DETECTING CANCER METASTASES ON GIGAPIXEL PATHOLOGY IMAGES

Jessie Wang (yw3765)

Zhucheng Zhan (zz2783)

AGENDA

- Introduction and Data
- Methodology
  - Data Preprocess
  - Model Experiments
  - Prediction on New Slides
- Evaluation
- Code Walkthrough
- Next Steps

DATA SOURCE

- Training Data from: https://camelyon16.grand-challenge.org/Data/
- 22 slides converted by Josh for ease of use with OpenSlide
- Whole-slide images are stored in a multi-resolution pyramid structure.
- Image files contain multiple downsampled versions of the original image.
- Example of contents included in 1 slide

```
Read WSI from tumor_091.tif with width: 61440, height: 53760
Read tumor mask from tumor_091_mask.tif
Slide includes %d levels 8
Level 0, dimensions: (61440, 53760) downsample factor 1
Level 1, dimensions: (30720, 26880) downsample factor 2
Level 2, dimensions: (15360, 13440) downsample factor 4
Level 3, dimensions: (7680, 6720) downsample factor 8
Level 4, dimensions: (3840, 3360) downsample factor 16
Level 5, dimensions: (1920, 1680) downsample factor 32
Level 6, dimensions: (960, 840) downsample factor 64
Level 7, dimensions: (480, 420) downsample factor 128
```

METHODOLOGY

Preprocess

- Pick 10 slides from 22 available
- Generate training images from desired levels
- Remove non-tissue parts
- Balance classes via downsampling and upsampling

Model Train

- Single tower
- Multi tower
- Data augmentation
- Transfer learning

Prediction and Evaluation

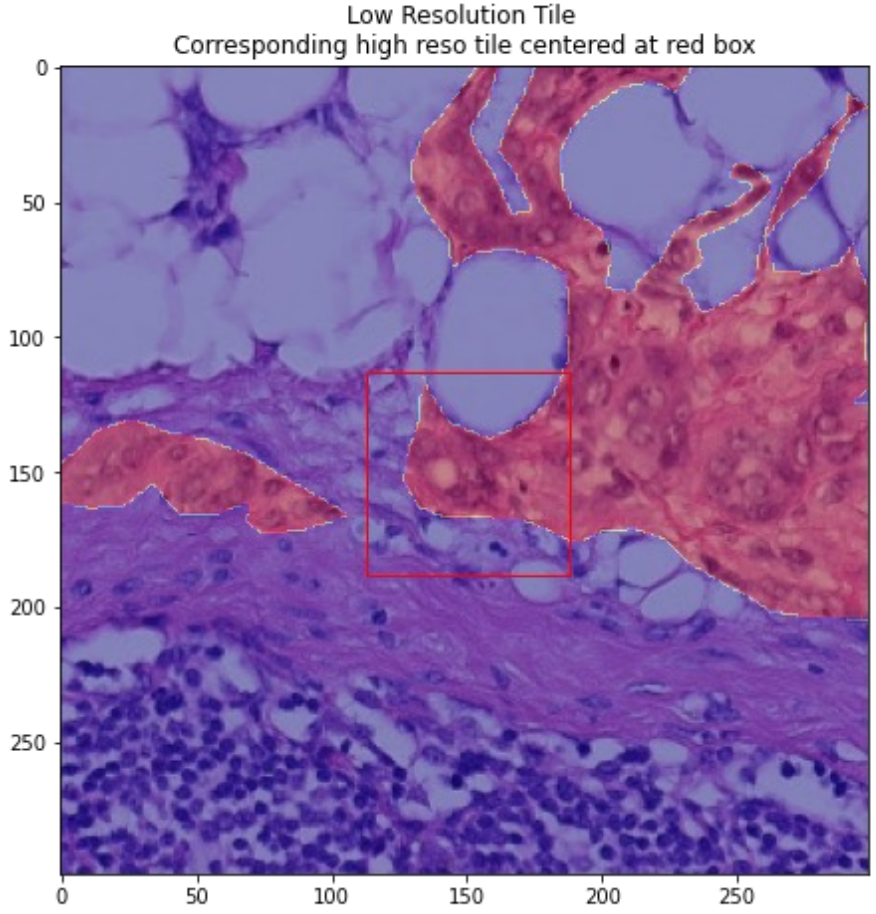
- Confusion matrix on test set
- Inference on new slides
- Generate heatmap

