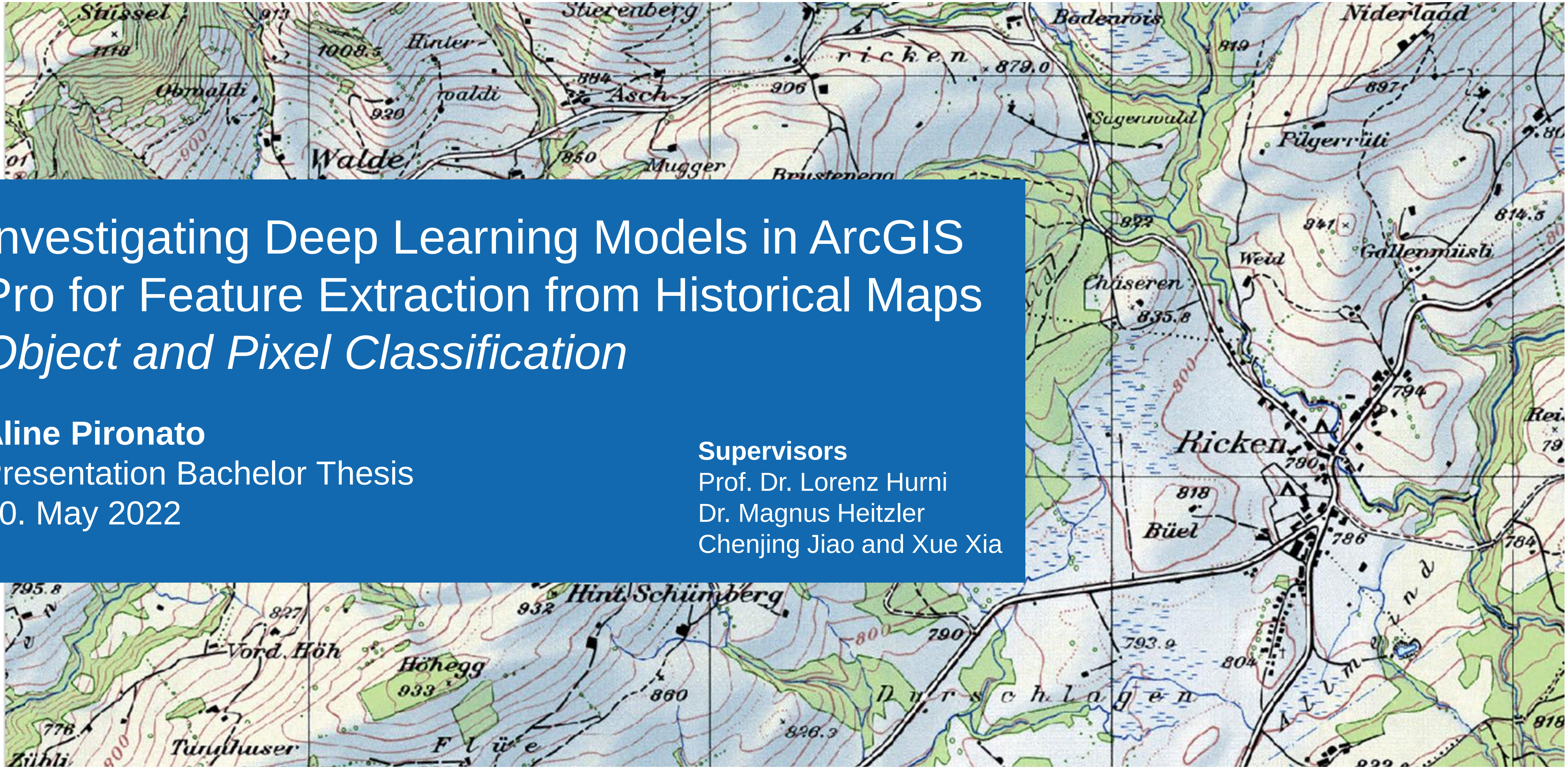
ETH zürich

IKG
Institut für Kartografie
und Geoinformation



Investigating Deep Learning Models in ArcGIS Pro for Feature Extraction from Historical Maps

Object and Pixel Classification

Aline Pironato
Presentation Bachelor Thesis
30. May 2022

Supervisors
Prof. Dr. Lorenz Hurni
Dr. Magnus Heitzler
Chenjing Jiao and Xue Xia

Introduction

Introduction

- Deep Learning uses neural networks as well as large amounts of data to make predictions
  → Its application led to a lot of progress in image processing technologies
- Use of historical maps plays an important role in a wide array of scientific disciplines
  → Use those emerging technologies to extract features from historical maps
- Wetlands have varying shapes and sizes
  → Adapt to those circumstances by using Neural Networks
- Deep Learning models are already being successfully applied in standard programs
  My task is to evaluate the available Deep Learning Models of ArcGIS Pro in their suitability with respect to the extraction of wetlands

Problem Definition & Research Question

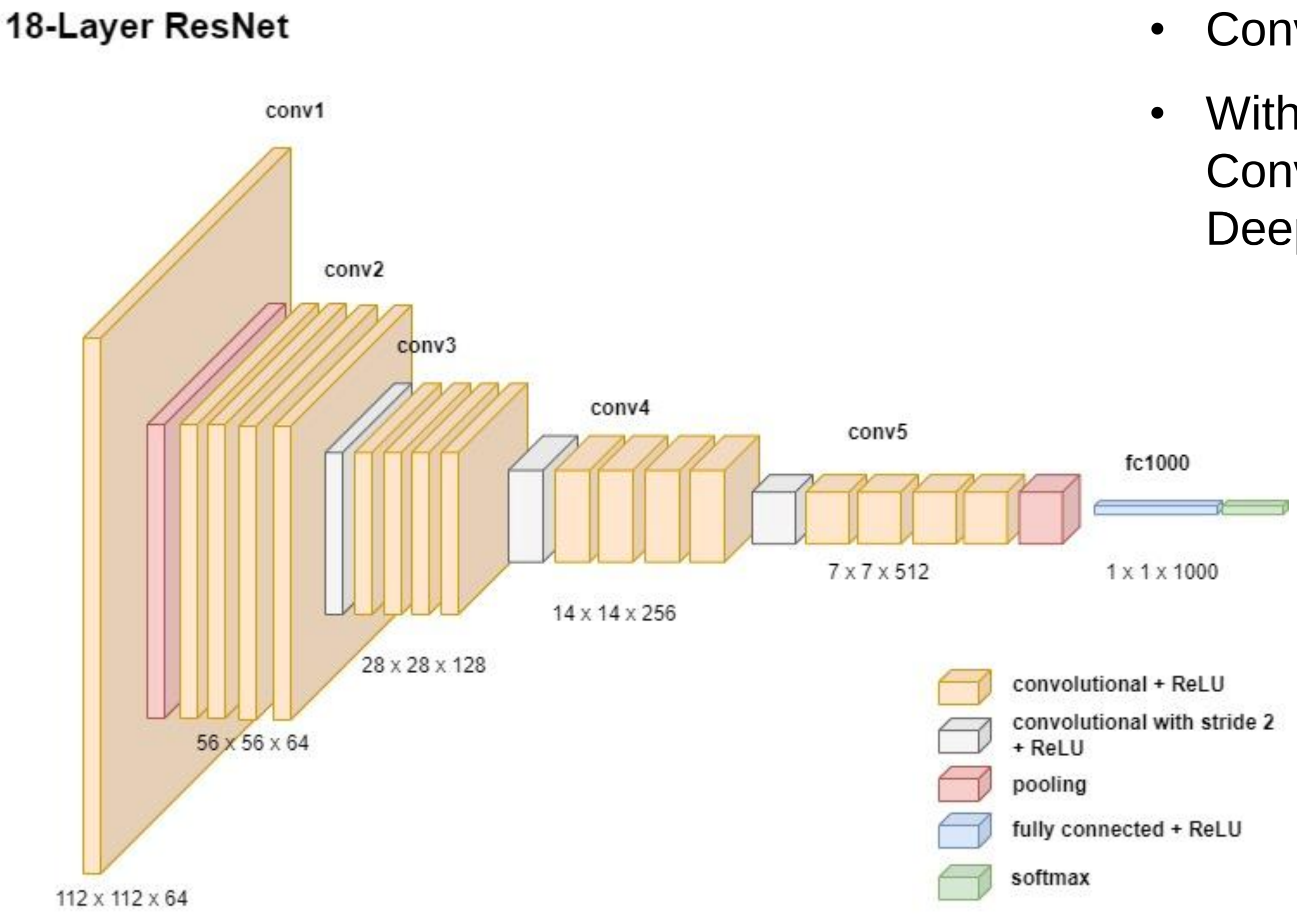
Which Deep Learning Model, based on pixel or object classification, from the ArcGIS Pro catalogue provides the best results when extracting wetlands from historical maps?
What is the associated workflow?

Additional Questions to discuss:

- To what extent do the models differ? How clearly can they be classified according to performance?
- How should the input data be prepared? What is the best way to augment the raster data?
- How will performance be measured & compared? How should metrics be weighted?
- Are all operations and procedures done within the ArcGIS Pro application or does it need support from external programs?
- What are the values of the hyperparameters? Are there significant differences between the models?

Theory

Convolutional Neural Network



Architecture of a ResNet-18 Backbone Model. *Own illustration*

- Convolutional layers followed by a pooling layer
- With enough Layers we have a Deep Convolutional Neural Network, which makes it Deep Learning

