU Grayscale Images	P-Value	Statistically Significant?
alpha	0.947	0.944	0.019	TRUE
beta	0.708	0.692	0.175	FALSE
gamma	0.908	0.906	0.034	TRUE

Table 2: IoU scores of colour input image models are better than that of grayscale images, but the differences are barely statistically significant.

We are unsure whether level 1 images (more zoomed in, elements on pictures are 4x larger) are better suited than level 2 images. It is a tradeoff between the number of details (level 1 are more detailed) versus context (level 2 image, being zoomed out, can contain 4x more elements). The results were inconclusive on this hyperparameter as well. There were many level-2 models that performed better than some level-1 models (and vice versa, Table 3). Combining several models trained on different zoom levels into an ensemble could likely lead to improvements.

Slide	IoU Zoom Level 1	IoU Zoom Level 2	P-Value	Statistically Significant?
alpha	0.944	0.947	0.115	FALSE
beta	0.742	0.658	0	TRUE
gamma	0.91	0.904	0	TRUE

Table 3: IoU scores of models trained on level 1 images performed better on slides beta and gamma but not on slide alpha.

Minimum required tubule area present on the training patches had little effect. We performed 3 T-tests to measure the effect of tubule area: 1% vs 10%, 10% vs 30% and 1% vs 30%. Only 10% vs 30% comparison was statistically significant. It only made any difference on the beta slide. The result was surprising because initially (when the training slides available were not fully annotated) the minimum tubule area had a large impact. However, once the training data quality was improved, this compensation technique became less important.

Slide	IoU Tubule Area 1%	IoU Tubule Area 10%	IoU Tubule Area 30%
alpha	0.946	0.947	0.944
beta	0.696	0.709	0.695
gamma	0.907	0.908	0.906

Table 4: IoU scores of different tubule areas. Different configurations only had an effect on the results of slide beta..

In conclusion, while the model ranking and the T-tests showed that models trained with zoom level 1, colour images, and minimum tubule area 10% performed best, it was not the single-best-performing model for any of the slides. However, it was in the top-3 for two of the three slides.

4.2 Wall thickness measurement results

More accurate raw and post-processed model outputs do not necessarily lead to more accurate wall thickness predictions. For example, holes in tubule bodies, or elongated tubule walls do not affect the final measurements while both of them do affect the IoU scores.

We measured the tubule walls' thicknesses on the post-processed images and ground truth segmentation maps, using accuracy of 5 pixels (1.2 micrometres for level 1 and 2.42 micrometres for level 2). The outputs include wall thickness values percentiles-wise (5, 25, 50, 75, 95, 99 and 99.9th percentiles). This allowed us to assess the distribution of the thicknesses. For example, if a relatively small number of tubules have extremely thick walls while there are also healthy tubules, the situation is different when compared to a slide where most of the tubules have relatively thick walls while not containing any with extremely thick ones.

While the differences in IoU scores were small, they made a large difference in wall thickness measurement accuracies (Table 5). We discovered two root causes for this where post-processing left artefacts which caused wall-thickness measurements to report wrong results, even though the model IoU score was relatively high.

Slide	RMSE Best Model	Mean RMSE	RMSE Worst Model	Std Deviation
alpha	3.273	26.290	58.002	13.560
beta	9.063	32.643	72.973	14.183
gamma	10.69	28.878	76.788	15.530

Table 5: RMSE values of best-performing and worst-performing configurations

One cause for high errors was borders within tubule bodies (Figure 14). This caused our algorithm to find the closest distance to the background which was very far away. To remedy this, we need to either improve post-processing to remove such phantom-borders or introduce additional heuristics which would require that the path from tubule-border-boundary to background-border boundary not contain any tubule pixels.

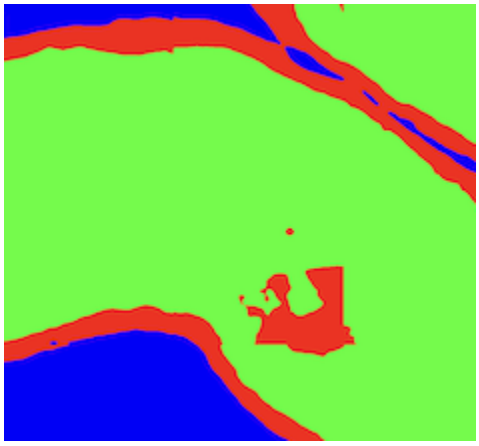


Figure 14: Errorrenous borders within the tubules confuse the border measuring algorithm.

