Technical University of Denmark

P28: Automating the Segmentation of X-ray Images with Deep Neural Networks

Julius Olander (s203225), Nima Taghidoust Sourkouhi (s222395) and Andreas Papanikolaou (s232477)

DTU Compute · Technical University of Denmark Kgs. Lyngby, Denmark

DTU Compute
Department of Applied Mathematics and Computer Science

Introduction

- We apply an existing Convolutional Neural Network framework (UNet) to an X-ray image segmentation problem
- We investigate how the model performs under different circumstances, to assess and improve upon its implementability in a real-world practical setting. We do this by:
  - Adding noise to training and validation data.
  - Modifying the size of the training dataset to test how the model performs on various subsets.

Model Specification

UNet for multiclass image segmentation

The structure of U-Net consists of a contracting path, responsible for context capture, and a symmetric expanding path, enabling precise localization. [1]

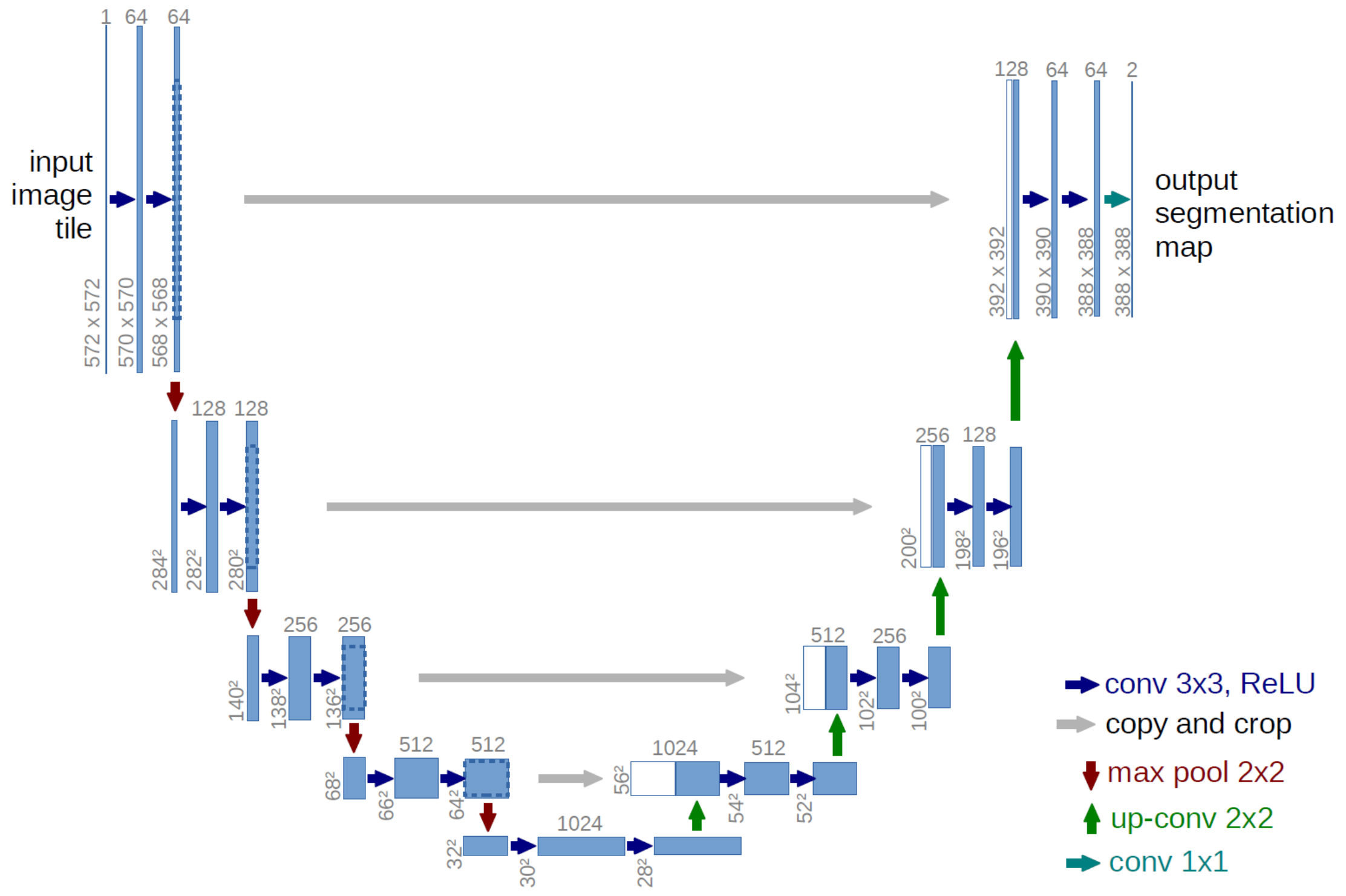

Figure 2: UNet Structure

- **Contracting Path**
  The contracting path serves to capture contextual information from the input image. It follows a typical convolutional network architecture with repeated applications of two 3x3 convolutions (unpadded convolutions). Each convolution is followed by 2D batch normalization, a rectified linear unit (ReLU) and a 2x2 max-pooling operation with a stride of 2 for downsampling. At each downsampling step, the number of feature channels is doubled.
- **Expanding Path**
  The expanding path is symmetric to the contracting path, and it facilitates precise localization. Each step in the expanding path consists of:
  Upsampling: Upsampling of the feature map through a 2D transposed convolution. Concatenation: Concatenation with the correspondingly cropped feature map from the contracting path.
  Two 3x3 Convolutions: Each followed by batch normalization and a ReLU unit.
  At the final layer of the expanding path, a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes.
  This design choice eliminates the need for fully connected layers, enhancing efficiency.

Dice Loss for Evaluation