Pixel Classification Models

U-Net Classifier

DeepLabV3

Pyramid Scene Parsing Network

BDCN Edge Detector

Holistically-Nested Edge Detector

Multi Task Road Extractor

ConnectNet

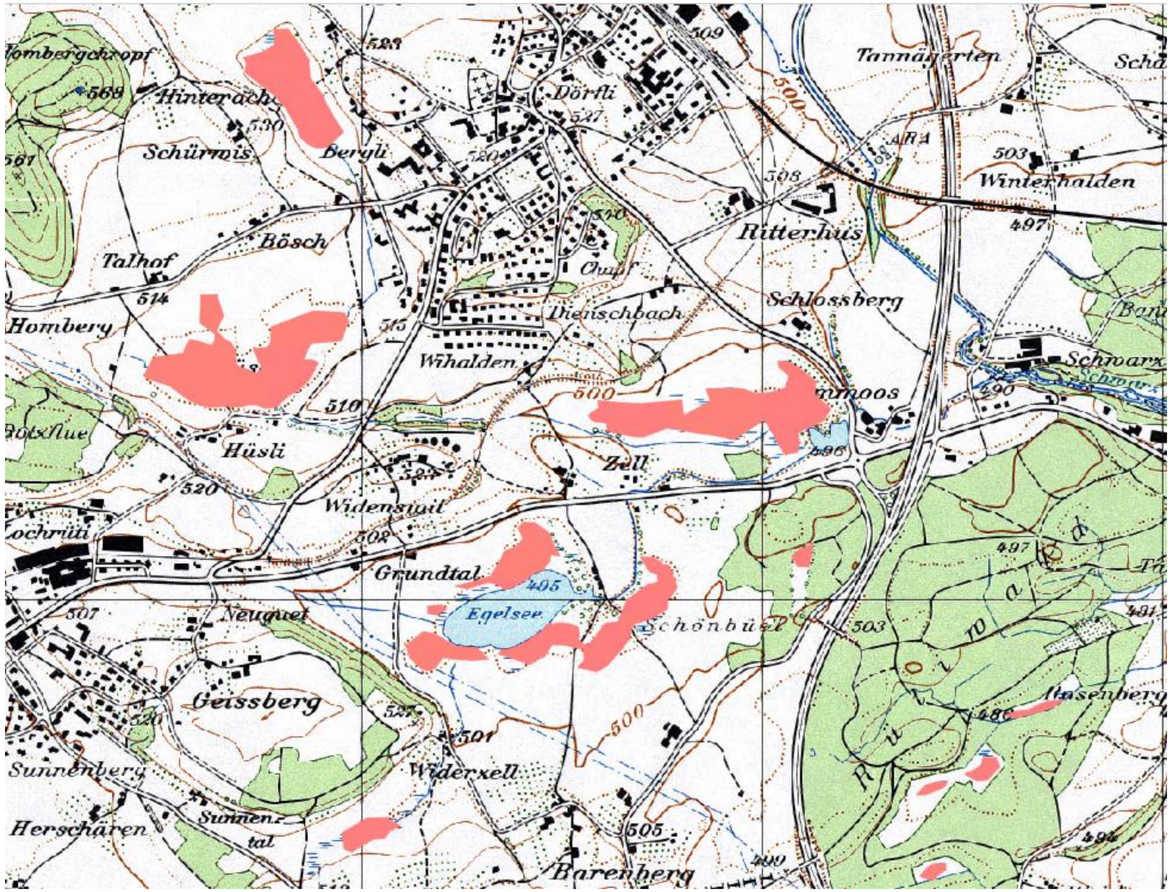
Change Detector

Object Classification Model

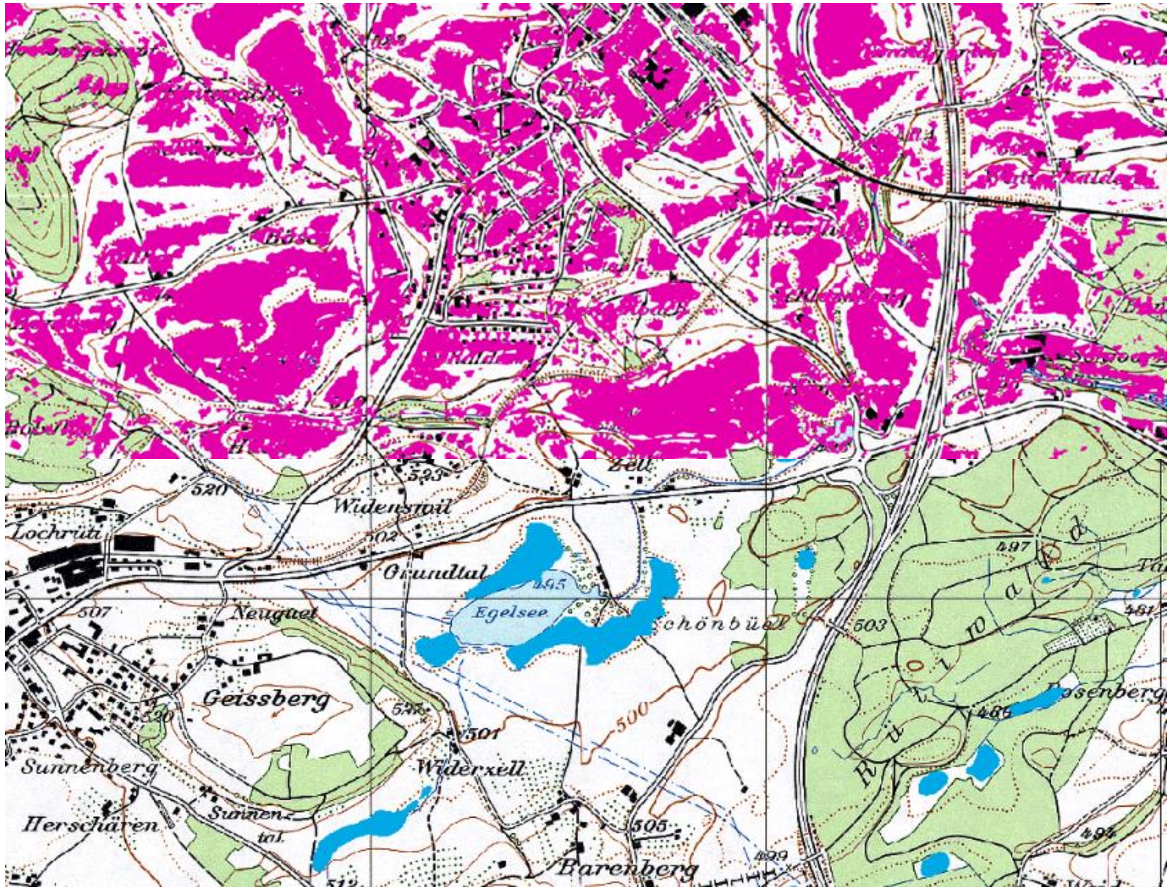
Feature Classifier

Comparison of Holistically-Nested Edge Detector and DeepLabV3

Ground truth



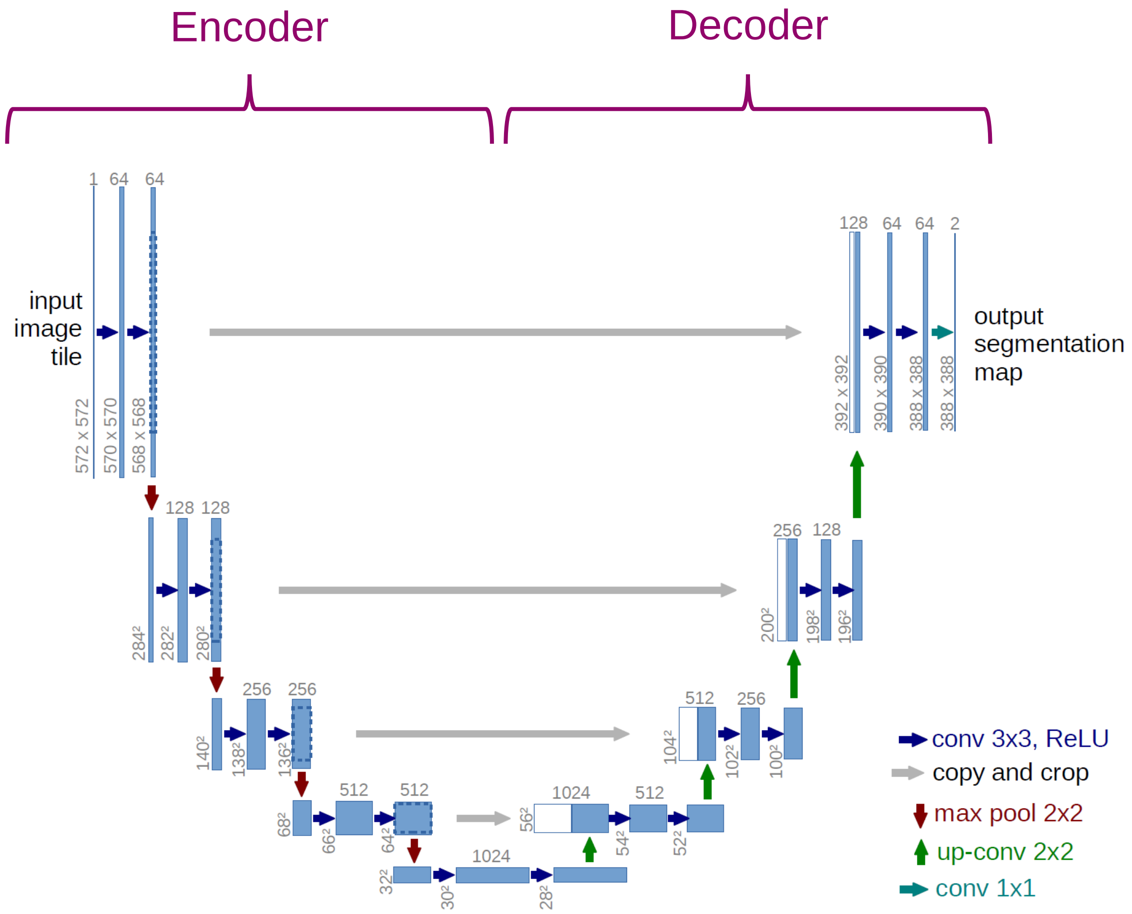
HED Model compared to DeepLabV3 Model



U-Net

- Convolutional Neural Network
- Class label assigned to each Pixel
- Two symmetric halves:
  Encoder with pooling operations
  Decoder with upsampling operations
  → Image resolution is retained
- Requires little training data
- Basis for many other image segmentation Models

In the context of my thesis?
→ *Developed for & applied in similar tasks*

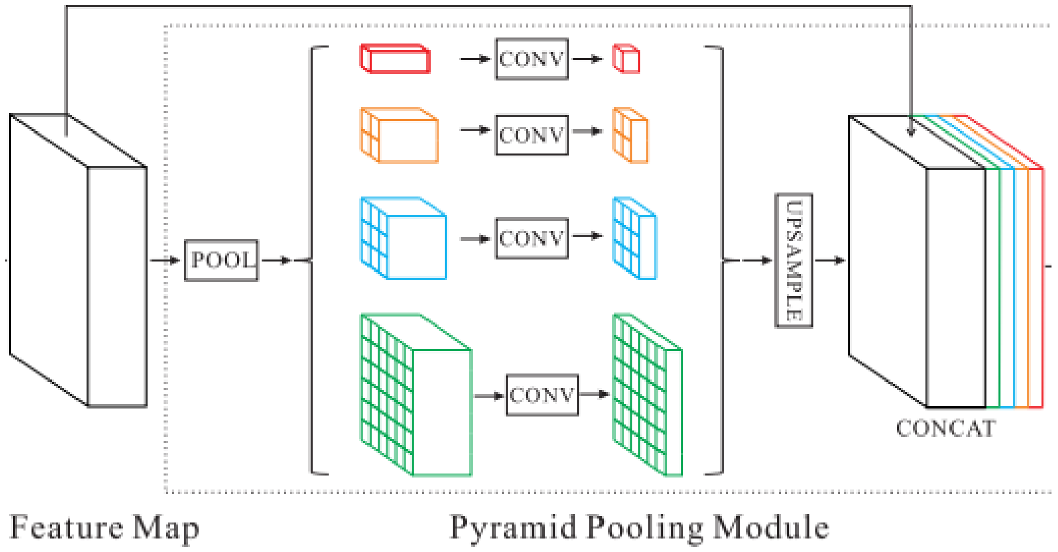


U-Net architecture. [1]

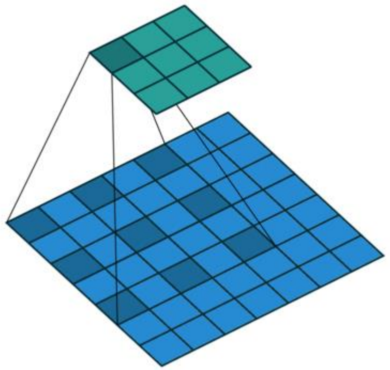
Pyramid Scene Parsing Network PSPNet

- Fully Convolutional Neural Network
- Pyramid Pooling Module with four different Pyramid scales
  Small sizes for big features
  Big sizes for small features
- Dilated network strategy to enlarge field of view
  → Atrous Convolution
- Upsampling with decoder like in U-Net
  → image Resolution is retained
- Considers global context when classifying

In the context of my thesis?
→ *No particular benefits for task at hand*



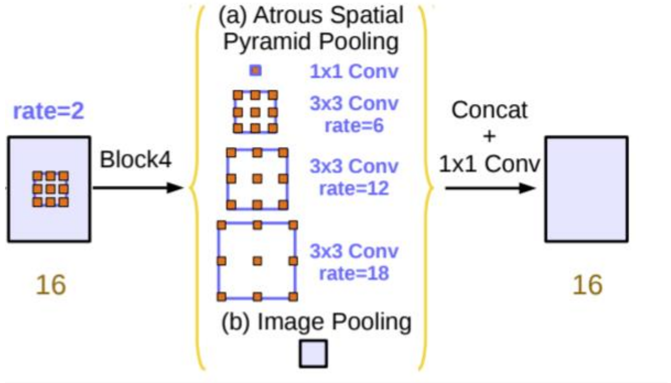
PSPNet Model Structure. [2]



Dilated network strategy [3]

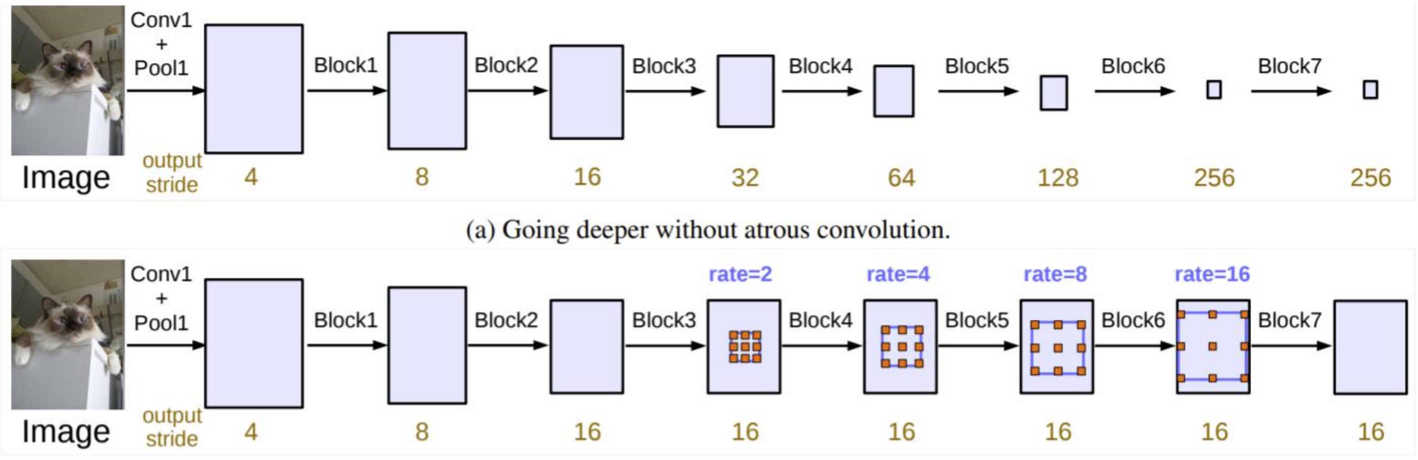
DeepLabV3

- Fully Convolutional Neural Network
- Controls the size of the feature map via Atrous Convolution
  → preserves some spatial resolution
- Useful for Deep Convolutional Neural Networks
- To obtain multi-scale context information,
  an Atrous Spatial Pyramid Pooling network
  is added to classify each pixel



ASPP network [4]

In the context of my thesis?
→ *Beneficial, if depth is required*



(a) Going deeper without atrous convolution.

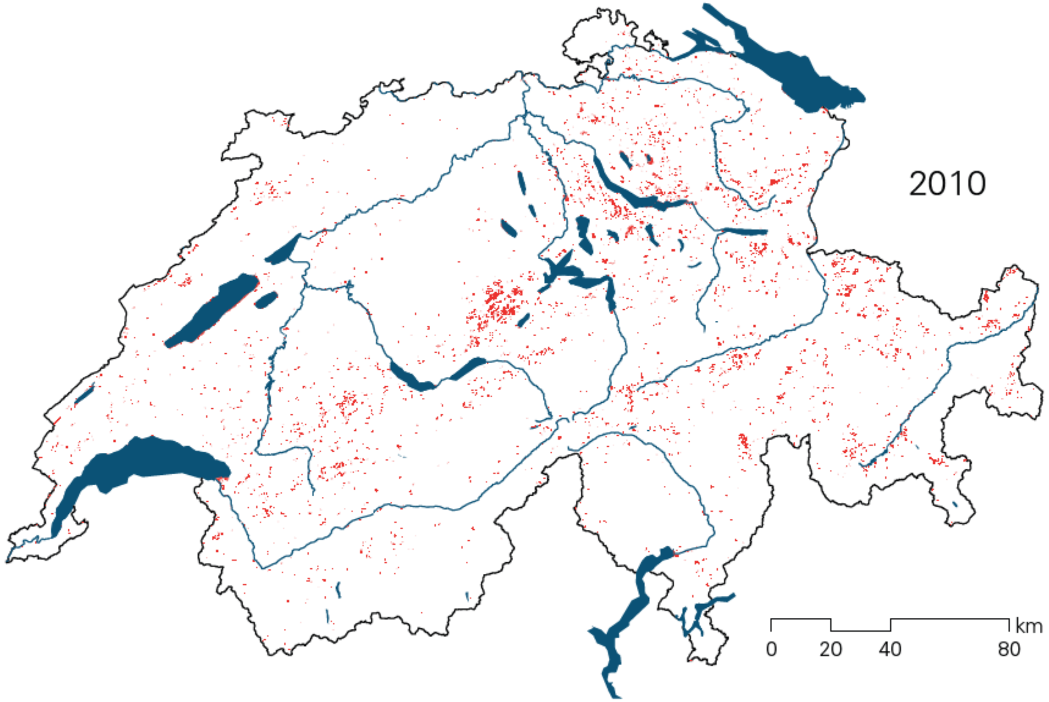
(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

