

# Glomerulus Segmentation in Kidney Biopsies

## Computer Vision Project, Group 9

Lukas Göbl, Peer Schäfer, Lukas Scheib

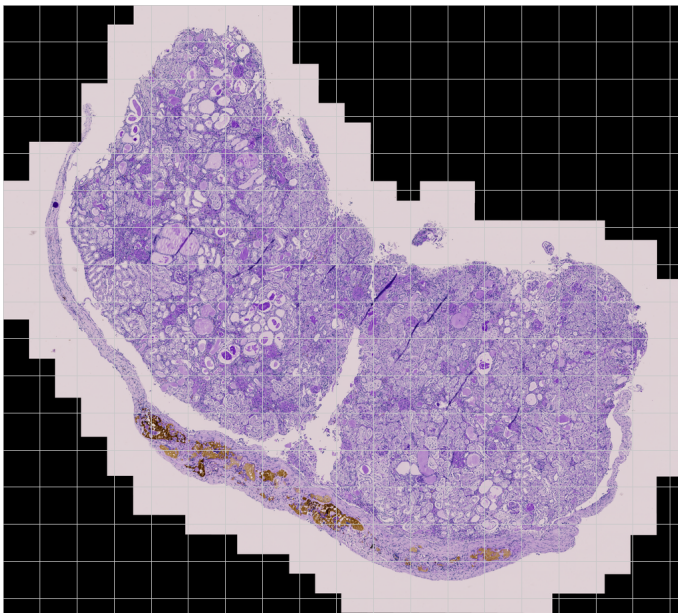
15.07.2025

# Problem Statement

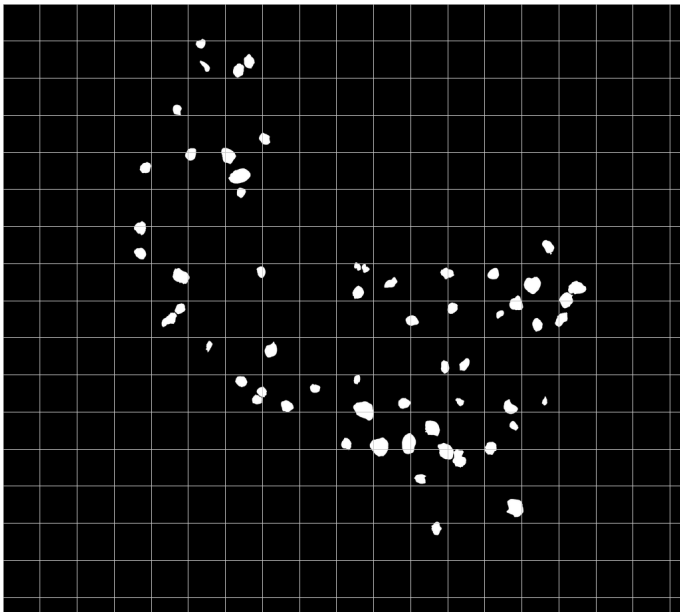
**General:** Accurate segmentation of glomeruli in kidney biopsy images is crucial for diagnosing chronic kidney diseases.

**Precise Tasks:** Segment glomeruli at patch-level and whole-slide level in kidney tissue images as part of the MICCAI 2024 KPIs Challenge.

# WSI Example: train\56Nx\12\_116



# WSI Mask Example: train\56Nx\12\_116



- Standard UNet architecture with 8 double convolution layers split into encoder and decoder layers with skip connections between them.
- Dilated convolutions in the bottleneck (with dilation rates 1, 2, 4 on  $\frac{1}{3}$  of the channels each) for multi-scale feature extraction.<sup>1</sup>
- Channel quantities scale with variable patch size.

---

<sup>1</sup>Li et al. (2021)

- Trained on official KPIs dataset (train/val/test splits).
- Combined binary cross entropy (BCE) and Dice loss to address class imbalance.
- Hyperparameter tuning: BCE/Dice ratio, Dice smooth, positive class weighting for BCE, learning rate for Adam optimizer.
- Experiments with different patch sizes ( $256^2$  and  $2048^2$  pixels) and data subsets.
- Evaluated using accuracy, precision, recall, and  $F_1$ -score.
- Qualitative assessment via visual inspection.

# Metrics of our best performing Models

Patch size	Split	Accuracy	Precision	Recall	$F_1$ -score
256	Validation	0.9781	0.7479	0.7660	0.7569
	Test	0.9757	0.7537	0.7133	0.7330
2048	Validation	0.9705	0.9299	0.3662	0.5255
	Test	0.9643	0.9159	0.2616	0.4070

## Hyperparameters:

- BCEDiceLoss Ratio = 0.8
- BCE Positive Weight = 25
- Dice Smooth =  $10^{-6}$
- Learning Rate =  $10^{-4}$

## Implemented patching pipeline with the following hyperparameters:

- Patch size
- Patch overlap (stride)
- Trivial patch threshold for positive pixels

## Possible next steps:

- Implementation of a stitching pipeline to combine patch predictions into a full WSI mask.
- Train VariableUNet using the complete patching and stitching pipeline.