In the context of image segmentation, the Intersection over Union (IoU or Dice score) gauges the similarity between two binary classification masks and has values between 0 and 1. The primary objective is to increase the similarity between these two masks, and this process is formalized as the minimization of the of the Dice loss function which is $Dice_{loss} = 1 - Dice_{score}$. [2] In our case of multiclass segmentation with 3 classes we calculate the dice score for each class treating it as binary classification with the rest of the classes used as background. The multiclass dice score is calculated as the average of the single class scores. The calculation for binary classification is as follows.

$$\text{DL2} = 1\text{ - }\left(\frac{\sum_{n=1}^{N} p_n r_n + \epsilon}{\sum_{n=1}^{N} p_n + r_n + \epsilon}\right) - \left(\frac{\sum_{n=1}^{N} (1-p_n)(1-r_n) + \epsilon}{\sum_{n=1}^{N} 2 - p_n - r_n + \epsilon}\right)$$

We studied and modified the following implementation https://github.com/milesial/Pytorch-UNet of UNet to conduct our experiments using Google Colab and DTU's HPC cluster. The following hyperparameters were used for the experiments:

Optimizer: RMSProp (L2 weight regularization, momentum)

LR scheduler: ReduceLROnPlateau (patience=5)

Loss fuction: CrossEntropy loss & multiclass dice loss

Normalization: Batch norm, gradient clipping by norm

Experiments

Experiment 1: Baseline

Training Data: 400 random images from original dataset

Validation Data: 100 random images from original dataset

Procedure: Train the model on the complete original training dataset and evaluate its performance on the original validation dataset.

Result: 99.105% validation dice.

Experiment 2: Noisy Images Addition

Noises are added by Gaussian distribution with standard deviation of 5000 in 16-bit image.

Training Data: 540 random images from original training dataset and newly transformed noisy images

Validation Data: 60 random images from original training dataset and newly transformed noisy images

Procedure: Train the model on a combination of original and noisy images and evaluate its performance on the original and noisy validation dataset (200 epochs).

Result: 99.08% validation dice.