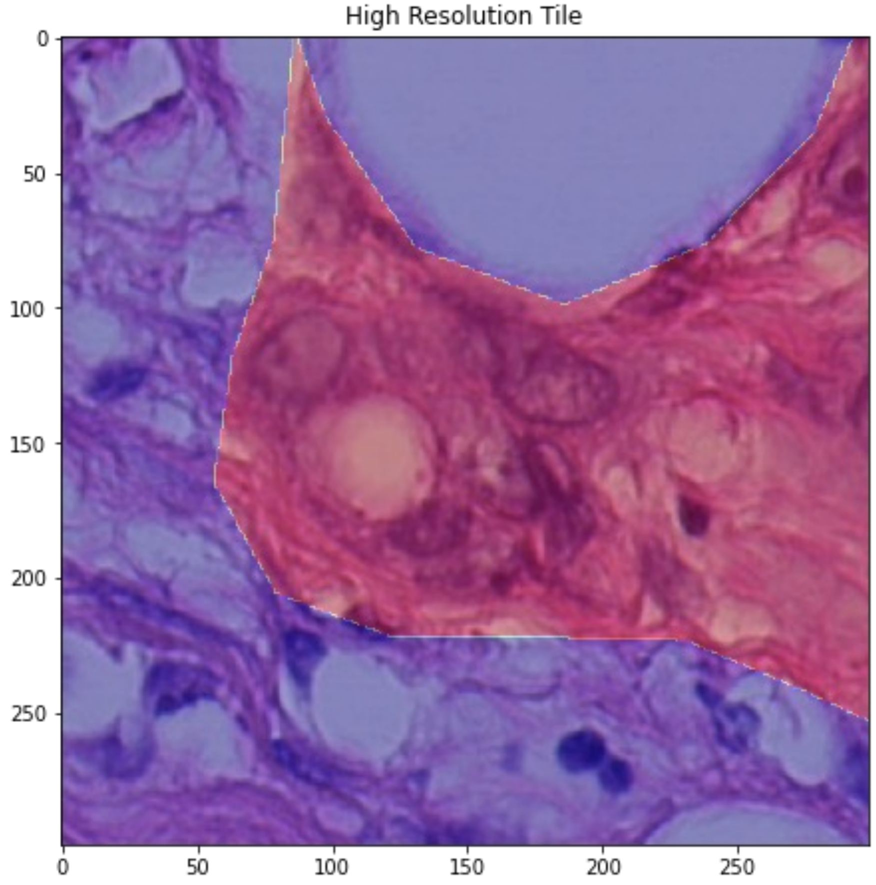
DATA GENERATION VIA IMAGE TILING

- Tiles paring from different levels →
- Non-tissue tiles filtering →
- Image pair labelling →
- Slide-wise class balancing