DCNN without Atrous Convolution and with Atrous Convolution. [4]

Method

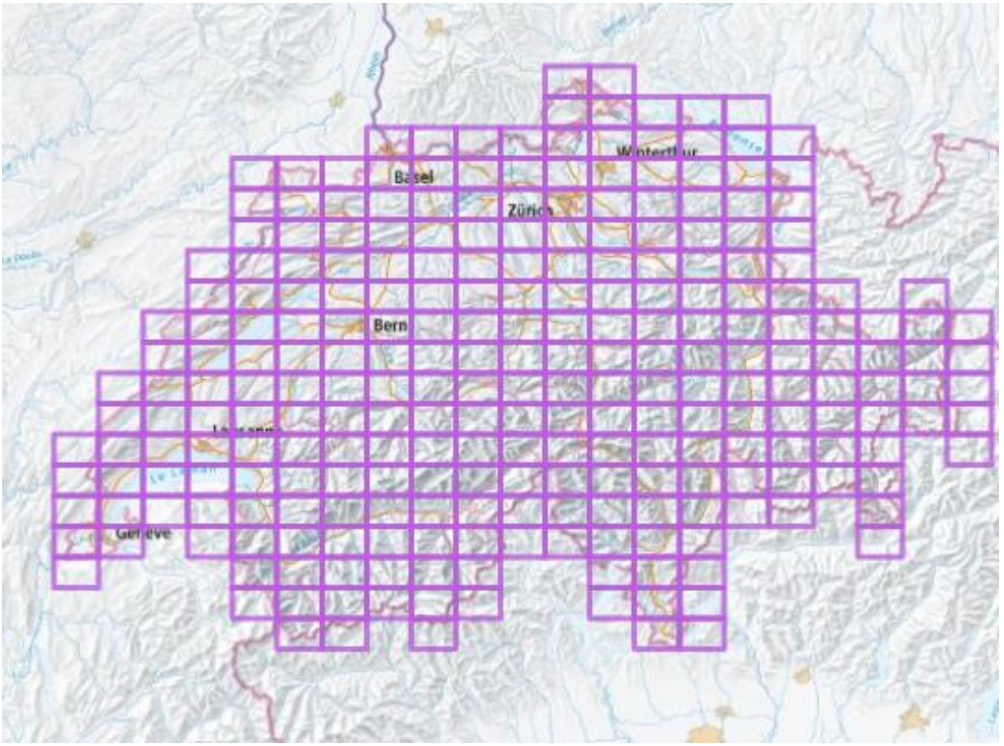
Data

- Feature Layer of a historical-cartographic reconstruction of the Swiss wetlands from 2010
- Ground truth and training data



Wetland feature layer used for training and validation. [6]

- 258 updated TIFF tiles of the Historical National Maps from 1952 at a scale of 1:25 000
- Data to classify



Subdivision of the TIFF tiles. [5]

Hardware

IKG TX 1070
Intel Xeon CPU
32 GB RAM
4 Cores and 8 Processors

GPU
NVIDIA GeForce GTX 1070
8 GB GPU Memory

Software

ArcGIS Pro 2.9.0
Toolbox: Image Analyst
Toolset: Deep Learning

Python
ArcGIS Pro Deep-Learning-Framework-Libraries
PyTorch Framework
ArcGIS API arcgis.learn module

Basic Approach & Workflow

Export Training Data for Deep Learning

- Exported Training Data from 10 tiles
  → 27’817 image chips containing wetlands

Train Deep Learning Model

- Trained approx. 40 different Models with varied hyperparameters
  Time duration needed for training ranged from 6 to 14 hours

Classify Pixels using Deep Learning

- Classified pixels from 7 tiles using those trained Models
  → tiles chosen to contain wide range of different environments

Statistical Evaluation

- Based on the performance on the validation dataset during training and the processed results of the pixel classification

Hyperparameters

- Model Type
- Maximum number of Epochs
- Batch Size
- Mixup
  Data augmentation to keep Neural networks from memorizing corrupt labels by mixing them up
- Focal Loss
  More Focus on hard misclassification
- Backbone Model
- Freeze Model
  Predefined weights and biases of the Backbone Model will not be altered

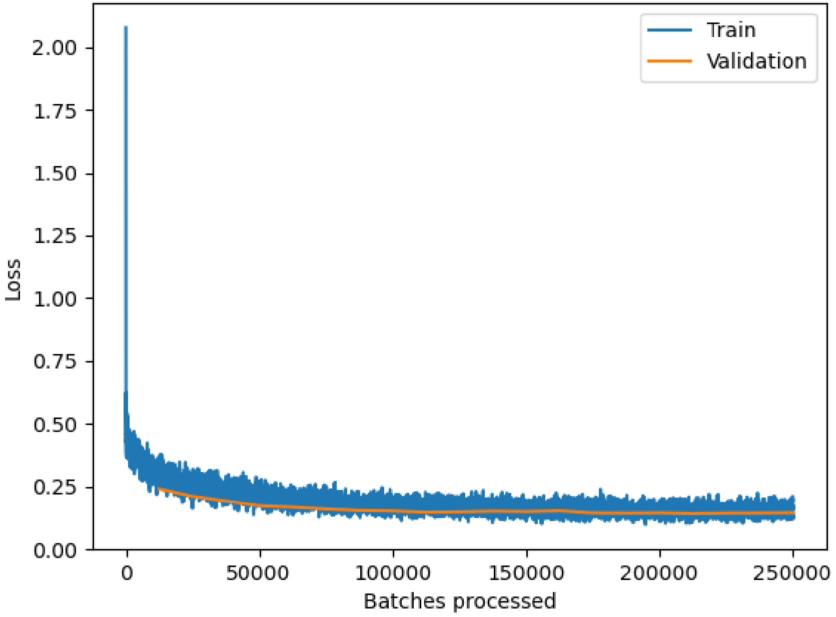
Parameters with the most drastic impact on the resulting model → Hence the parameters I varied

Statistical Evaluation

Automatic evaluation during training

- Train & Validation Loss function
- Precision, Recall, F1 Score and IoU

Exemplary Output for a DeepLabV3 Model



	NoData	Undefined
precision	0.963815	0.841727
recall	0.979599	0.746840
f1	0.971643	0.791450

Evaluation of classified map tiles

- No tools provided by ArcGIS Pro to asses performance of pixel classification

→ Applied spatial Operations to find *hits, correct rejections , type I* and *II errors*

Which allowed the calculation of metrics like

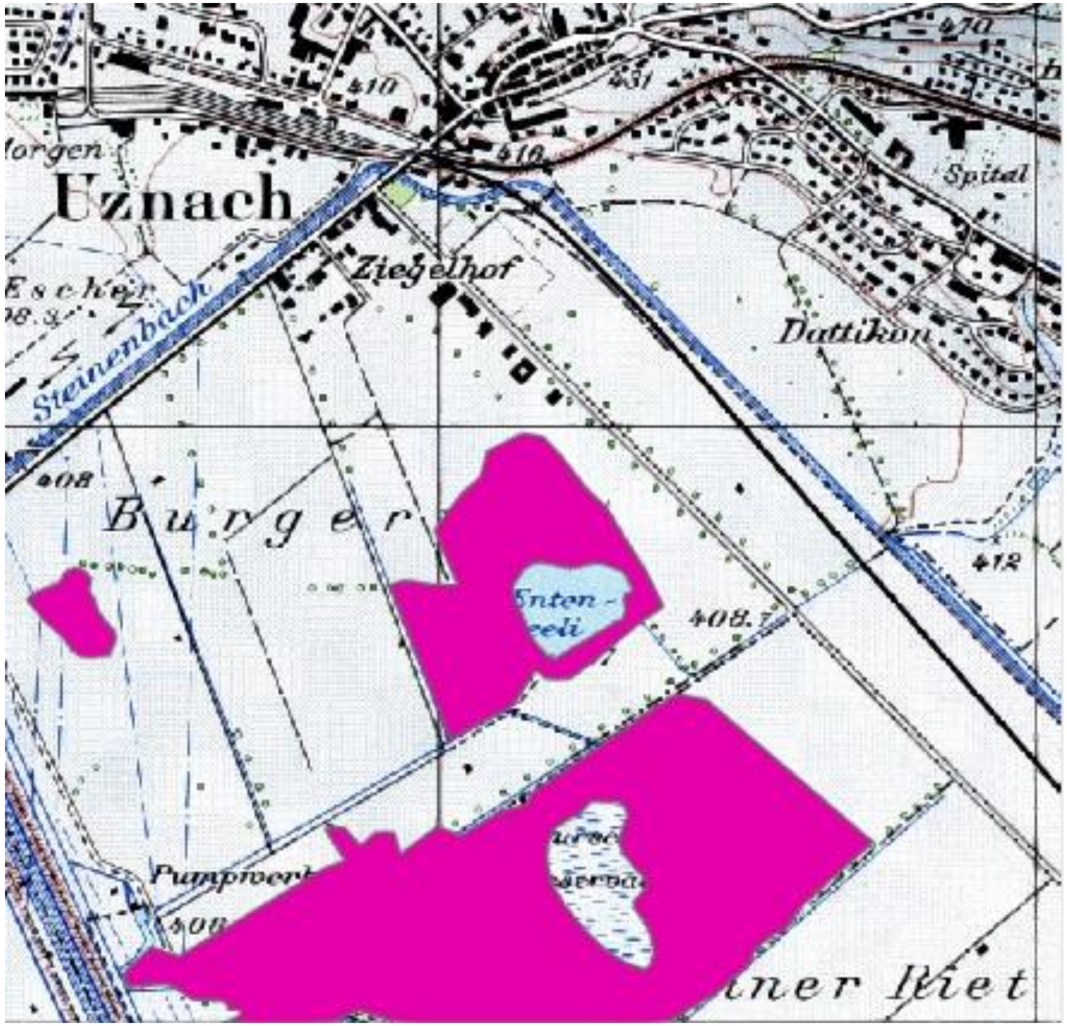
- Precision
- Recall
- F1 Score
- IoU
- Accuracy
- Selectivity
- *And others*