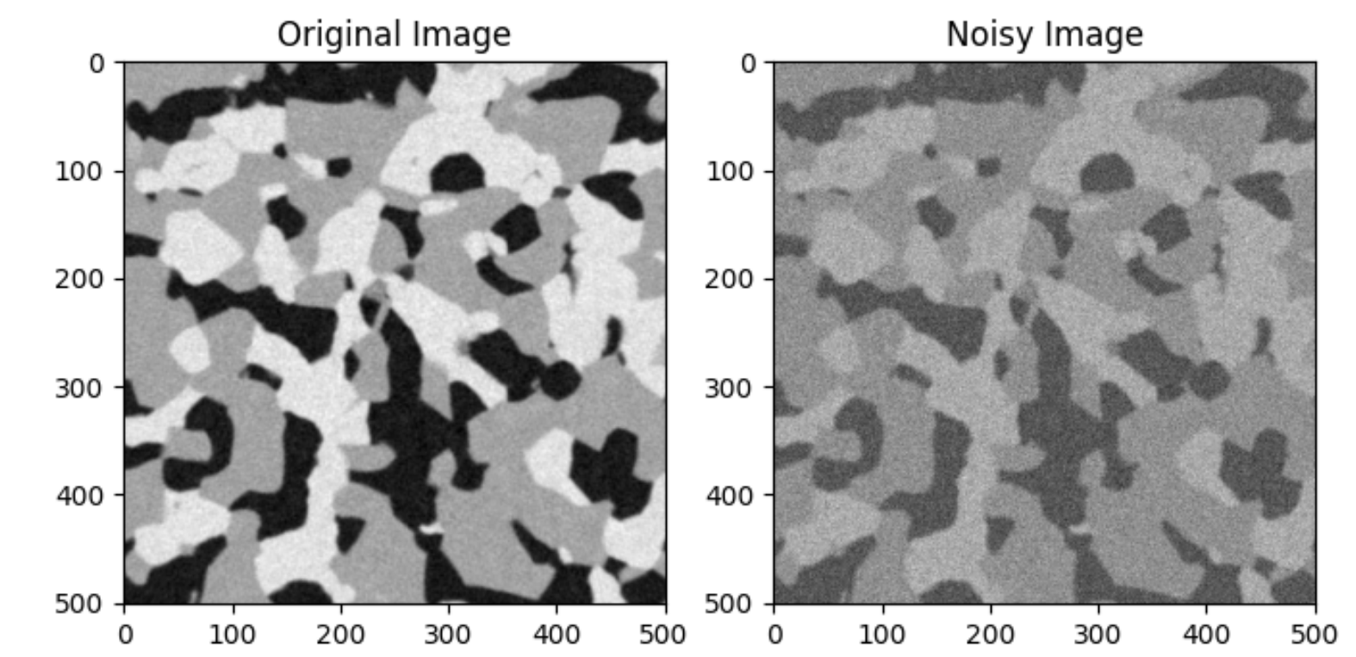

Figure 3: Original Image vs transformed Noisy Image

Experiment 3: Noisy Images Only in Validation

Training Data: 480 random images from original training

Validation Data: 60 random images from original training dataset + 60 random images from newly transformed noisy images

Procedure: Train the model on the original training dataset but validate on a dataset containing original and noisy images.

Result: 66.48% validation dice was the highest score before overfitting and declining to 63.65%