Another cause was low quality in segmentation model output exacerbated by Otsu thresholding. There were regions of dark red (unconfident border) and dark green (unconfident tubule body) in the segmentation map (Figure 15). In case of argmax thresholding this would not have been an issue, but Otsu thresholding processed each channel independently and set the threshold values such that there occurred regions where neither of the channels was present. This caused the wall thickness algorithm to report artificially low border thicknesses. Unfortunately we had chosen Otsu thresholding because the initial IoU scores were higher when using that method. This is easily remedied by switching to an argmax segmentation.

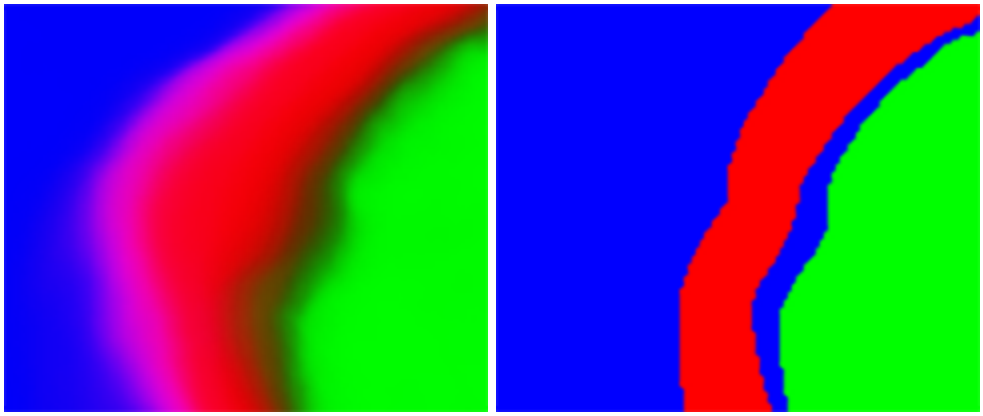


Figure 15: Unconfident raw model output combined with Otsu thresholding led to gaps between tubule bodies and borders.

The problems above serve as a good example where model IoU scores can be misleading. Despite the issues above, we present the impacts of the different hyperparameter values on the final wall thickness measuring accuracies. However, these results can change once the underlying post-processing issues are resolved.

Slide	RMSE colour	RMSE grayscale	P-Value	Significant
alpha	23.057	29.523	0.043	True
beta	32.775	32.511	0.938	False
gamma	30.230	27.526	0.464	False

Table 6: Colour images performed better on the alpha slide while having little effect on other ones.

Slide	RMSE level 1	RMSE level 2	P-Value	Significant
alpha	35.201	17.379	0.000	True
beta	26.366	38.920	0.000	True
gamma	32.233	25.523	0.068	True

Table 7: Strong statistical significance of zoom level. Unfortunately the best configuration differs slide by slide.

Slide	RMSE overlap 25%	RMSE overlap 50%	P-Value	Significant
alpha	27.191	25.389	0.576	False
beta	32.634	32.652	0.996	False
gamma	28.675	29.081	0.913	False

Table 8: Slide patch overlap during inference did not make a difference. Using smaller overlap works faster however.

Luckily, not every model suffered from these problems. This allowed us to present the output of the following configuration: zoom level 1, minimum tubule overlap 30%, colour images, inference slide overlap 25% (Figure 16). We chose to showcase it because it performed reasonably well on all slides.