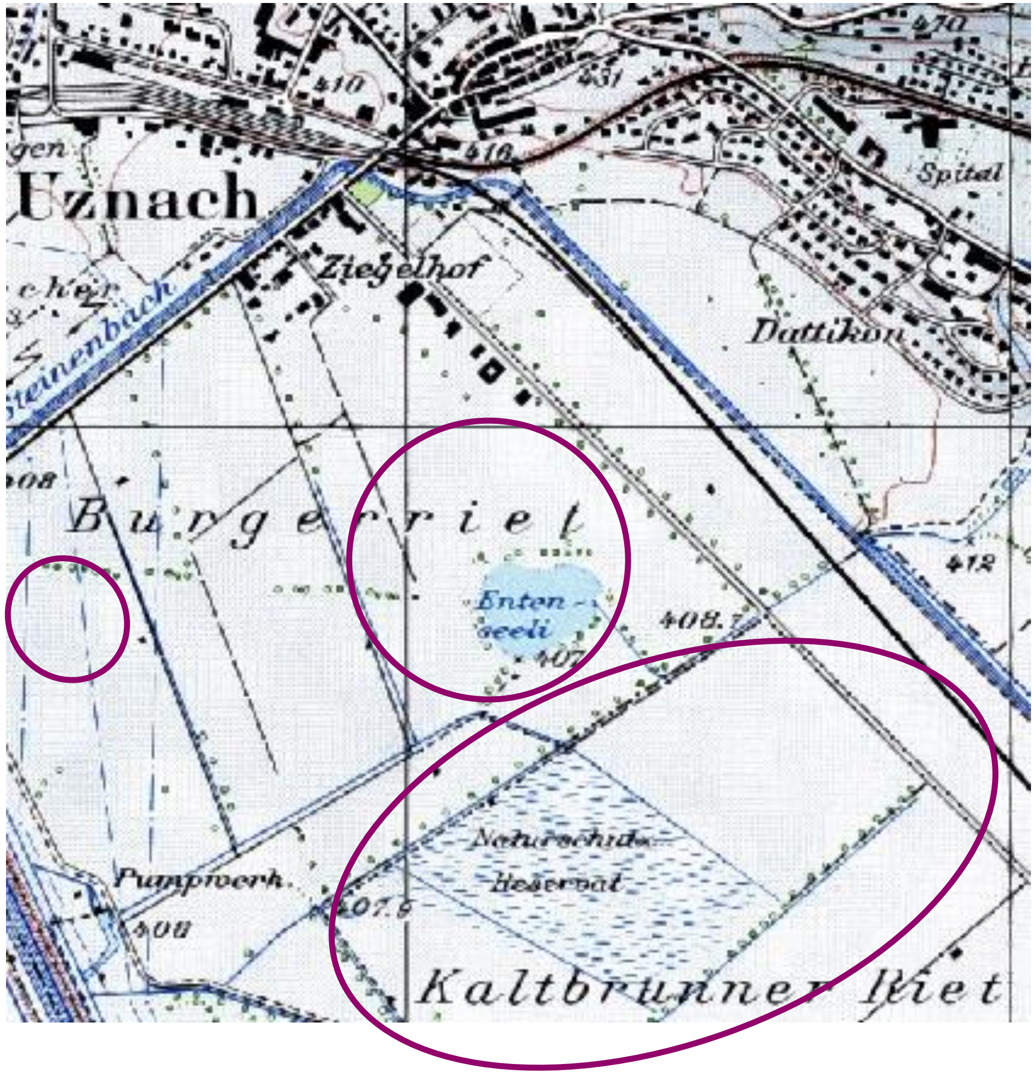
Results & Discussion

General Conclusions

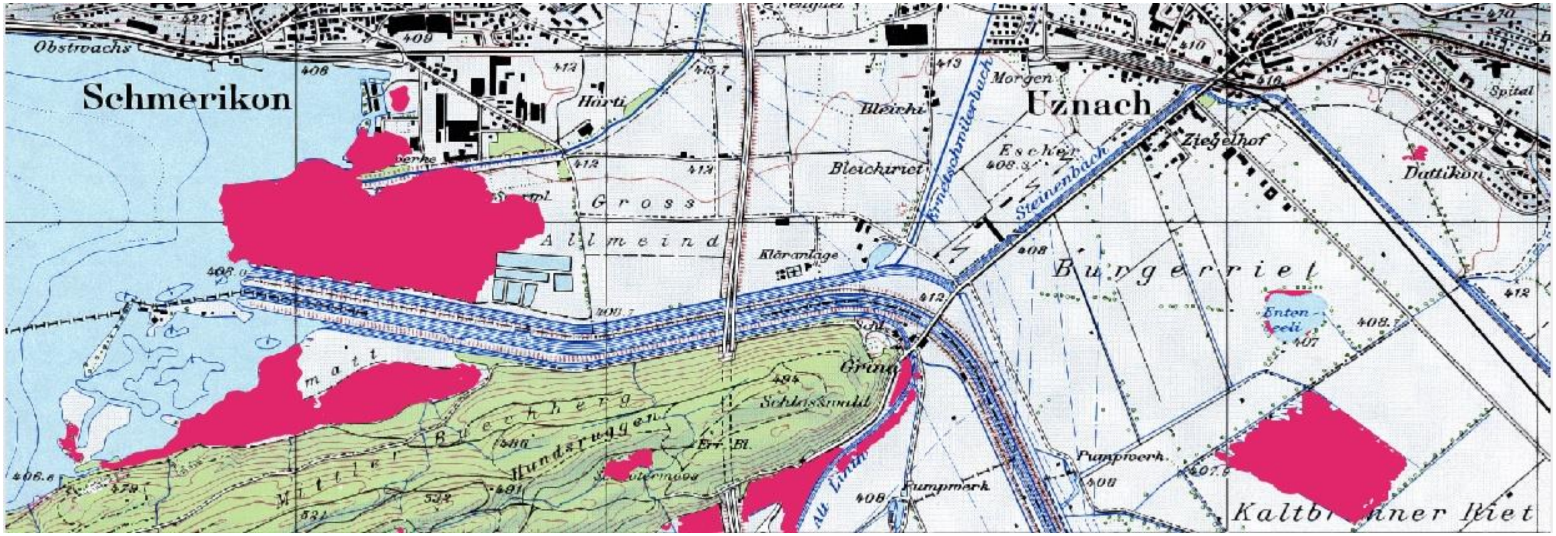
- All Models handled task well, independent of chosen hyperparameters
  → Very robust
- Ideal Workflow dependent on desired Output
  As close as possible to the training data?
  Convenient to integrate into a map?
  Large interconnected shapes or small clusters?
- More Layers does not necessarily mean better results
  → More often than not, the same Setup with a deeper ResNet model leads to a worse visual result and has lower statistical values
- Metrics shouldn't be taken at face value, if the ground truth dataset is flawed

Distorting data discrepancies





Results of the statistical evaluation



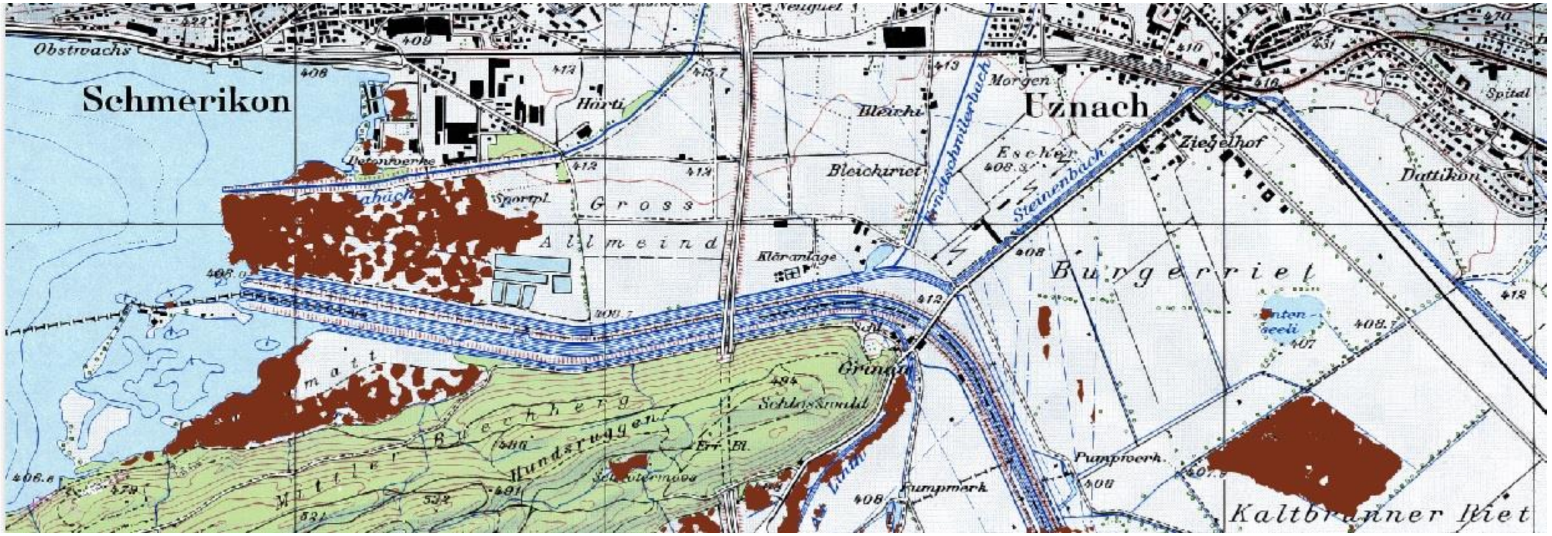
U-Net Model with best metrics:

Values from training

IoU	Precision	Recall	F1 Score
0.671	0.858	0.804	0.830

Values from evaluated pixel classification

IoU	Precision	Recall	F1 Score
0.366	0.710	0.275	0.536



U-Net Model with worst metrics:

Values from training

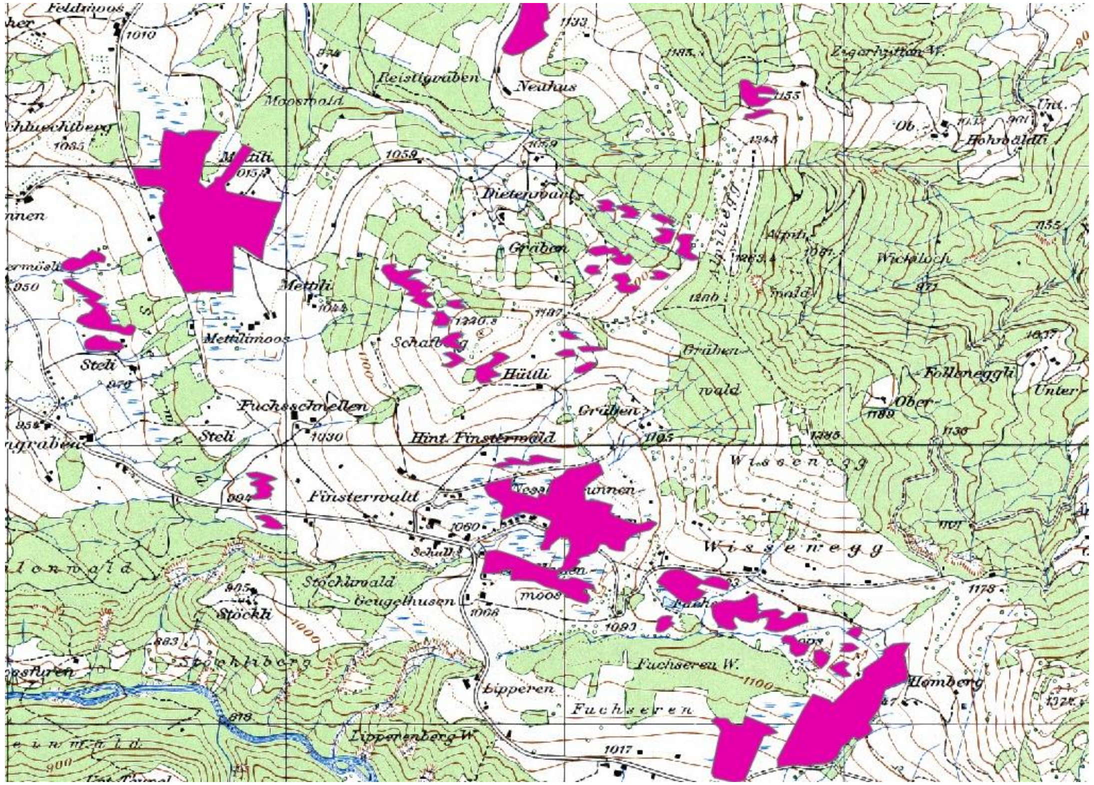
IoU	Precision	Recall	F1 Score
0.614	0.841	0.746	0.791

Values from evaluated pixel classification

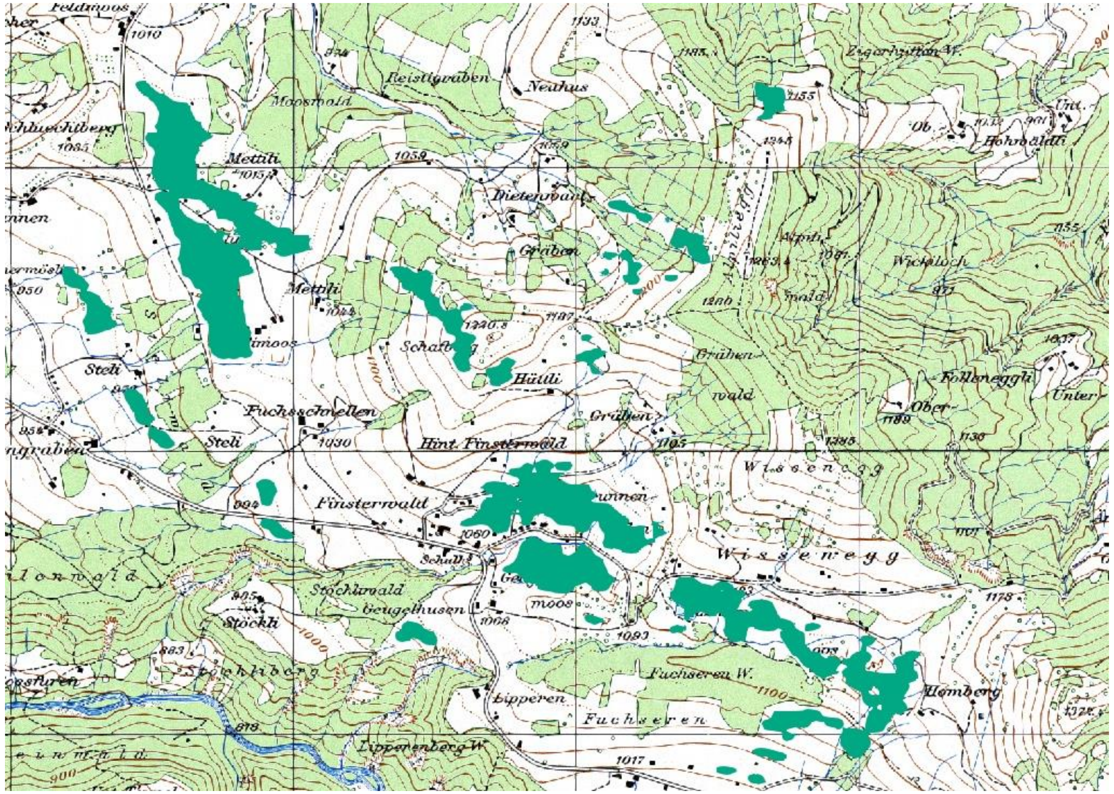
IoU	Precision	Recall	F1 Score
0.325	0.732	0.236	0.490

Effect on small & jagged areas

DeepLabV3 Model



Ground Truth



From Training

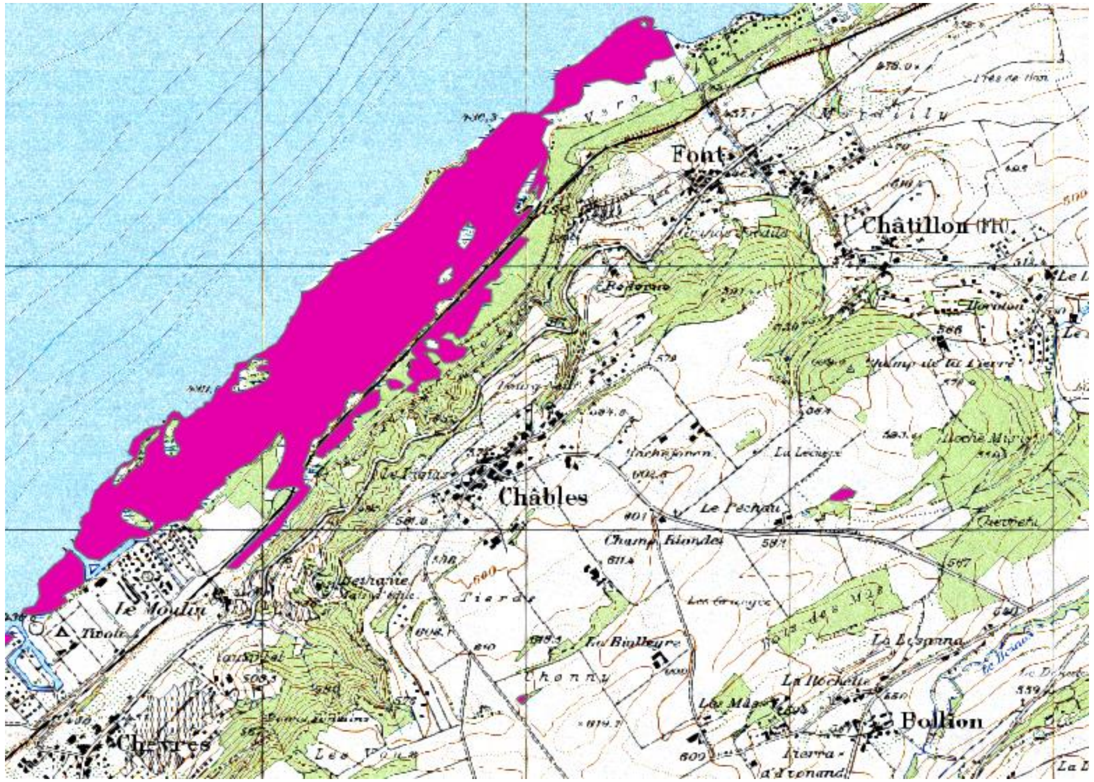
IoU	Precision	Recall	F1 Score
0.616	0.860	0.737	0.793