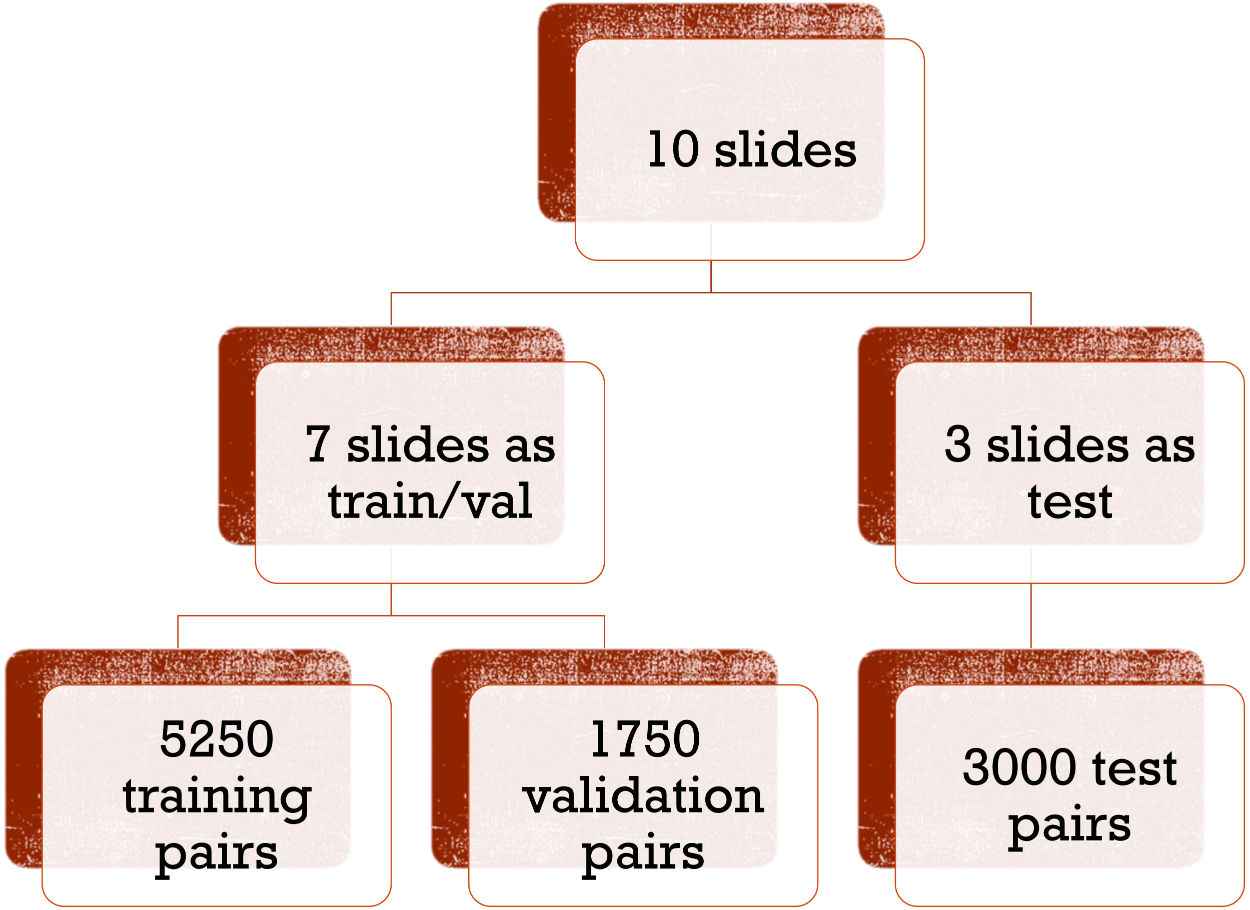
INPUT DATA

- Slide 096 (117806, 22126) at lvl 0; Tumor Status = True (indicated by red mask)



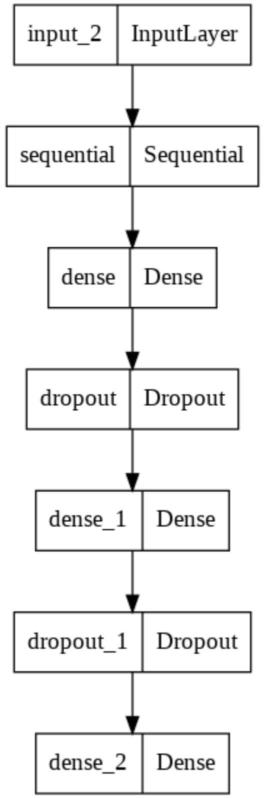


TRAIN TEST SPLIT

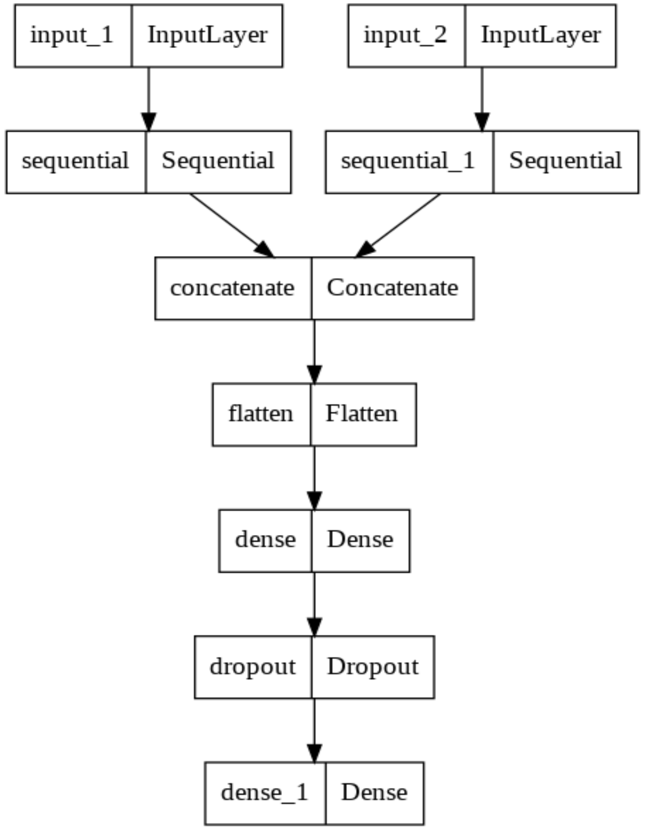


MODEL STRUCTURE

- Single tower with level 0 data
- Explored performance with transfer learning and data augmentation