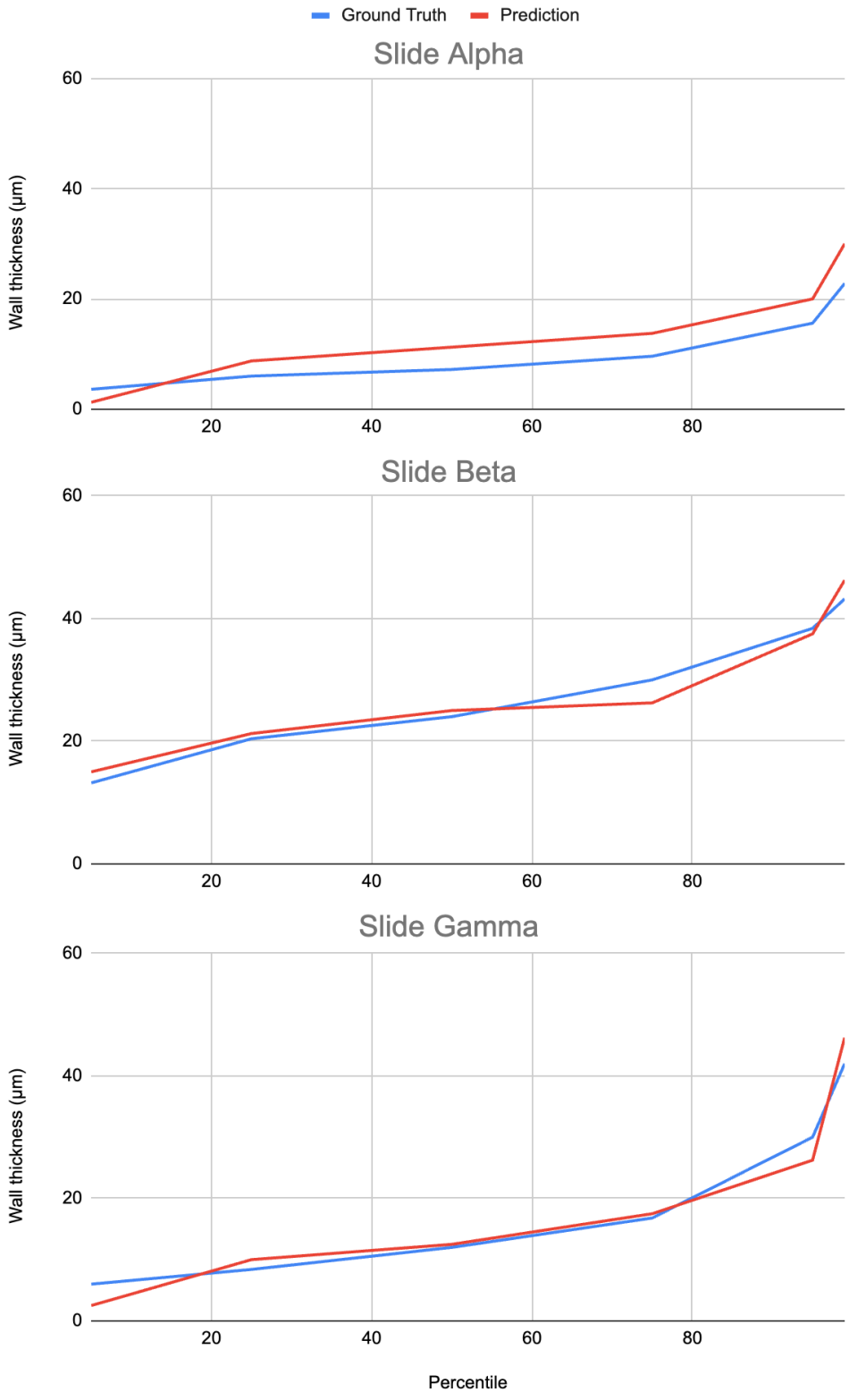


Figure 16: Wall thickness ground truth vs predictions by percentiles. The model predicts average thicknesses relatively accurately but performs slightly worse on both extremes (very thick and very thin sections).

5 Discussion & Conclusions

The initial results are promising. The segmentation model is successful in separating the image pixels into tubules and their borders. More importantly, out of the three slides we received one had enlarged tubule walls' thicknesses (slide beta), which is exactly what the software determined (figure 17).