From Pixel Classification

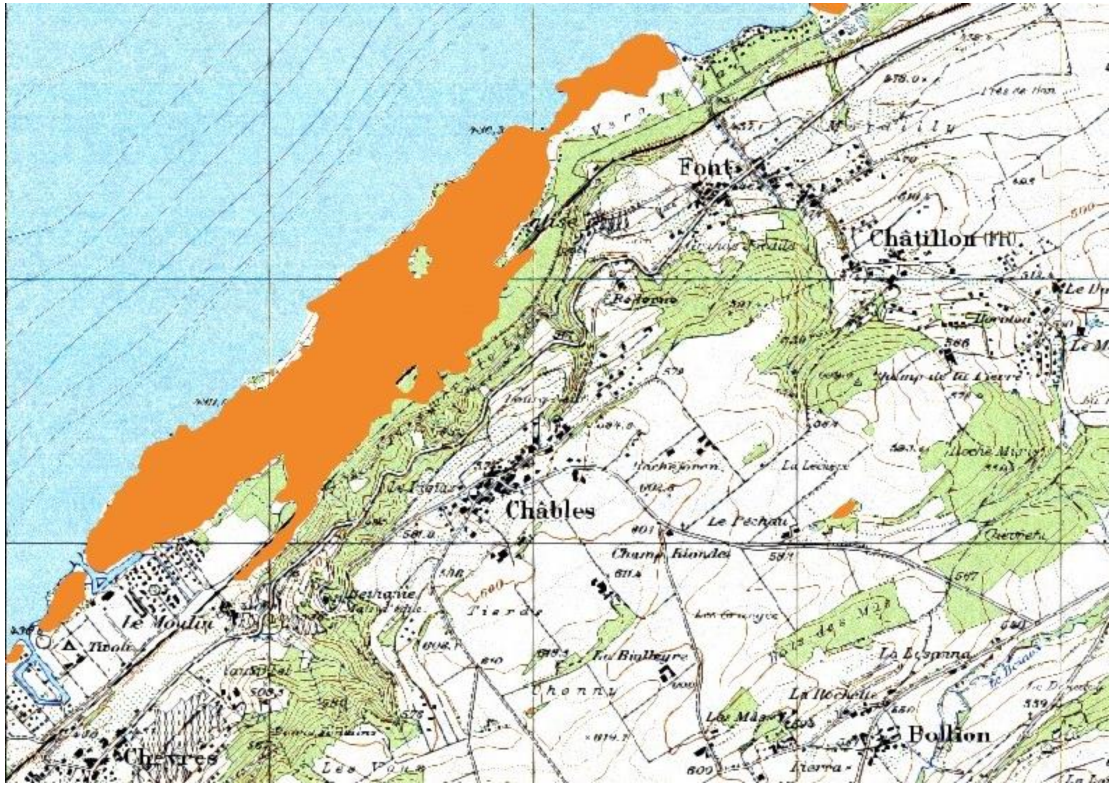
IoU	Precision	Recall	F1 Score
0.457	0.597	0.423	0.627

Ideal classification conditions

PSPNet Model



Ground Truth



From Training

IoU	Precision	Recall	F1 Score
0.629	0.850	0.754	0.799

From Pixel Classification

IoU	Precision	Recall	F1 Score
0.755	0.873	0.543	0.860

Workflows & Recommendation

- Not one perfect single solution
- Factors:
  - *Time Constraint*:
    U-Net clearly the quickest
    Set Max Epochs between 5 and 10
    Choose Backbone Model with less layers
  - *Performance Constraint:*
    Reduce Batch Sizes
    PSPNet and DeepLabV3 use less memory
    Choose Backbone Model with less layers
  - *Good Precision:*
    PSPNet with Mixup
  - *Good Recall*
    U-Net with Focal Loss

Exemplary Workflow for a variation of a PSPNet Model. *Own illustration*

General Parameters for Pixel Classification

Modeltype:	U-Net
Backbone:	Res-Net 34
Max Epochs:	20
Freeze Weights:	True

Outlook

Outlook

- How would it work for multiple classes?
- Application on a dataset with less discrepancies
- Use “bad” statistical metrics to document how wetlands are changing
- Statistical Analysis tool from ArcGIS?

Sources

[1] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. 2016

[3] S. Tsang. Review: DilatedNet — Dilated Convolution (Semantic Segmentation). 2019

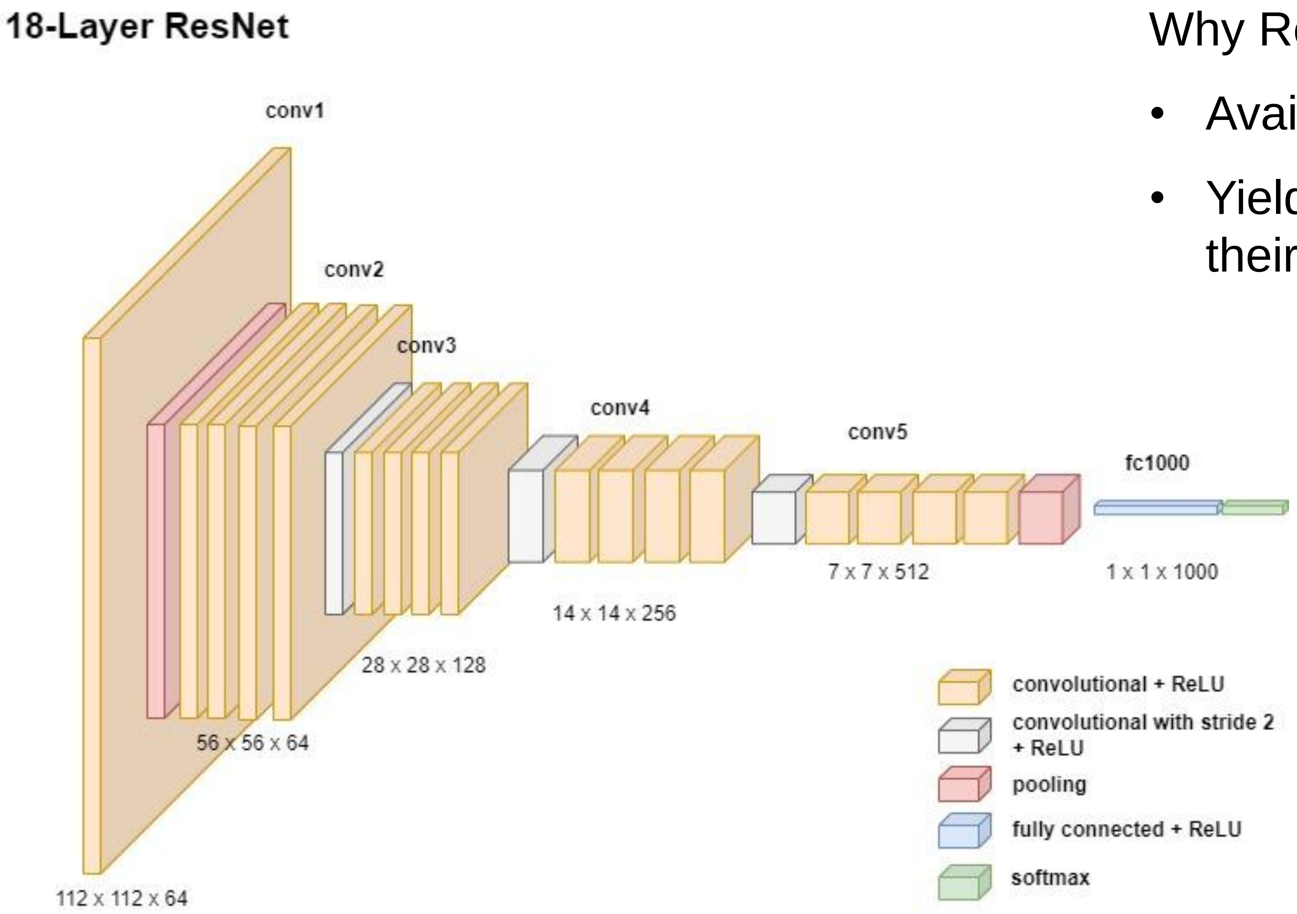
[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. 2017

[5] swisstopo. Swiss Map Raster 25 - Digitale Landeskarten der Schweiz im Rasterformat 1:25 000. Accessed on 29.May 2022 from Bundesamt für Landestopografie: *https://www.swisstopo.admin.ch/de/geodata/maps/smr/smr25.html*

[6] M. Stuber, M. Bürgi. Vom «eroberten Land» zum Renaturierungsprojekt. Geschichte der Feuchtgebiete in der Schweiz seit 1700. 2018

Additional Slides

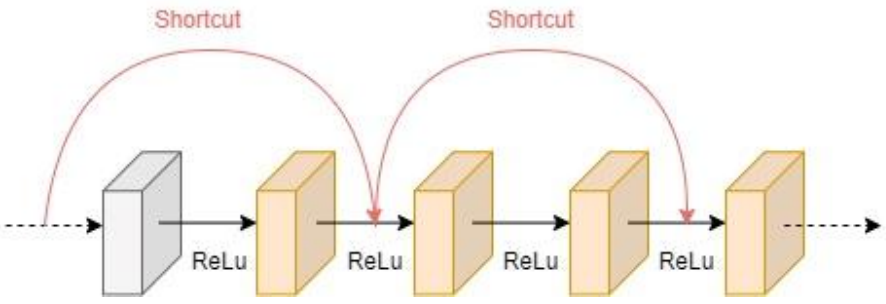
Hyperparameters:
Backbone Model ResNet



Architecture of a ResNet-18 Backbone Model. *Own illustration*

Why ResNet?

- Available for all considdered Models
- Yielded best metric for PSPNet and DeepLab in their respecitve research paper



Residual Learning Building Block. *Own illustration*

Discarded Object Classification Model

Feature Classifier

- Requires Labeled Tiles as training data
- Model Architecture dependant on choice of Backbone, e.g. ResNet 34

Suitability

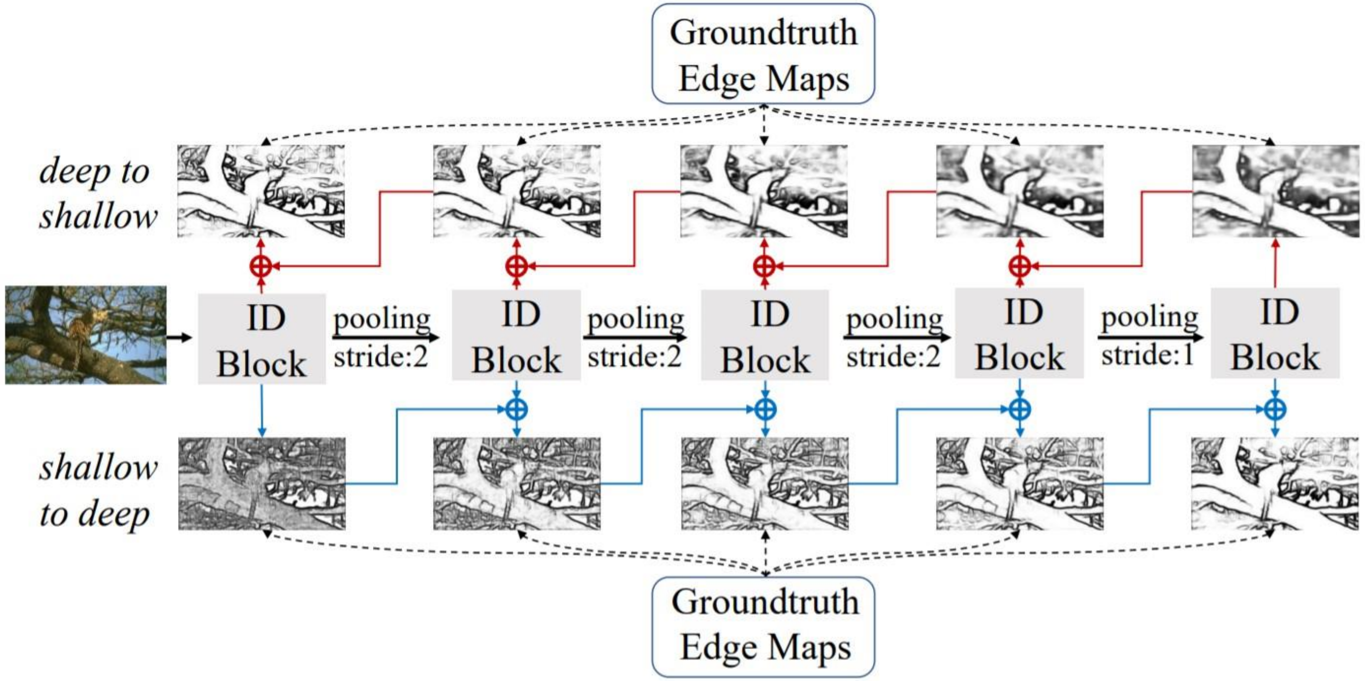
- Not applicable to our task at hand
- Needs established footprint in order to classify a feature