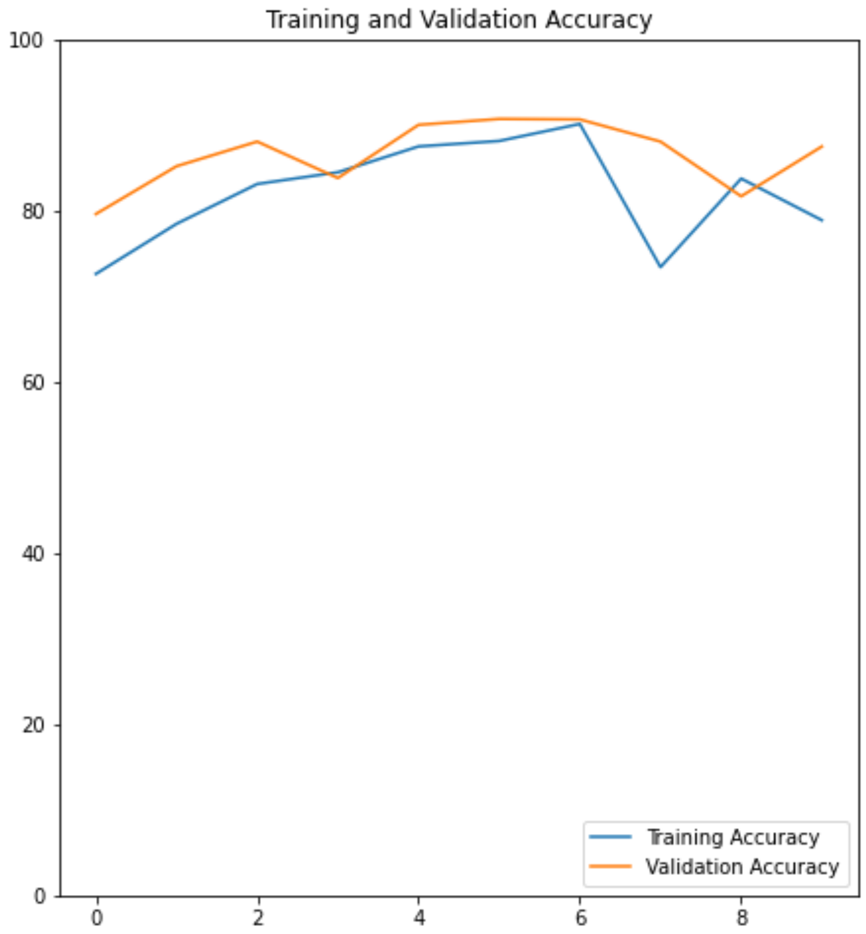


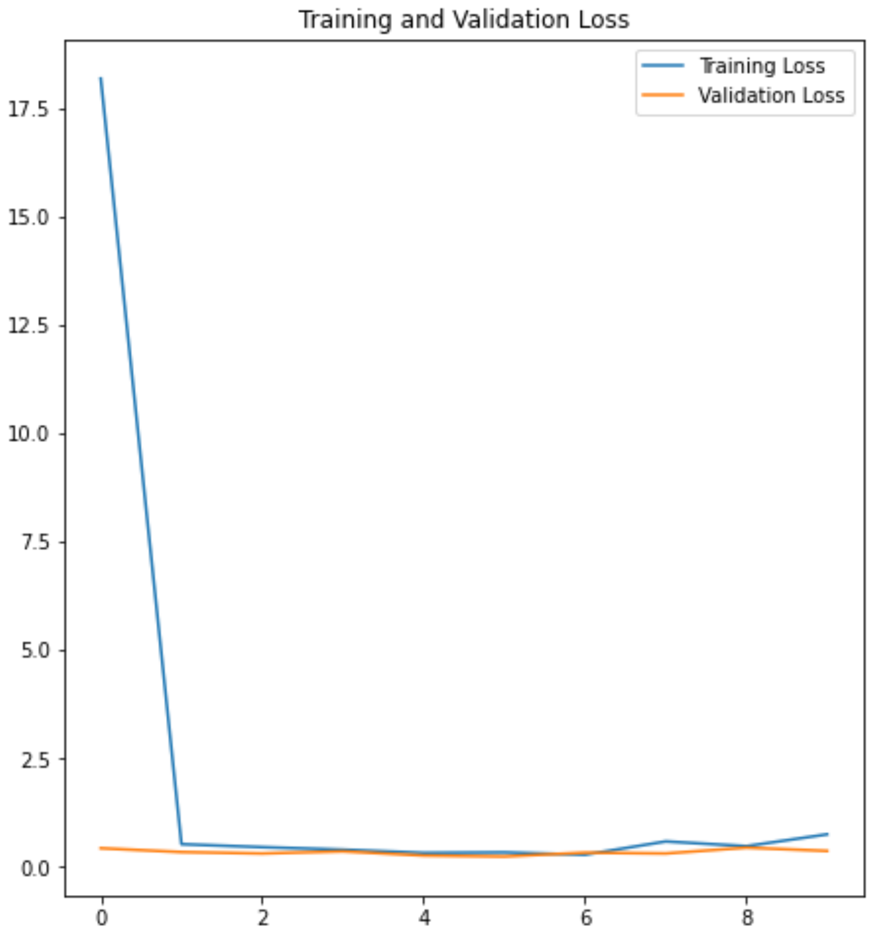
- Two tower with user-input levels (0 and 2 as example)