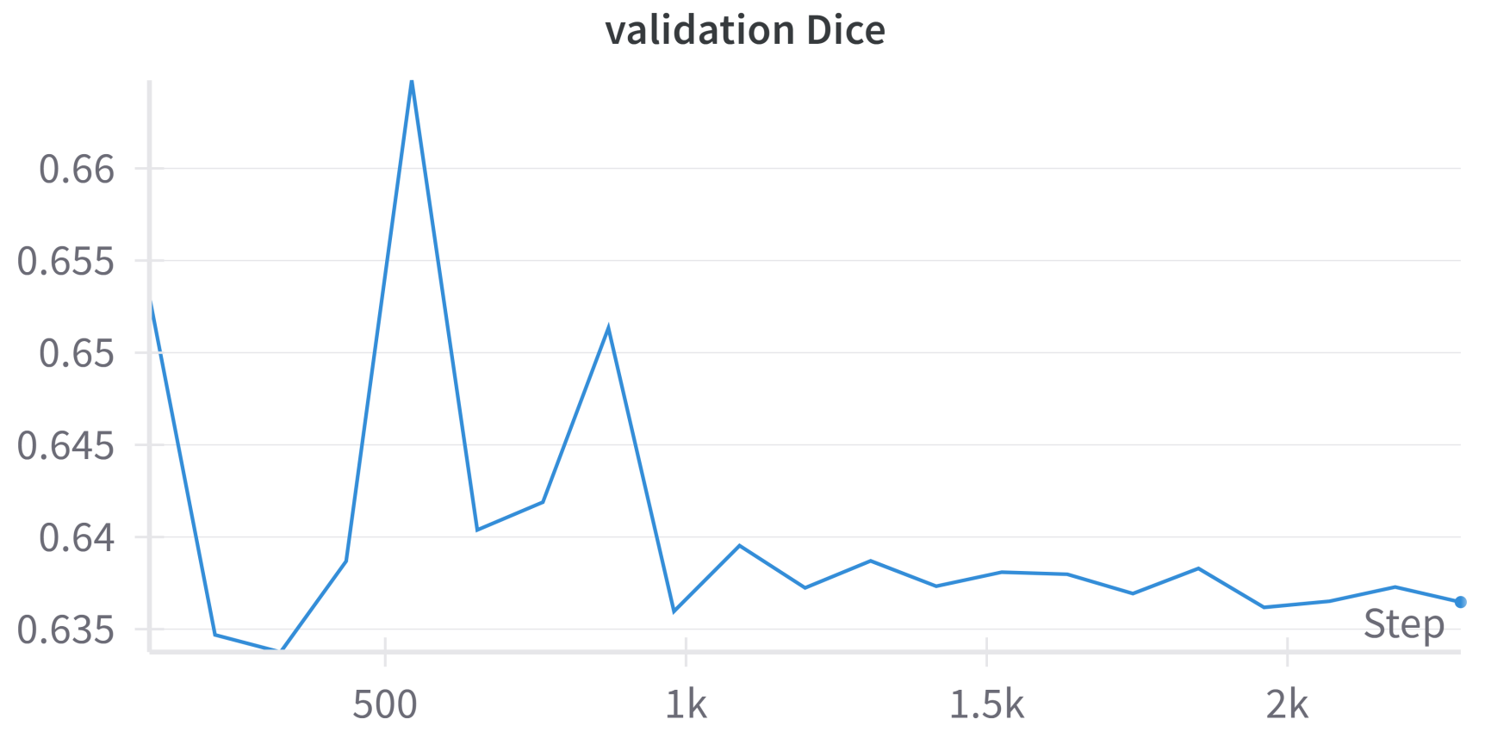

Figure 4: Validation Dice for "Noisy Images Only in Validation" Experiment

Experiment 4: Variation in Training Dataset Size

Training Data: Different subsets of rows from the original training dataset (varied dataset sizes). The range of images used is from 5-105 (step of 5 until 25, step of 20 up to 105)

Validation Data: Original validation dataset

Result: Even with the lowest examined number of images (5 rows in the training dataset), the model achieved a validation dice of 98.667%