Exemplary Application of FeatureClassifier. Source: Esri

Discarded Pixel Classification Models

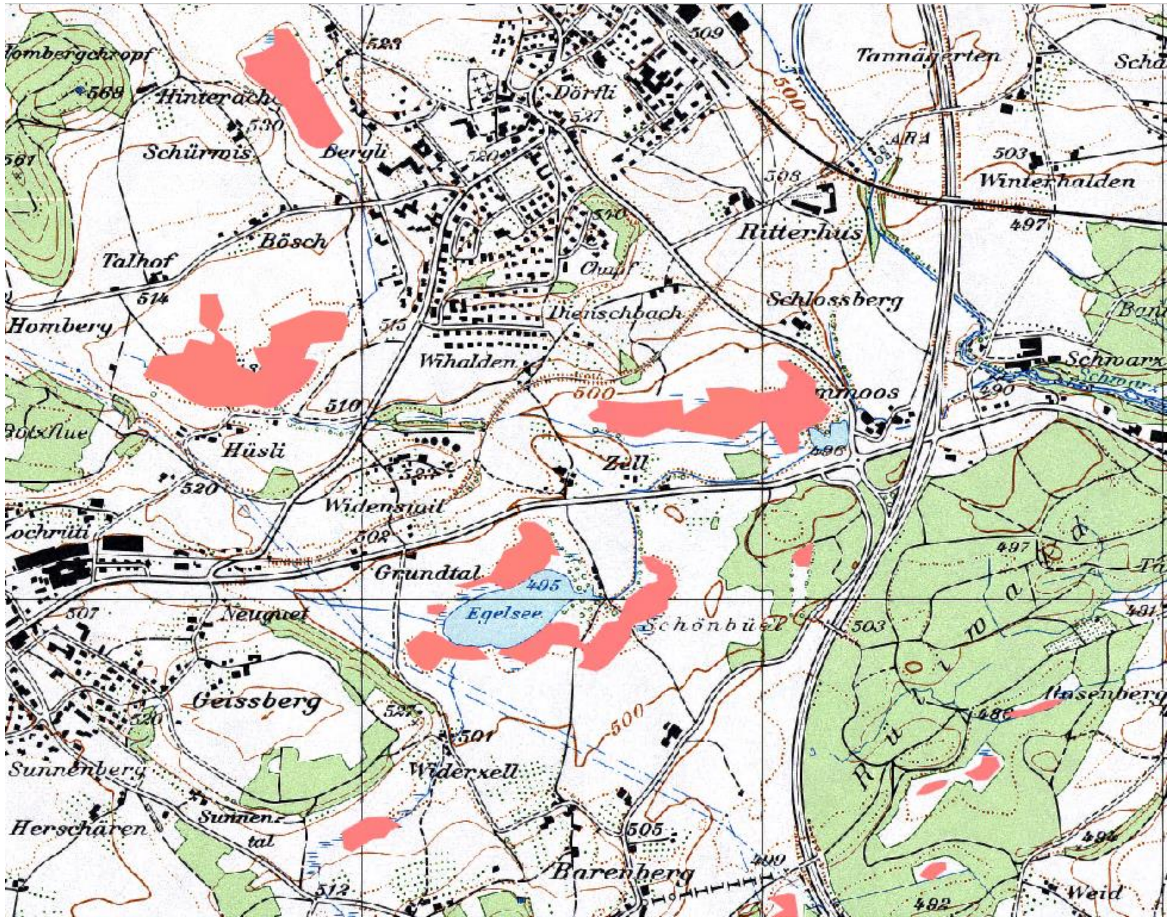
Bi-Directional Cascade Network for Edge Detector (BDCN)

- Focus on the edge detection of objects with varying sizes
  Sizes get adapted for each layer
- Depending on the depth in the model, other properties are highlighted
  Deep layers for outlines, top layers for details

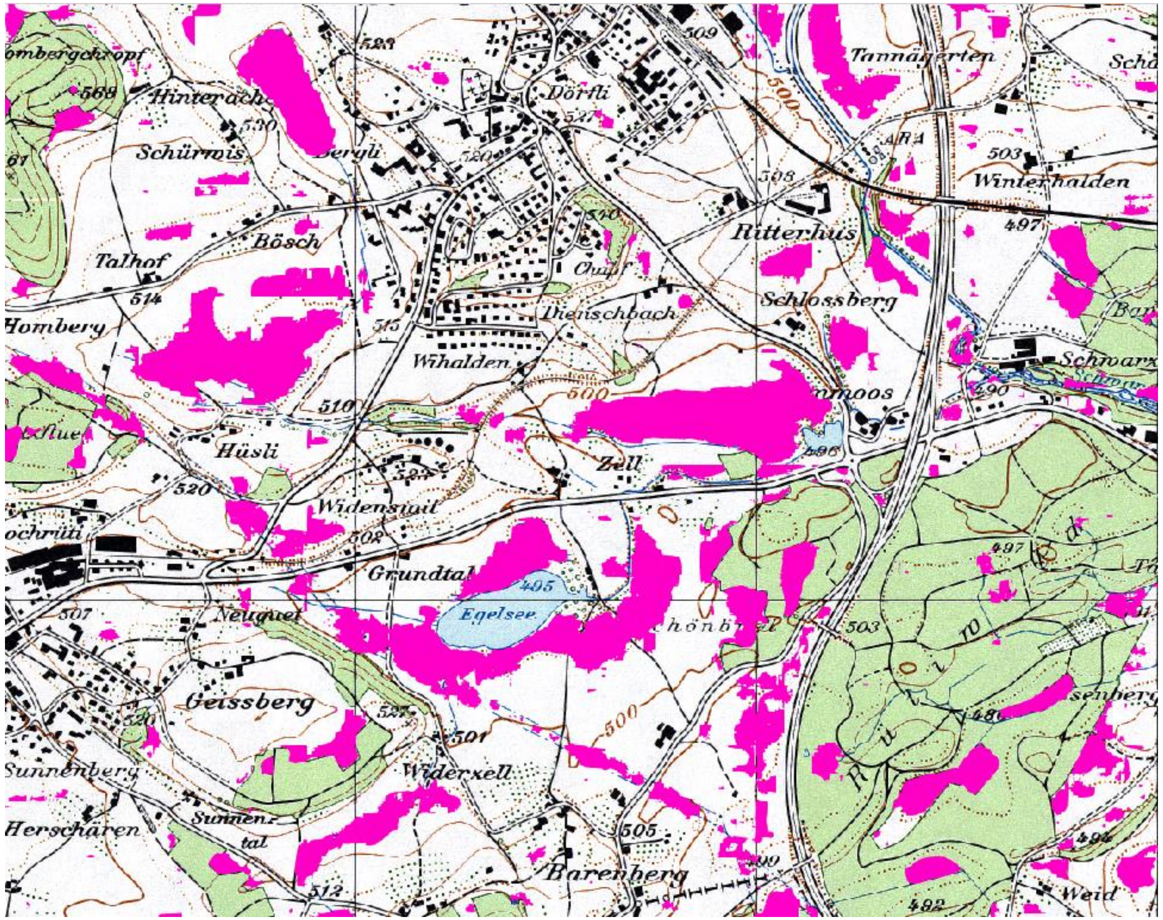


Architecture Bi-Directional Cascade Network for Edge Detector. Source: Esri

Bi-Directional Cascade Network for Edge Detector BDCN



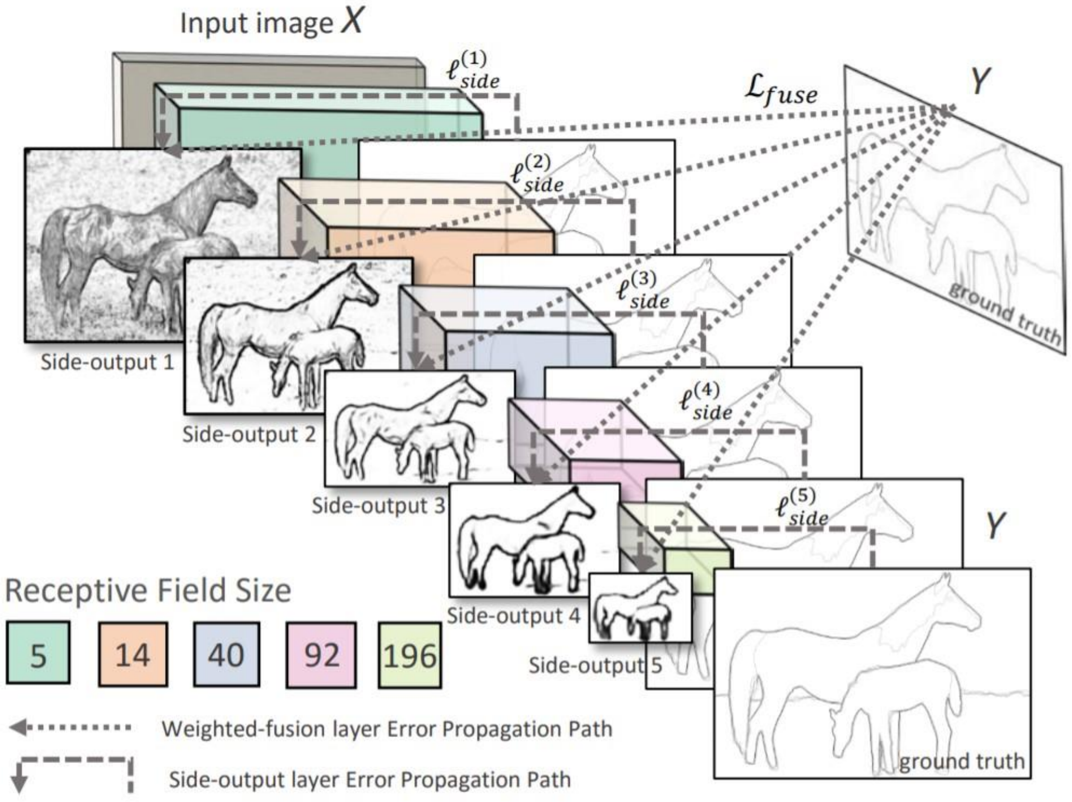
Ground truth



Trained BDCN Mode

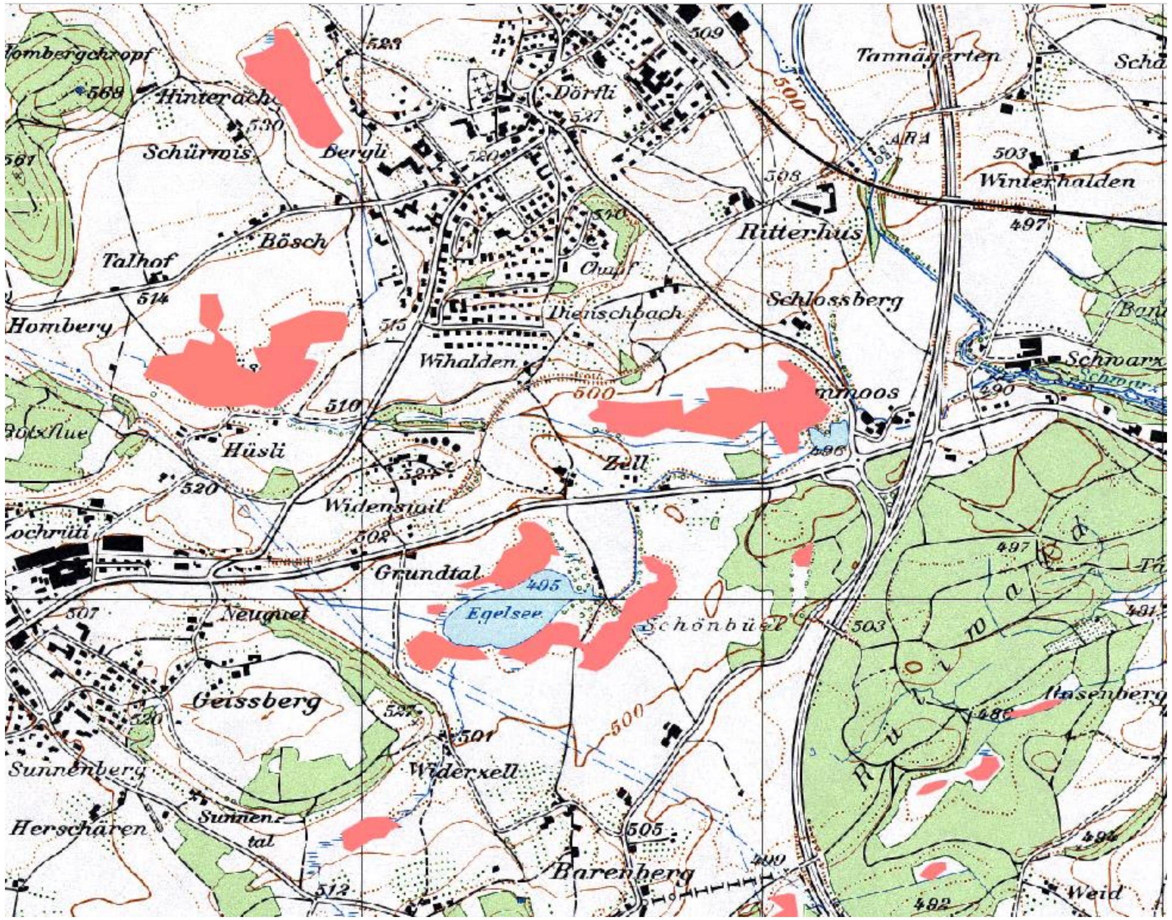
Holistically-Nested Edge Detector (HED)