MODEL PERFORMANCE – SINGLE TOWER

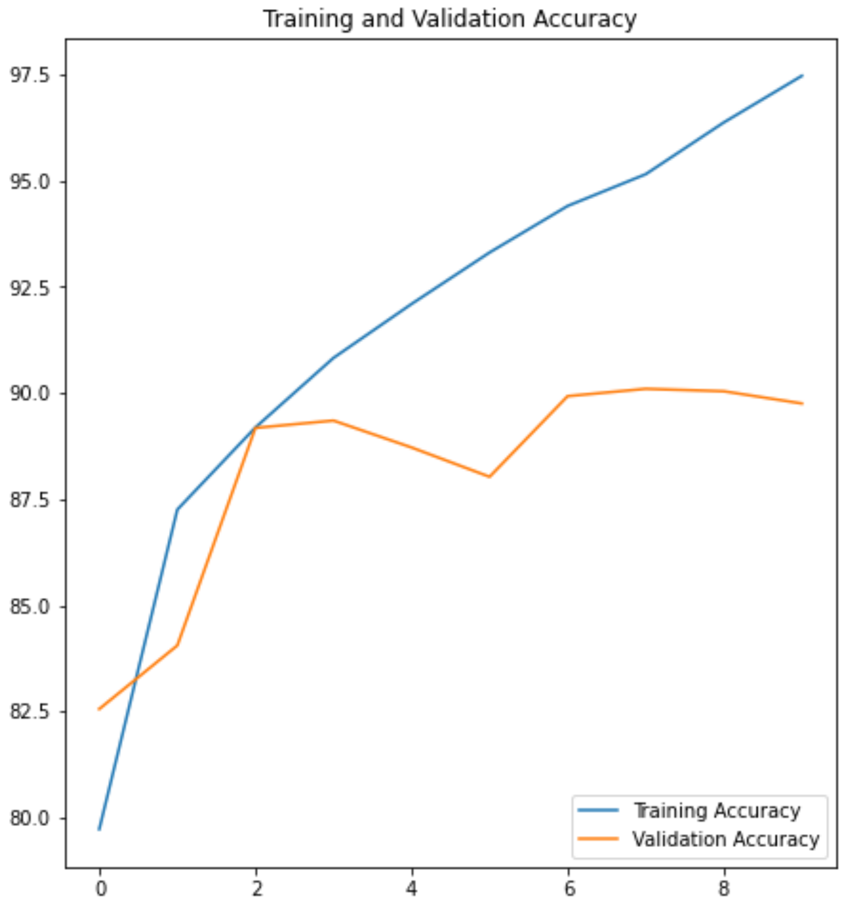
- Single tower (level 0) with transfer learning (InceptionV3)
- Train Accuracy: 88.15, Val Accuracy 90.73