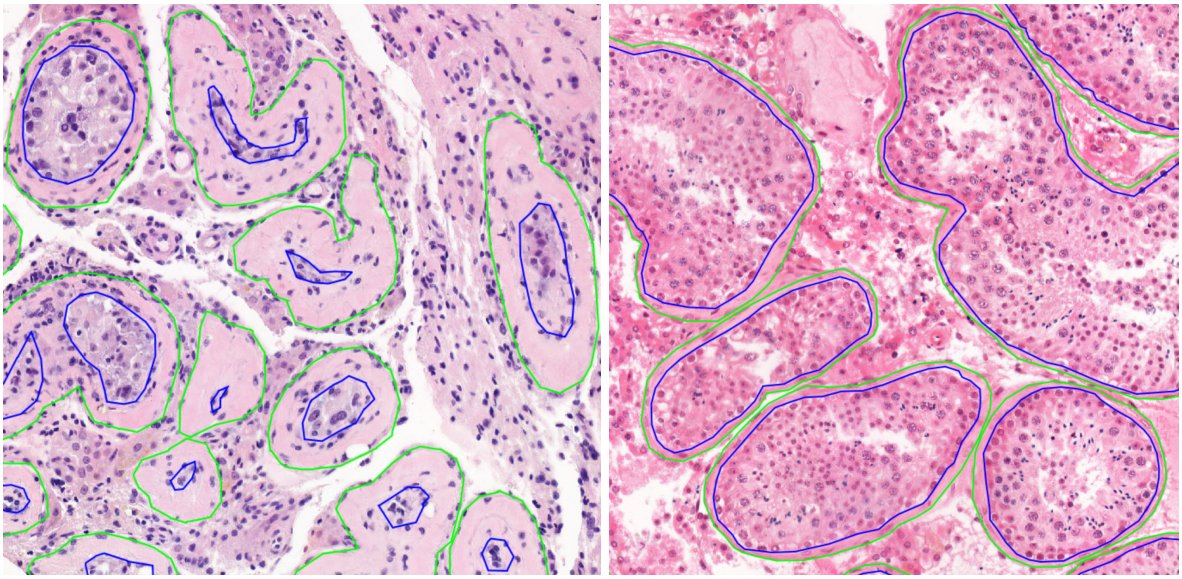


Figure 17: Slide beta (left) contains tubules with enlarged wall thicknesses in comparison to slide alpha (right)

To increase the usefulness even further, we wish to report measurements tubule by tubule, which would allow easier determination of the ratio of healthy and pathological tissue.

We could also have done more thorough exploration of segmentation models. Given more time, we would like to investigate if using a simpler (such as ResNet-18 backed U-Net) or a different (U-Net++, FPN) model would lead to a different accuracy. Our (unverified) assumption is that it does not and using a much simpler segmentation model would lead to the same accuracy with much faster training time.

We believed that the most important differentiating effort lies in pre-processing the images and post-processing the model outputs. While we still believe in the importance of pre-processing, the exact hyperparameter values do not make a huge difference and should be optimised for computation speed. Post-processing, however, may not be important at all. Even though the IoU scores improved dramatically between the raw model and post-processed output, it may not necessarily lead to more accurate wall thickness measurements. On the contrary - we noticed that the post-processing is sometimes too aggressive and could even have detrimental effects. Testing

the pipeline's performance with reduced post-processing is one of the top priorities for the work going forward.

Comparisons between colour and grayscale models may not have been just. Colour models took 3 times more time to train than grayscale ones. The grid search should be repeated with equal training time. That would lead to the grayscale models being trained for roughly 3 times more epochs and could make the measured performance differences disappear (or even reverse them).

There was no overwhelmingly best set of hyperparameters. On the contrary: multiple very different models appeared in the top ranking and the best performing models differed significantly between the slides. This indicates that ensembling the models could lead to noticeable improvements in performance and should be investigated. One of the highest priorities should be fixing post-processing issues. Luckily the root causes are understood and relatively easy to remedy.

For real world adoption and trustworthiness the model should be trained with more than 3 histopathology slides. We were surprised how well the segmentation model performed with the limited training data being available to us, but 3 specimens is clearly insufficient for an application in biomedical and clinical domains, where sample sizes are usually in the thousands.

One of the main contributions of this work is the software pipeline. While the thesis is solely focused on measuring tubule walls' thickness in testis tissue, the pre-processing codebase can be reused for training the model on other tissue types and/or for detecting other abnormalities. The software's architecture can be made dynamic, in order to facilitate different models and post-processing modules while retaining the pre-processing functionality.

The software is also not yet ready for adoption. It currently consists of multiple scripts which are run separately by hand. Ideal solution would be a deployable web application which is interfaced with medical facility's imaging equipment and include a graphical user interface for further results analysis. For example, the application could display the unhealthy tubules it has identified along with the statistical measurements. It is the author's hope that the work towards this goal could continue and the software be deployed. It could prove to be a valuable tool in widespread screening of common ailments.

References

Bankhead, P. *et al.* (2017) ‘QuPath: Open source software for digital pathology image analysis’, *Scientific reports*, 7(1), p. 16878.

Chaurasia, A. and Culurciello, E. (2017) ‘LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1707.03718.

Eelbode, T. *et al.* (2020) ‘Optimization for Medical Image Segmentation: Theory and Practice When Evaluating With Dice Score or Jaccard Index’, *IEEE transactions on medical imaging*, 39(11), pp. 3679–3690.