- Precursor of BDCN
- Built on five convolution blocks

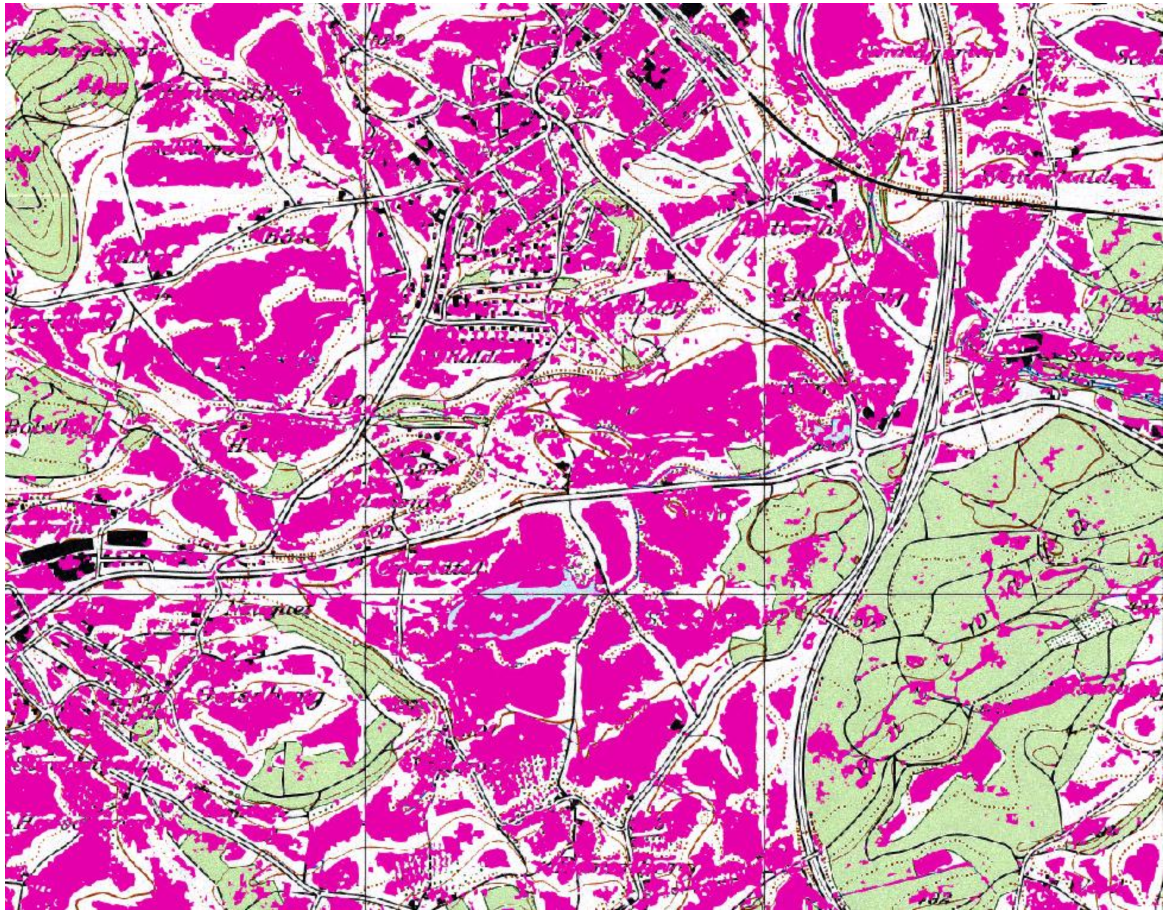


Architecture of Holistically-Nested Edge Detection. Source: Esri

Holistically-Nested Edge Detector (HED)



Ground truth



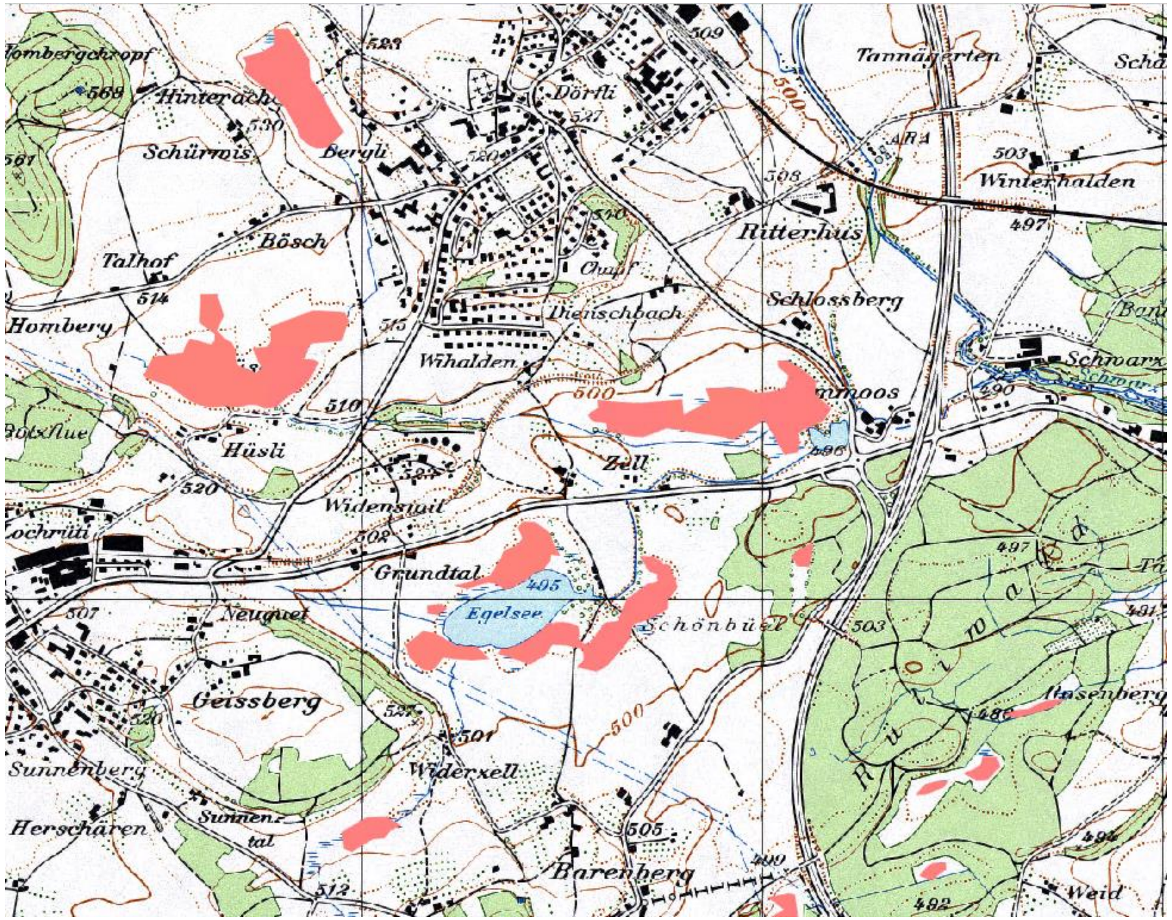
Trained HED Model

Multi Task Road Extractor & ConnectNet

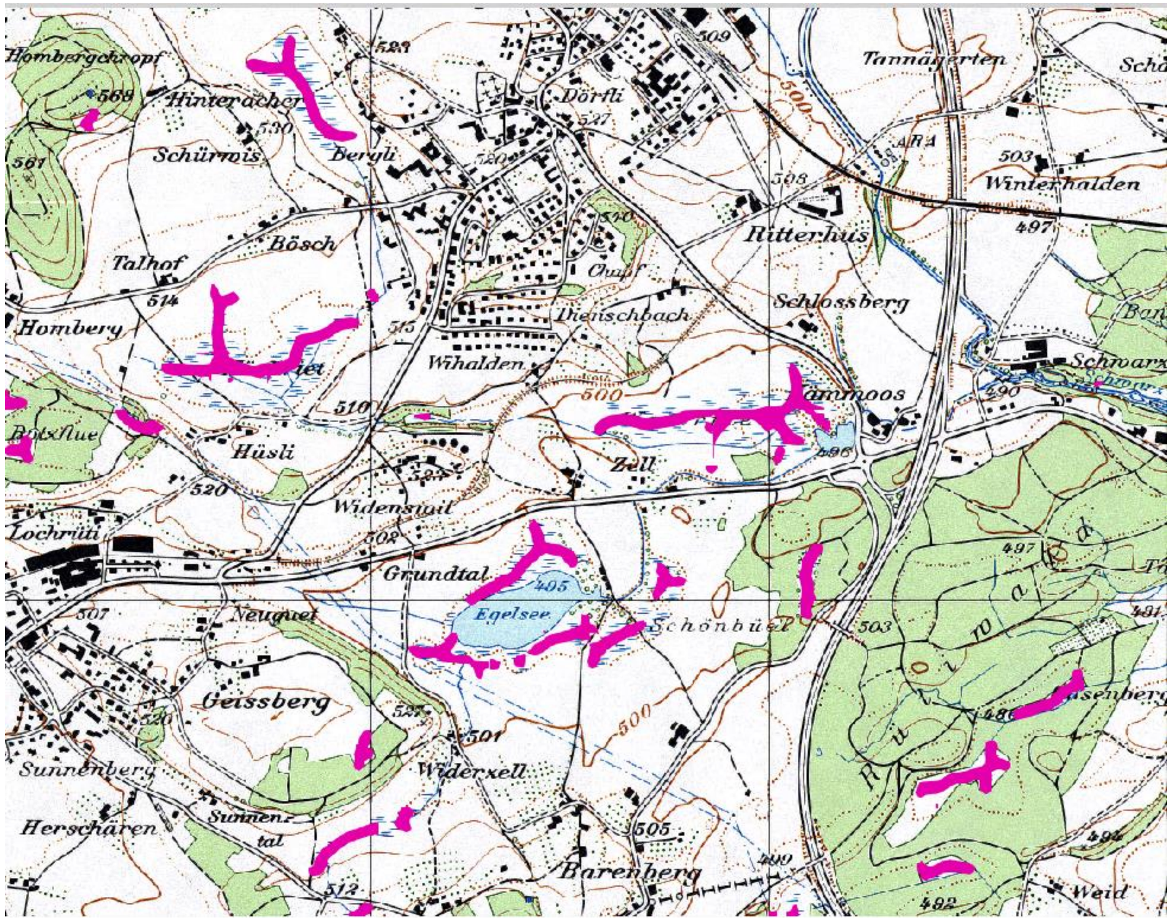
- Both architectures based on same published paper
- Focus of the models is to combine individual segments of a continuous line into a whole