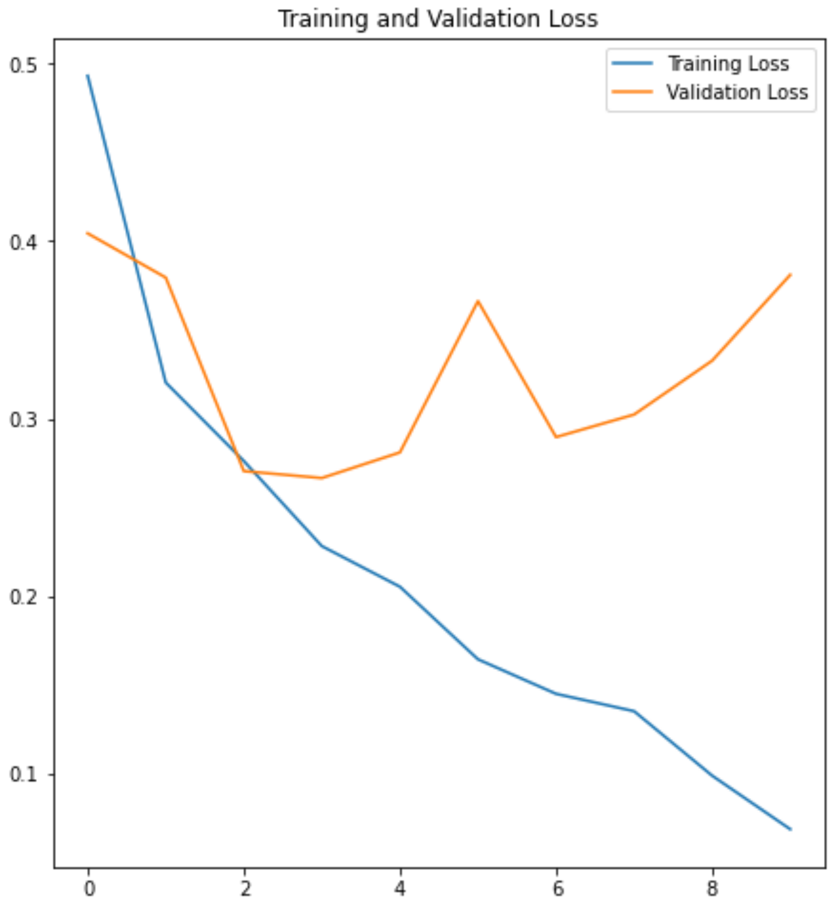




MODEL PERFORMANCE – SINGLE TOWER

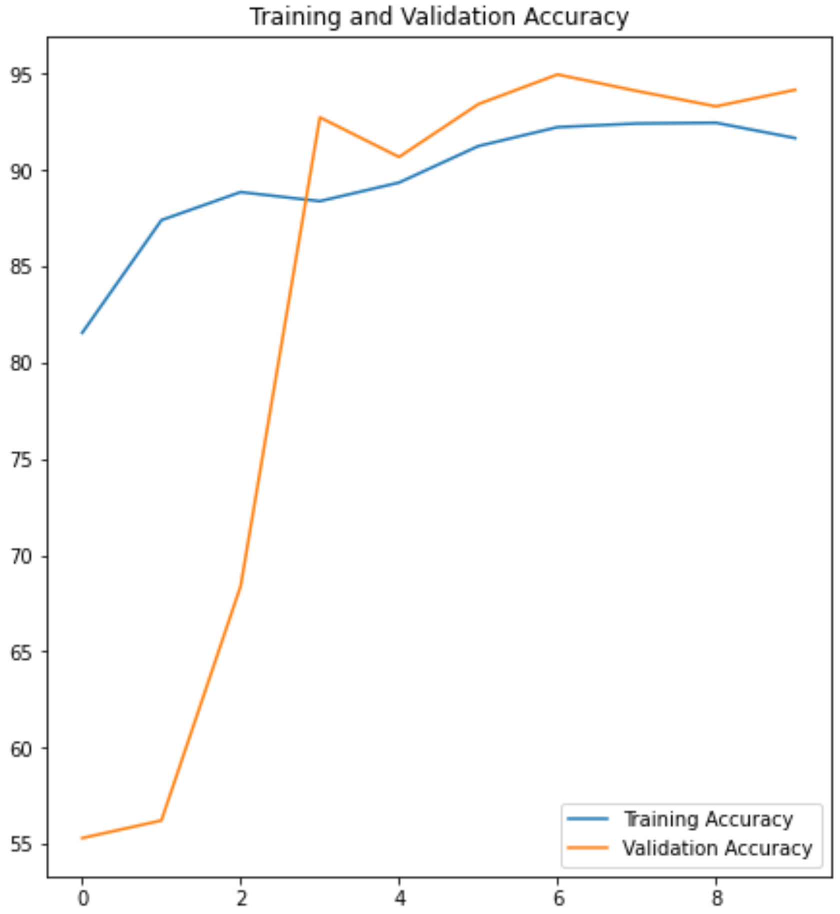
- Single tower (level 0) self-defined model
- Train Accuracy: 95.15, Val Accuracy 90.10