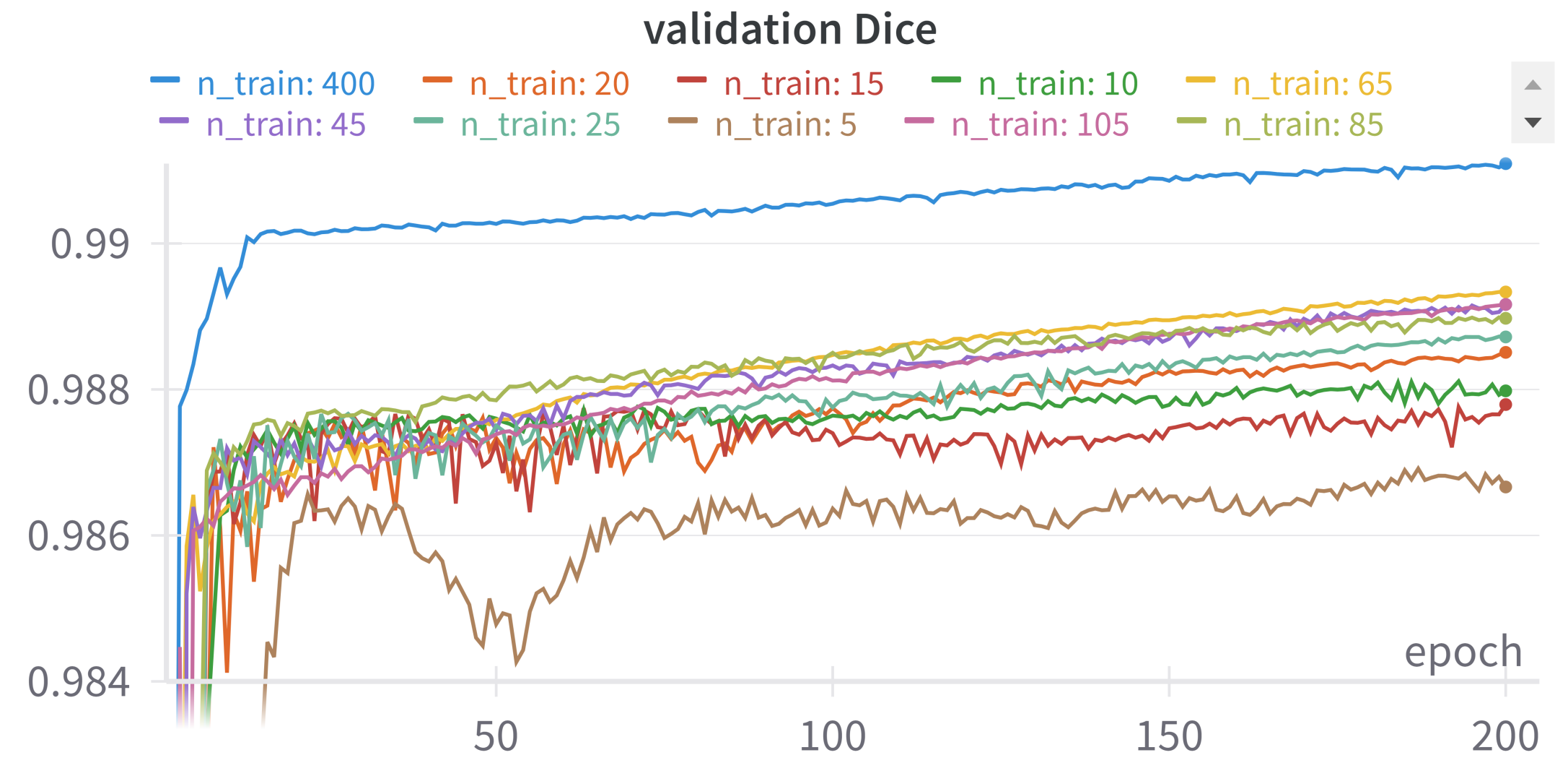

Figure 5: Validation Dice for "Training Dataset Size" Experiment

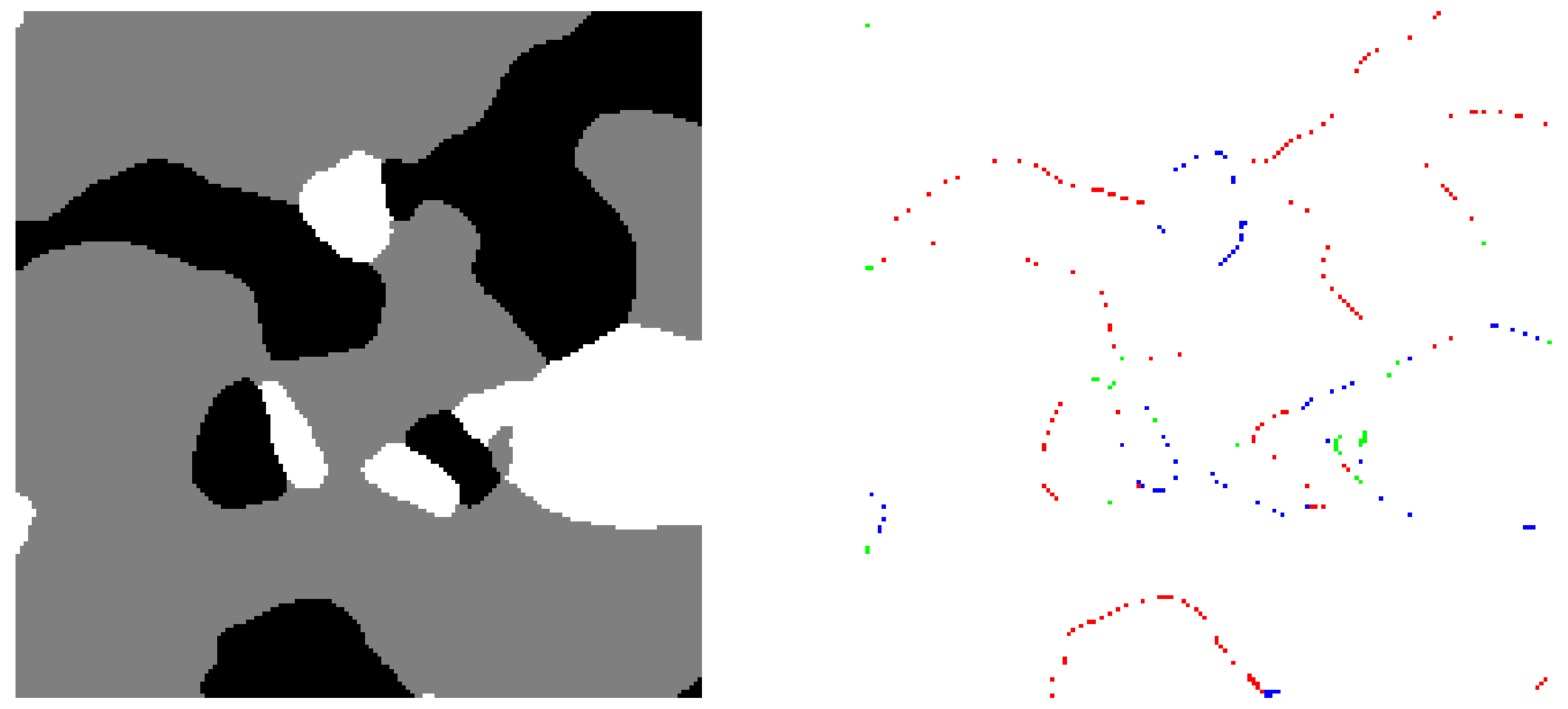

Figure 6: True mask for a validation image with $n_{train} = 5$ and visualization of missclassified pixels, (R, G, B) true class is (0, 1, 2).

Conclusions & Outlook

- Applying the UNet framework to the problem at hand has proven to be a highly robust solution
- Model is sensitive to noise in the validation data if not trained on such. To enhance robustness for its practical implementation, the model should be trained on noisy data as well.
- To gain a satisfactory validation Dice loss and score, the model can be trained on significantly less data than the original 400 labeled images used. For real-world practical application, discovering where this threshold lies is essential. We found out that using as little as 5 images you reach a validation dice score of 98.667%, which compared to the baseline score of 99.105% marks a percentile difference of 3.3% and an absolute difference of 0.438.
- Currently performing additional robustness checks to find the balance between a highly accurate model and a minimum viable product solution. Checks under development include:
  - training with a single image's augmentation transformations
  - cropping and using images of smaller size
  - training only with noisy images to find the threshold of diminishing returns and then check the efficacy of augmentation transformations with respect to lowering the threshold
  - using the predictor with the best noisy performance to predict the labels of the lis dataset and visualize the faults for visual inspection and subjective scoring
- Through further development we are interested in expanding the model to handle 3D image segmentation.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[2] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS*