DLMIA

Praktikum 1

In the first part of the lab we needed to build a U net with keras framework.

In the preprocessing part I rescaled the images to 320\*320 pixels.

My training was done with test.py and some exploration done with the jupyternotebook.

My training configuration was

BATCH\_SIZE = 32  
EPOCHS =50  
LEARNING\_RATE = 3e-3

I experimented with more learning rates but smaller learning rates didn’t converge in 100 epochs.

Without any data augmentation this is the result on validation data with the best model I achieved these results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Jaccard Dist. | Dice Coef. | Sensitivity | Specicifity | Accuracy |
| 0.0795 | 0.93 | 0.78 | 0.99 | 0.92 |

With Data augmentation(horizontal flipping and rotating the images) the numbers are much better:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Jaccard Dist. | Dice Coef. | Sensitivity | Specicifity | Accuracy |
| 0.04 | 0.97 | 0.93 | 0.98 | 0.96 |

Prediction with the best model

A black and white image of a map

Description automatically generated

With more time and more computing power I think I could get even better predictions with more epochs

By applying various transformations to my dataset, such as image rotation or flipping, I was able to significantly increase the diversity and size of my training data. This, in turn, helped my machine learning models become more robust and better at generalizing patterns. Data augmentation not only made my models less susceptible to overfitting but also allowed them to capture a broader range of real-world variations