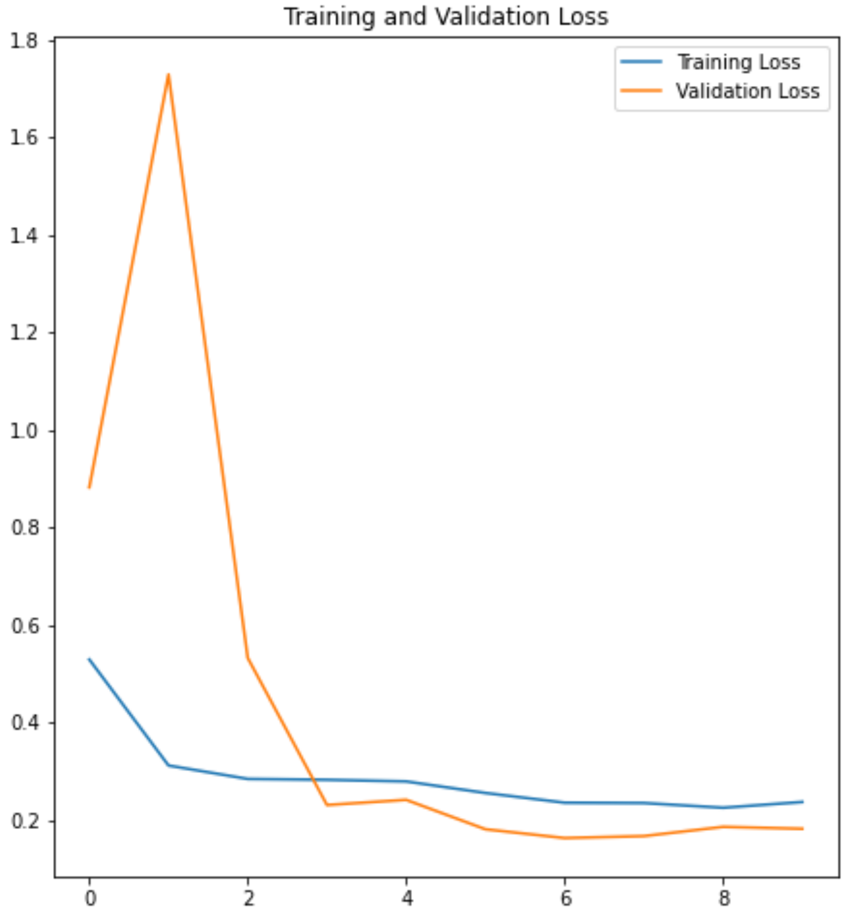




MODEL PERFORMANCE – TWO TOWER

- Best performing model: two tower (level 0 and 2) self-defined model
- Train Accuracy: 92.23, Val Accuracy 94.97





MODEL PERFORMANCE

- Best performing model: two tower (level 0 and 2) self-defined model
- Test Accuracy: 92.6

True Label \ Prediction	Predicted Positive	Predicted Negative
Labeled Positive	**46.2%**	**<u>3.8%</u>**
Labeled Negative	**<u>3.6%</u>**	**46.4%**