## What worked:

- BCE + Dice loss crucial for recall and  $F_1$ -score.
- Hyperparameter tuning improved glom detection.
- WSIs can be patched for patch-level segmentation.

## Limitations:

- Instability in training, especially with full-size patches.
- Our configurations, trained on small patches, did not generalize well to larger patches.
- No empirical validation on WSI segmentation due to time constraints.

## In retrospect:

- Greater emphasis on pre- and post-processing (downsampling patches and upsampling predictions).
- More in-depth data exploration regarding samples of different diseases.

## Future work:

- Exploring different model architectures and methods.
- Search for more robust hyperparameters and domain adaptation.
- Continued work on WSI segmentation pipeline.

# Backup

# UNet Architecture

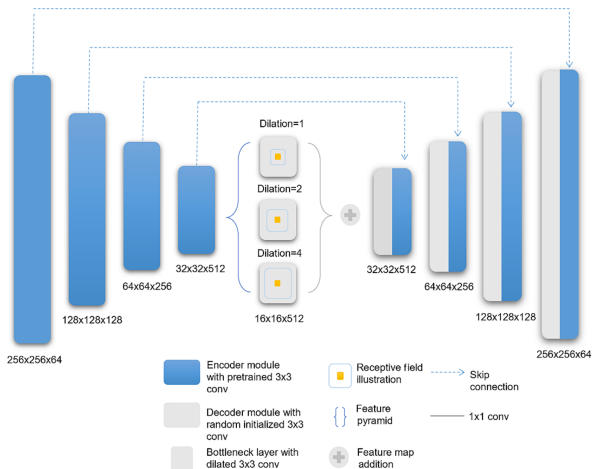
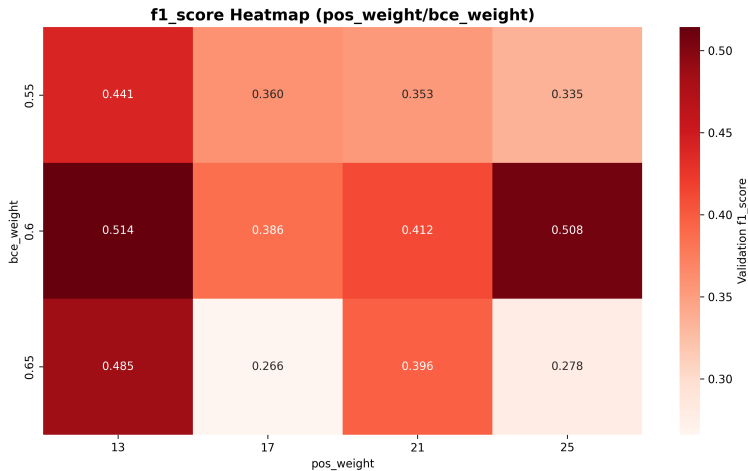


Figure: Source: Li et al. (2021), Figure 3 (edited)

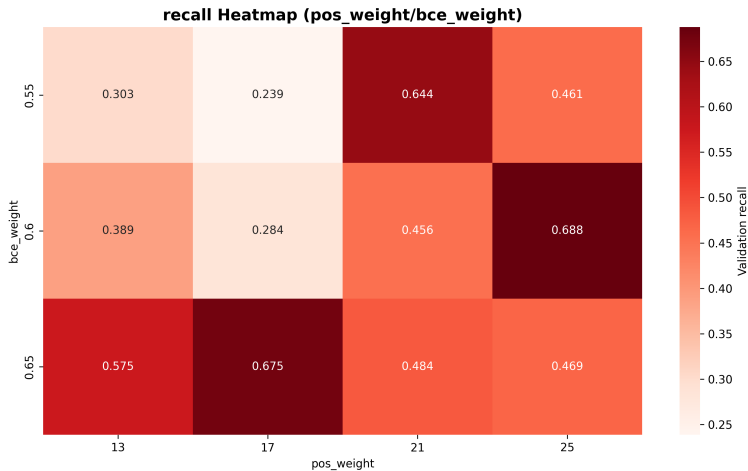
# Double Convolution in our Code Base

```
1 class DoubleConv(nn.Module):
2     def __init__(self, in_channels, out_channels):
3         super().__init__()
4         self.double_conv = nn.Sequential(
5             nn.Conv2d(in_channels, out_channels, 3,
6                       padding=1),
7             nn.BatchNorm2d(out_channels),
8             nn.ReLU(inplace=True),
9             nn.Conv2d(out_channels, out_channels, 3,
10                      padding=1),
11             nn.BatchNorm2d(out_channels),
12             nn.ReLU(inplace=True),
13         )
14
15     def forward(self, x):
16         return self.double_conv(x)
```

# $F_1$ -Score Heat Map from Experiment 2



# Recall Heat Map from Experiment 2



# FlipFlop Graph from Experiment 3