Fakhrzadeh, A. *et al.* (2023) ‘Deep learning-based method for segmenting epithelial layer of tubules in histopathological images of testicular tissue’, *arXiv [eess.IV]*. Available at: http://arxiv.org/abs/2301.09887.

Godart, E.S. and Turek, P.J. (2020) ‘The evolution of testicular sperm extraction and preservation techniques’, *Faculty reviews*, 9, p. 2.

He, K. *et al.* (2015) ‘Deep Residual Learning for Image Recognition’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1512.03385.

Iakubovskii, P. (2019) ‘Segmentation Models’, *GitHub repository* [Preprint]. GitHub. Available at: https://github.com/qubvel/segmentation_models.

Kumar, N. and Singh, A.K. (2015) ‘Trends of male factor infertility, an important cause of infertility: A review of literature’, *Journal of human reproductive sciences*, 8(4), pp. 191–196.

Lin, T.-Y. *et al.* (2016) ‘Feature Pyramid Networks for Object Detection’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1612.03144.

Minaee, S. *et al.* (2020) ‘Image Segmentation Using Deep Learning: A Survey’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/2001.05566.

‘MIRAX Virtual Slide File’ (2020). Available at: https://fileinfo.com/extension/mrxs (Accessed: 6 May 2023).

OpenSlide (2015). Available at: https://openslide.org/ (Accessed: 18 April 2023).

Otsu, N. (1979) ‘A threshold selection method from gray-level histograms’, *IEEE transactions on systems, man, and cybernetics*, 9(1), pp. 62–66.

Robert Fisher, Simon Perkins, Ashley Walker and Erik Wolfart (no date) *Morphology*, *Hypermedia Image Processing Reference*. Available at: https://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm.

Ronneberger, O., Fischer, P. and Brox, T. (2015) ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1505.04597.

Roth, H.R. *et al.* (2018) ‘An application of cascaded 3D fully convolutional networks for medical image segmentation’, *Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society*, 66, pp. 90–99.

Saood, A. and Hatem, I. (2021) ‘COVID-19 lung CT image segmentation using deep learning methods: U-Net versus SegNet’, *BMC medical imaging*, 21(1), p. 19.

Siddique, N. *et al.* (2021) ‘U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications’, *IEEE Access*, 9, pp. 82031–82057.

Sklearn.Metrics.Jaccard_score (no date) *scikit-learn*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_score.html (Accessed: 2 May 2023).

Sziva, R.E. *et al.* (2022) ‘Accurate Quantitative Histomorphometric-Mathematical Image Analysis Methodology of Rodent Testicular Tissue and Its Possible Future Research Perspectives in Andrology and Reproductive Medicine’, *Life* , 12(2). Available at: https://doi.org/10.3390/life12020189.

Tiu, E. (2019) *Metrics to Evaluate your Semantic Segmentation Model*, *Towards Data Science*. Available at: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2 (Accessed: 18 April 2023).

Wang, H., Li, K. and Xu, C. (2022) ‘A New Generation of ResNet Model Based on Artificial Intelligence and Few Data Driven and Its Construction in Image Recognition Model’, *Computational intelligence and neuroscience*, 2022, p. 5976155.

Zhou, Z. *et al.* (2018) ‘UNet++: A Nested U-Net Architecture for Medical Image Segmentation’, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, S...*, 11045, pp. 3–11.

Appendix

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dmytro Fishman for offering me an impactful as well as interesting thesis topic, also for his guidance, extremely helpful suggestions and support.

I am grateful to our clinical collaborators Georgi Dzaparidze and Erik Tamp from the East Tallinn Central Hospital who provided necessary datasets and had ample patience to describe the process of histopathology and the ailments troubling male testicles.

Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, Arnel Pällo,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,
   Measuring Testis Tubule Wall Thickness in Histopathology Images,
   supervised by Dmytro Fishman.

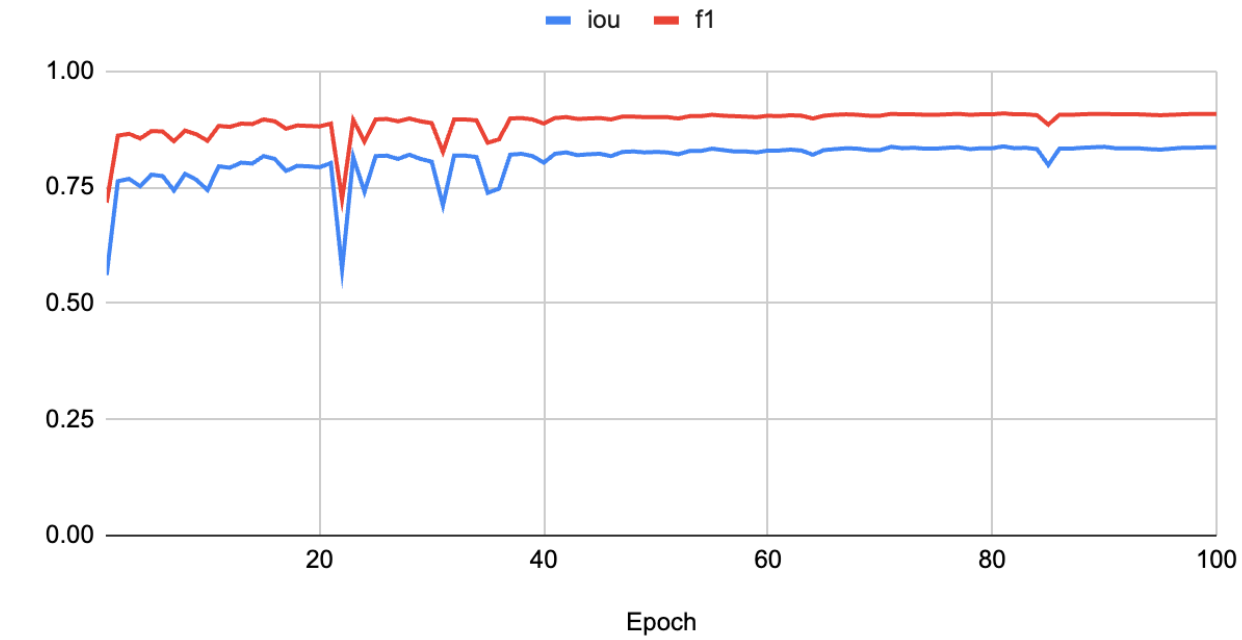
2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author,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

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

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

Arnel Pällo

09/05/2023



Appendix Figure 1: IoU and F1 scores from one of our model training logs. The scores correlate perfectly (spearman correlation coefficient 0.998). Even though the latter is always greater, they report the same information and either one could be used.

Model IoU Scores

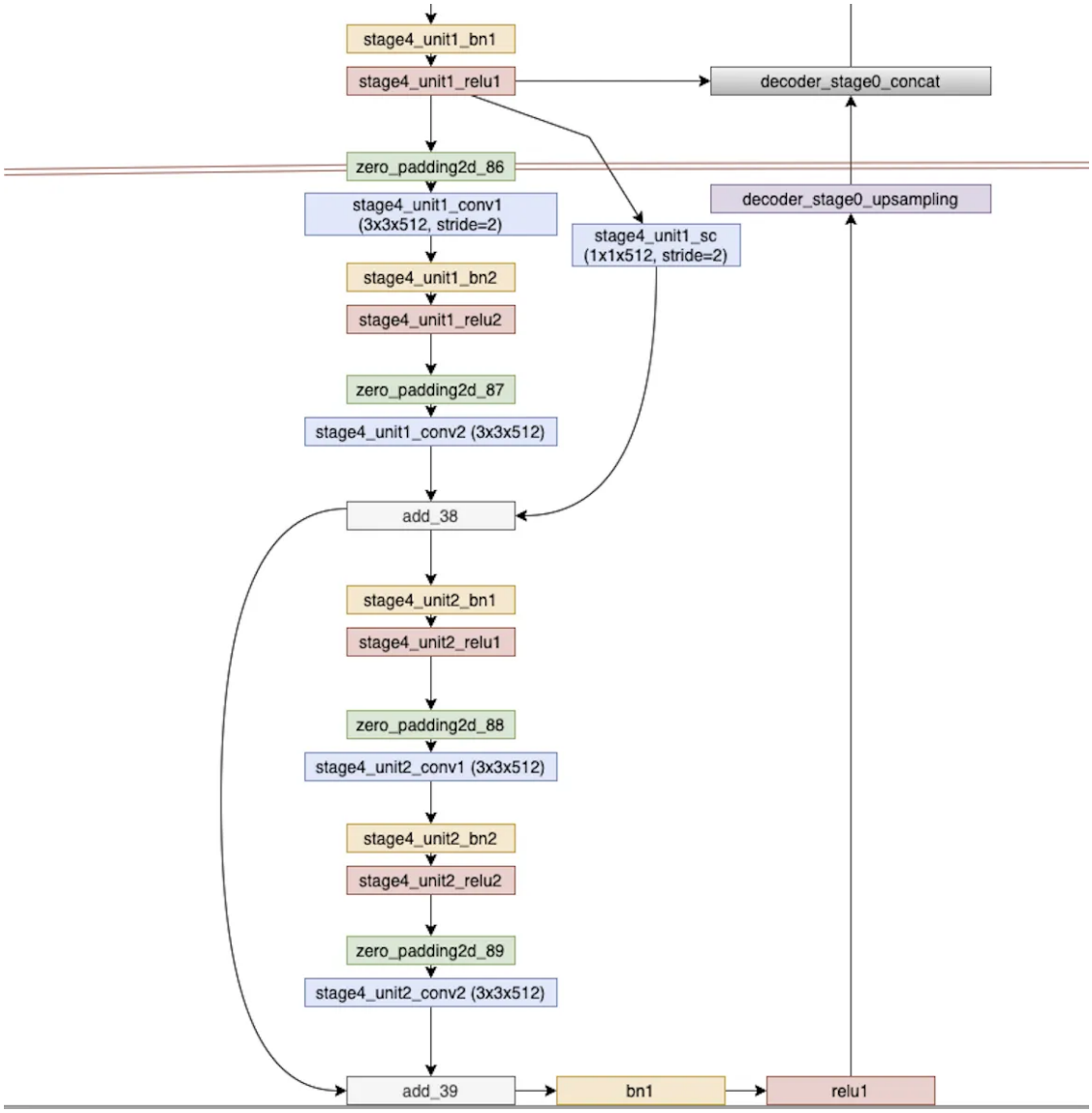
level	tubule area	colour mode	generation	model iou alpha 25	model iou alpha 50	model iou beta 25	model iou beta 50	model iou gamma 25	model iou gamma 50
1	1	colour	0	0.532	0.542	0.513	0.516	0.592	0.595
1	10	colour	0	0.522	0.534	0.488	0.493	0.569	0.571
1	30	colour	0	0.558	0.569	0.520	0.523	0.496	0.521
1	1	grayscale	0	0.519	0.524	0.435	0.437	0.538	0.533
1	10	grayscale	0	0.495	0.513	0.423	0.436	0.530	0.541
1	30	grayscale	0	0.543	0.547	0.514	0.519	0.490	0.484
2	1	colour	0	0.419	0.427	0.418	0.419	0.544	0.549
2	10	colour	0	0.551	0.563	0.536	0.540	0.601	0.602
2	30	colour	0	0.528	0.543	0.502	0.504	0.580	0.582
2	1	grayscale	0	0.429	0.432	0.408	0.418	0.520	0.524
2	10	grayscale	0	0.488	0.500	0.392	0.401	0.569	0.572
2	30	grayscale	0	0.466	0.490	0.442	0.444	0.543	0.552
1	1	colour	1	0.500	0.508	0.493	0.498	0.579	0.582
1	10	colour	1	0.445	0.464	0.462	0.463	0.550	0.559
1	30	colour	1	0.552	0.558	0.457	0.463	0.564	0.564
1	1	grayscale	1	0.415	0.419	0.314	0.317	0.493	0.496
1	10	grayscale	1	0.490	0.501	0.419	0.432	0.533	0.539
1	30	grayscale	1	0.514	0.531	0.472	0.476	0.551	0.561
2	1	colour	1	0.420	0.424	0.423	0.429	0.554	0.558
2	10	colour	1	0.512	0.530	0.540	0.541	0.601	0.602
2	30	colour	1	0.516	0.543	0.459	0.460	0.575	0.577
2	1	grayscale	1	0.602	0.622	0.542	0.540	0.641	0.644
2	10	grayscale	1	0.587	0.592	0.519	0.519	0.599	0.600
2	30	grayscale	1	0.485	0.502	0.494	0.494	0.557	0.564
1	1	colour	2	0.498	0.506	0.499	0.507	0.583	0.586
1	10	colour	2	0.513	0.523	0.479	0.485	0.580	0.586
1	30	colour	2	0.432	0.443	0.403	0.414	0.417	0.412
1	1	grayscale	2	0.502	0.510	0.404	0.422	0.554	0.560
1	10	grayscale	2	0.574	0.580	0.456	0.465	0.571	0.576
1	30	grayscale	2	0.539	0.547	0.411	0.425	0.543	0.549
2	1	colour	2	0.449	0.454	0.421	0.424	0.562	0.563
2	10	colour	2	0.504	0.521	0.499	0.503	0.583	0.586
2	30	colour	2	0.486	0.509	0.455	0.455	0.566	0.569
2	1	grayscale	2	0.468	0.478	0.417	0.419	0.557	0.562
2	10	grayscale	2	0.572	0.579	0.496	0.499	0.599	0.600
2	30	grayscale	2	0.551	0.562	0.453	0.458	0.567	0.571

Appendix Table 1: Raw model output IoU scores

level	tubule area	colour mode	generation	postp iou alpha 25	postp iou alpha 50	postp iou beta 25	postp iou beta 50	postp iou gamma 25	postp iou gamma 50
1	1	colour	0	0.947	0.947	0.773	0.769	0.914	0.914
1	10	colour	0	0.946	0.946	0.760	0.758	0.912	0.913
1	30	colour	0	0.946	0.949	0.751	0.749	0.912	0.914
1	1	grayscale	0	0.943	0.944	0.721	0.742	0.909	0.911
1	10	grayscale	0	0.942	0.944	0.762	0.765	0.906	0.907
1	30	grayscale	0	0.943	0.947	0.738	0.733	0.911	0.911
2	1	colour	0	0.952	0.952	0.637	0.637	0.901	0.901
2	10	colour	0	0.951	0.950	0.665	0.667	0.908	0.908
2	30	colour	0	0.945	0.947	0.685	0.685	0.903	0.904
2	1	grayscale	0	0.947	0.947	0.676	0.677	0.901	0.902
2	10	grayscale	0	0.939	0.938	0.641	0.642	0.902	0.904
2	30	grayscale	0	0.944	0.945	0.637	0.634	0.903	0.903
1	1	colour	1	0.954	0.955	0.747	0.747	0.912	0.912
1	10	colour	1	0.942	0.943	0.718	0.696	0.910	0.909
1	30	colour	1	0.948	0.945	0.739	0.743	0.910	0.910
1	1	grayscale	1	0.927	0.928	0.667	0.693	0.908	0.910
1	10	grayscale	1	0.942	0.945	0.767	0.775	0.906	0.906
1	30	grayscale	1	0.945	0.946	0.725	0.754	0.907	0.909
2	1	colour	1	0.938	0.940	0.649	0.649	0.898	0.898
2	10	colour	1	0.945	0.946	0.650	0.650	0.905	0.905
2	30	colour	1	0.949	0.949	0.657	0.658	0.904	0.904
2	1	grayscale	1	0.952	0.954	0.660	0.660	0.909	0.909
2	10	grayscale	1	0.952	0.953	0.655	0.656	0.906	0.906
2	30	grayscale	1	0.939	0.939	0.621	0.620	0.894	0.894
1	1	colour	2	0.943	0.944	0.746	0.748	0.912	0.913
1	10	colour	2	0.950	0.953	0.772	0.771	0.912	0.913
1	30	colour	2	0.948	0.944	0.734	0.732	0.910	0.911
1	1	grayscale	2	0.941	0.945	0.703	0.699	0.909	0.911
1	10	grayscale	2	0.939	0.945	0.771	0.782	0.908	0.910
1	30	grayscale	2	0.937	0.942	0.725	0.720	0.903	0.905
2	1	colour	2	0.952	0.953	0.717	0.711	0.904	0.904
2	10	colour	2	0.955	0.956	0.685	0.685	0.909	0.910
2	30	colour	2	0.930	0.935	0.668	0.667	0.905	0.905
2	1	grayscale	2	0.942	0.945	0.633	0.633	0.907	0.908
2	10	grayscale	2	0.954	0.953	0.661	0.662	0.906	0.906
2	30	grayscale	2	0.943	0.945	0.650	0.654	0.904	0.903

Appendix Table 2: Post-processed model Output IoU Scores

Zoomed in view of the U-Net model with multiple skip-connections



Appendix Figure 2: Excerpt from the Segmentation Model's U-Net. Reverse engineered from the compiled model.