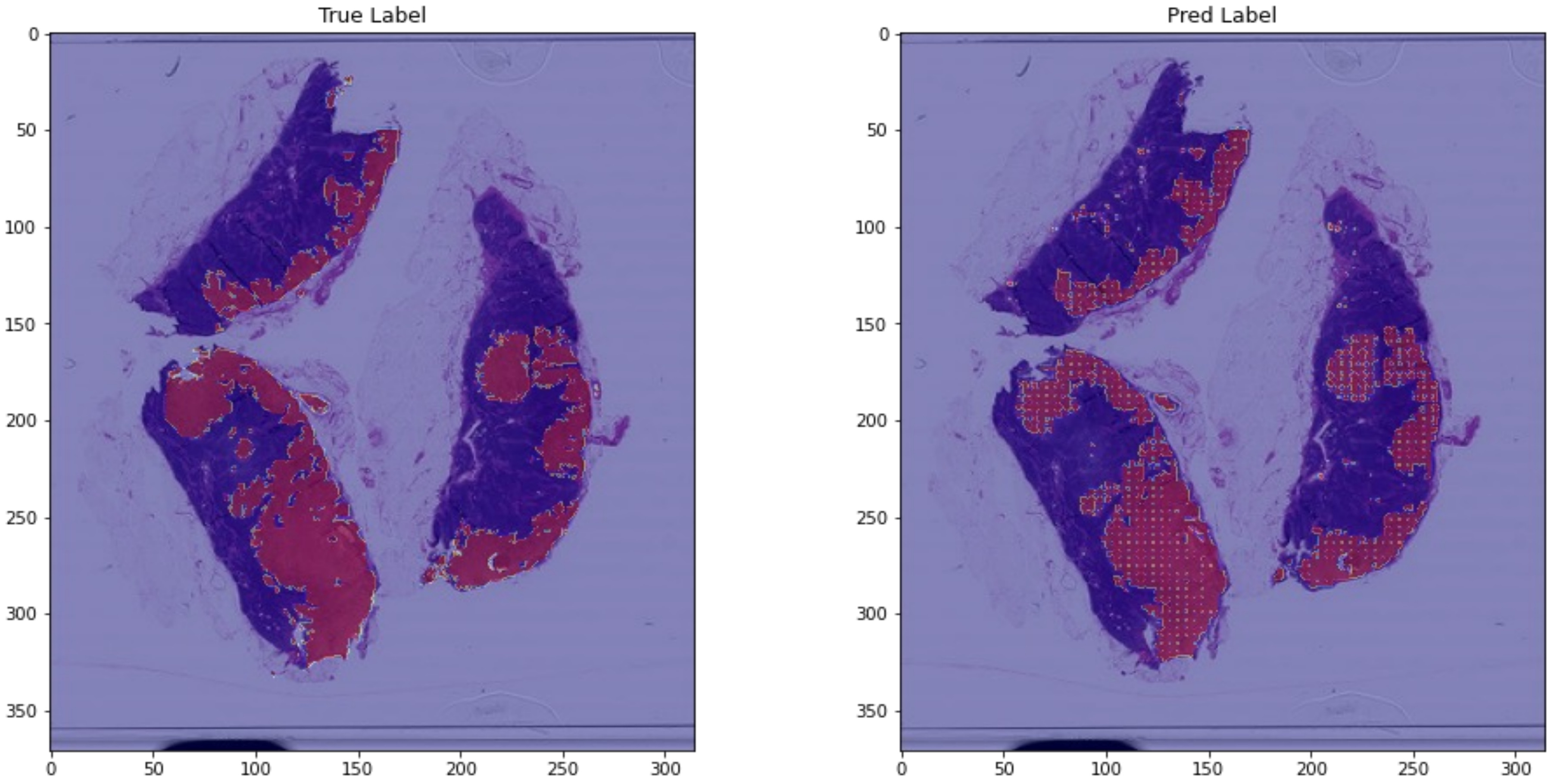
- A good balance between FP and FN

HEATMAP GENERATION

Non-tissue part pre-filtering → Batch-processing for prediction → Predicted mask generation and overlayed

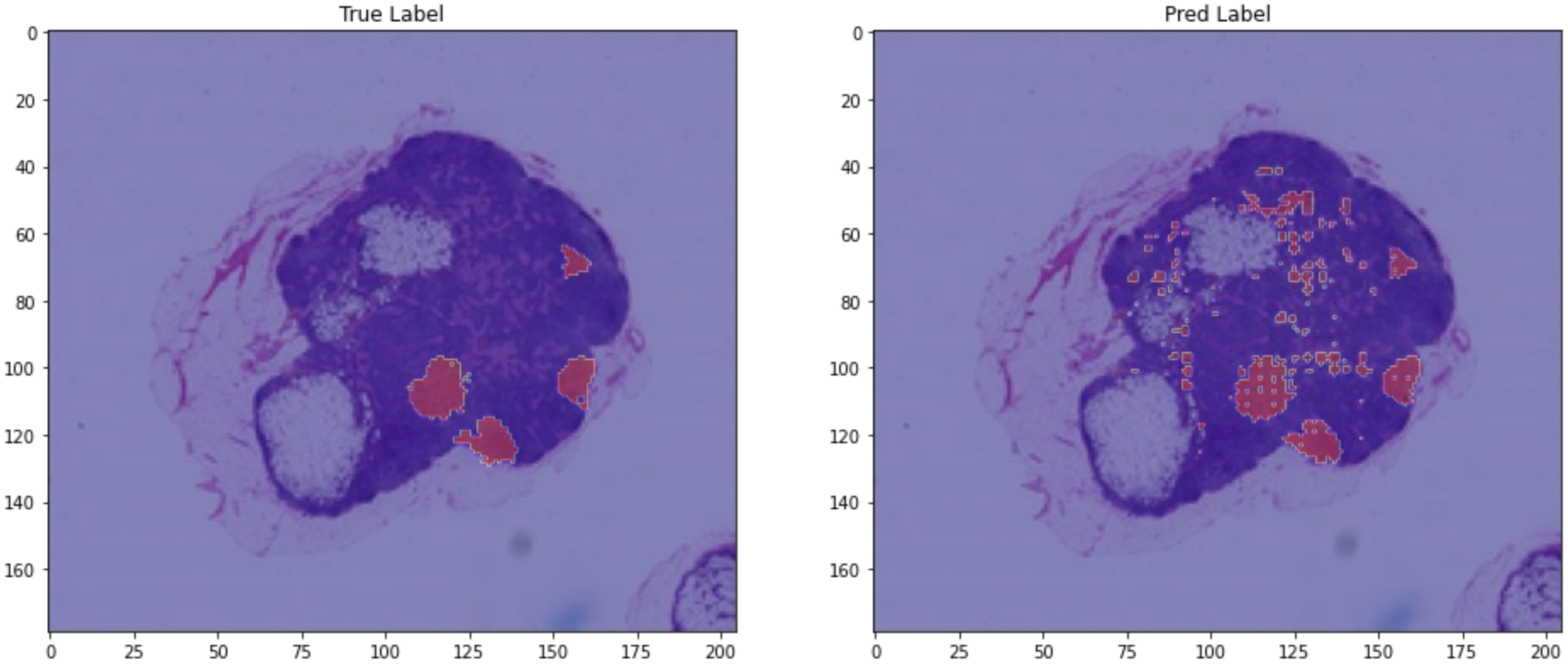
PREDICTION HEATMAP

- Heatmap generated on slide 78 (train slide)



PREDICTION HEATMAP

- Heatmap generated on slide 91 (test slide)



CODE WALKTHROUGH

OBSERVATIONS AND NEXT STEPS

- Model performs well on train slides and test sets but generated some false positives when inferring on whole test slides
- Data quality (largely) determines performance of a model
- Current approach of defining tissue patches could be improved
- With sufficient data, improvement from data augmentation is minor
- Advanced models require a lot more computing resources
- Computational intensive to generate predicted mask with high resolution
- Class balancing and model calibration remain to be challenges