Exemplary Application of Multi Task Road Extractor.
Source:Esri

Multi Task Road Extractor

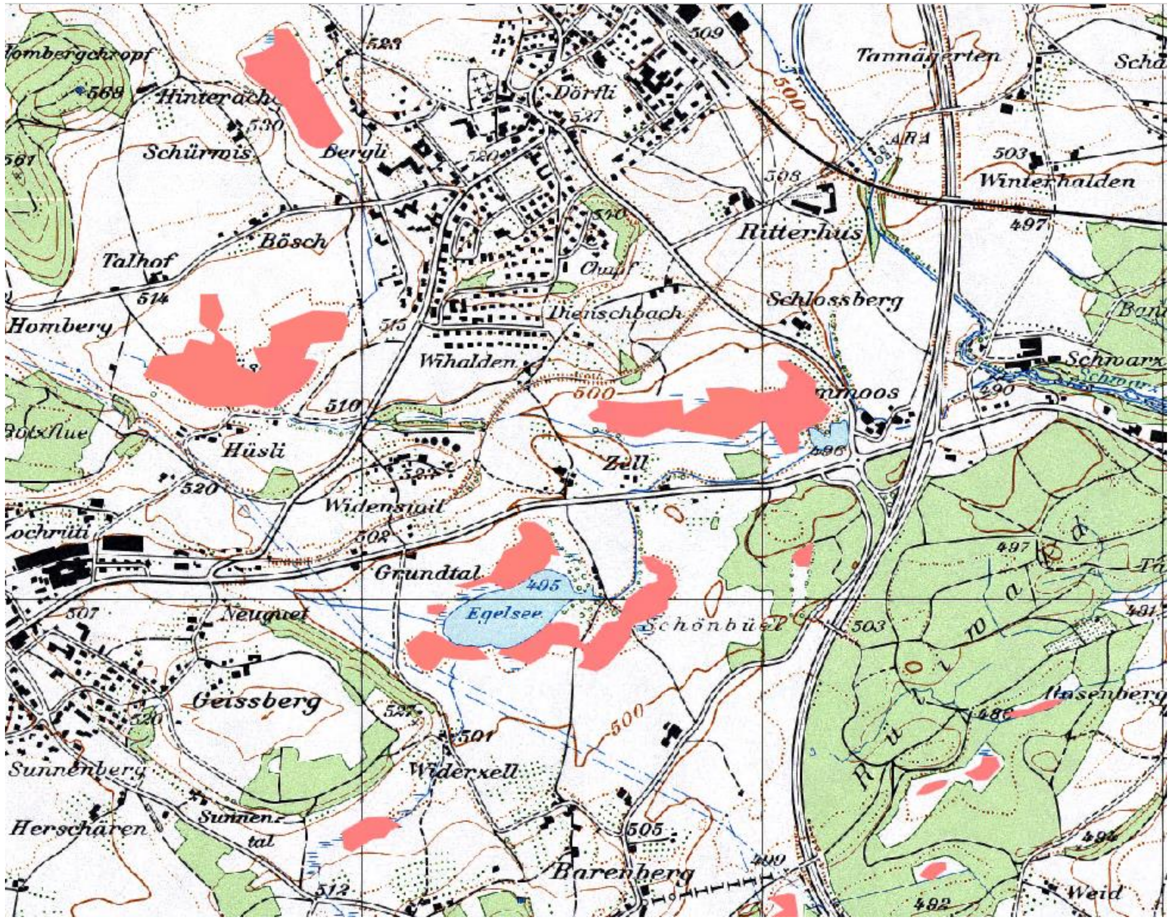


Ground truth

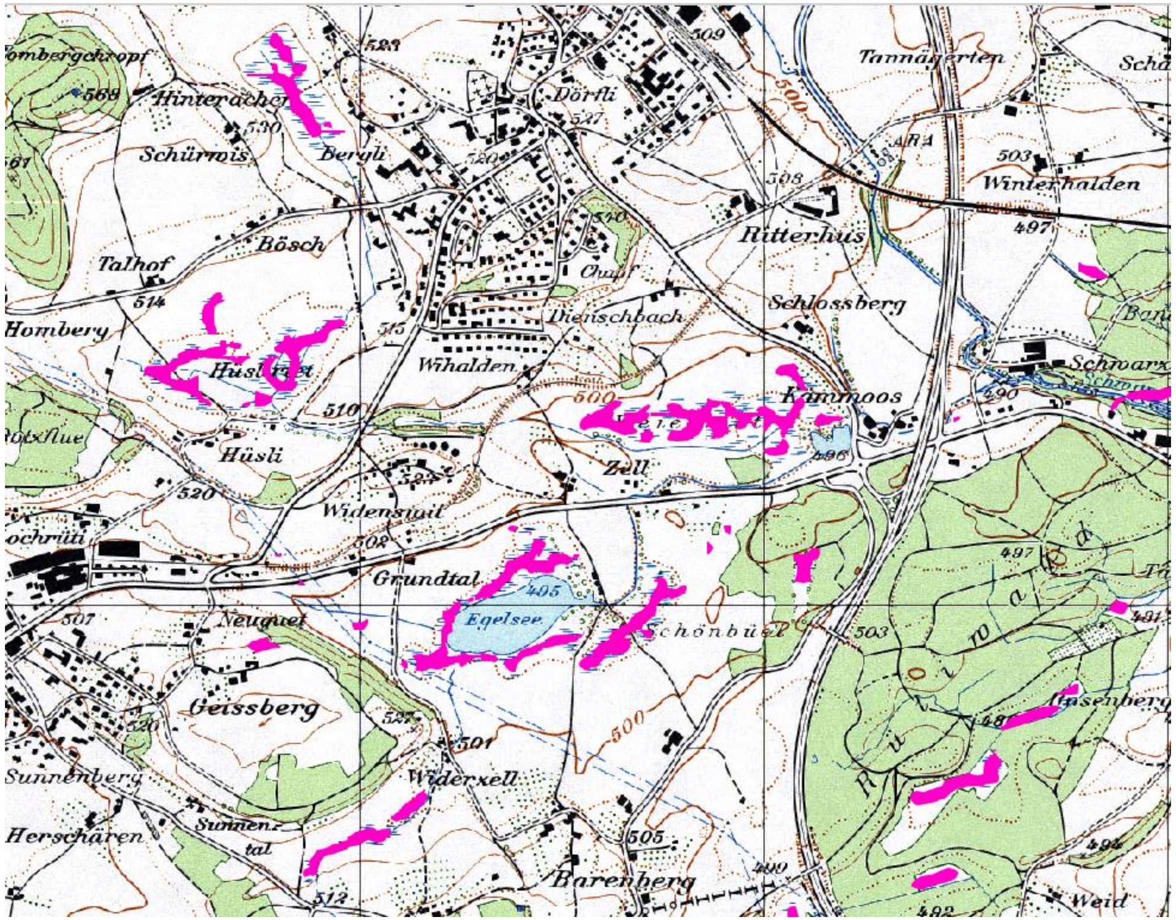


Trained Multi Task Road Extractor Model

ConnectNet



Ground truth wetland layer



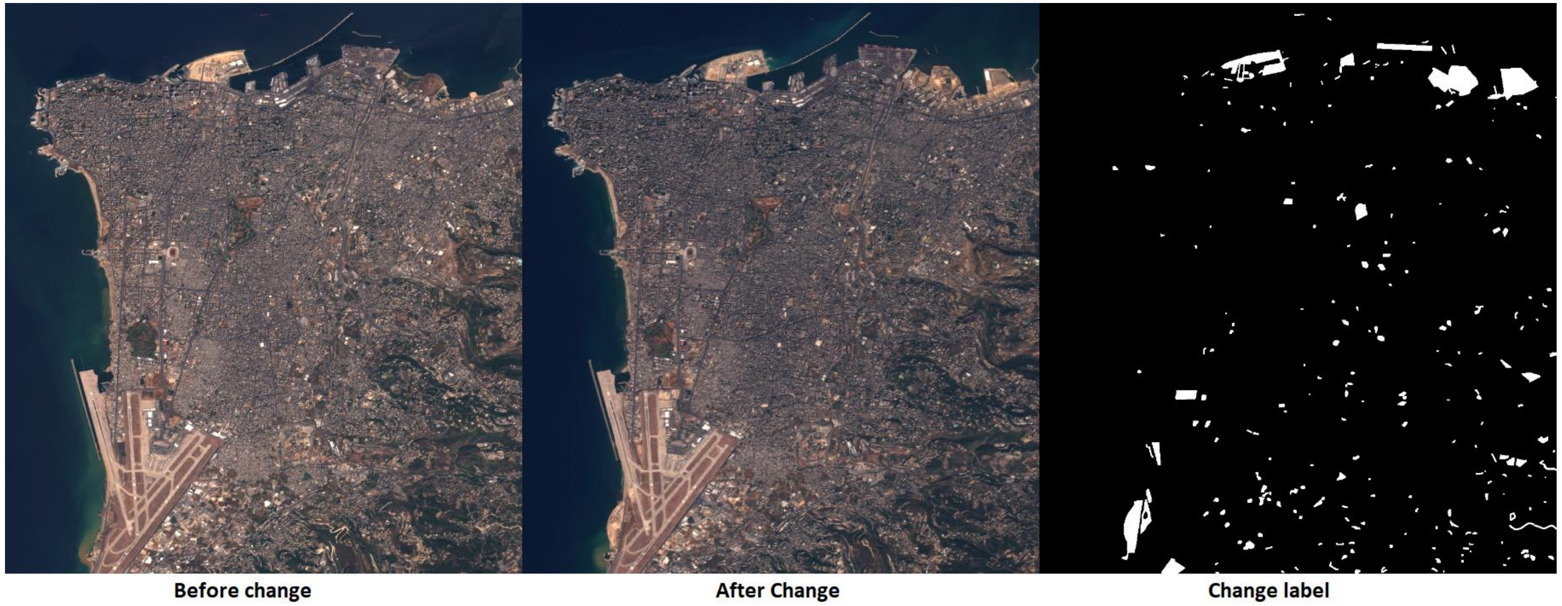
Model output

Change Detector

- Input are two images, depicting two different points in time
- Output entails whether the pixel has changed or not

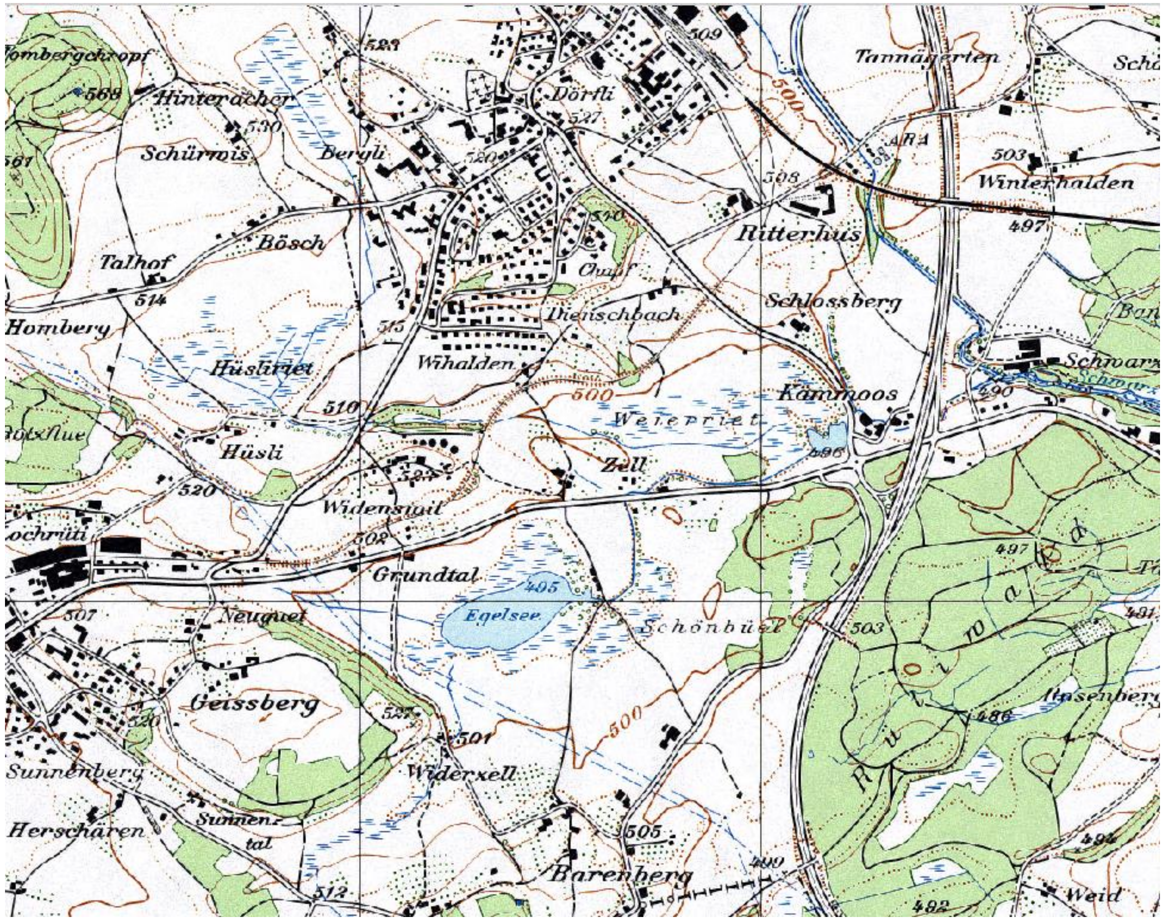
Suitability

- Not applicable to our Problem because there is no change to be detected

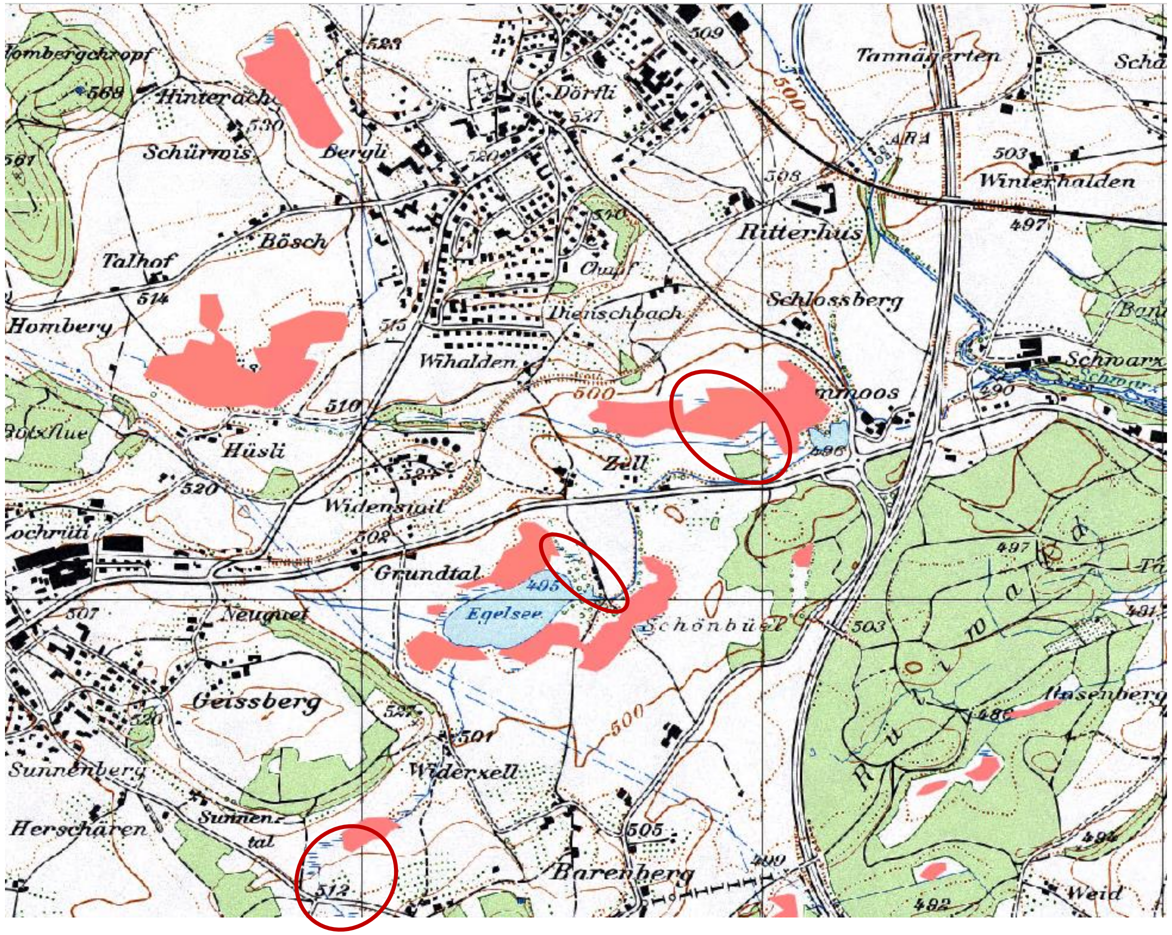


Exemplary Application of the Change Detector. Source: Esri

Data



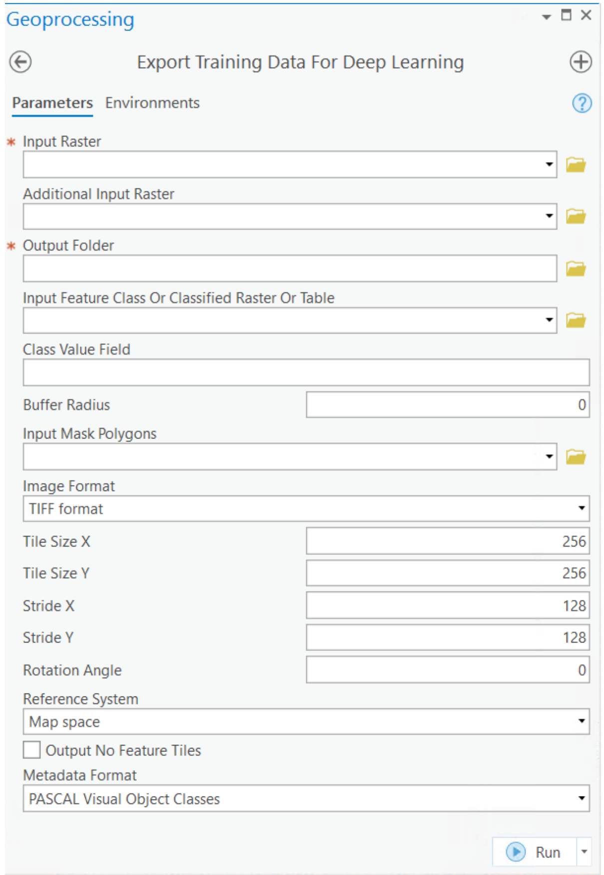
Excerpt of the National Map with various wetlands



Corresponding wetlands from our feature layer

Procedures in ArcGIS Pro

- Blackbox
- Tracing of functions is not possible
- Error Messages often times not useful




```python
# Import system modules and check out ArcGIS Image Analyst extension license
import arcpy
arcpy.CheckOutExtension("ImageAnalyst")
from arcpy.ia import *

# Set local variables
inRaster = "Y:/private/LKg_1112_1990.tif"
out_folder = "C:/Users/apironato/Documents/ArcGIS/Projects/Test_DL/ehmmm_folder"
in_training = "V:/Data_Bachelor_Thesis/Wetlands_Aline.gdb/ch_2010_map"
image_chip_format = "TIFF"
tile_size_x = "256"
tile_size_y = "256"
stride_x= "128"
stride_y= "128"
output_nofeature_tiles= "ALL_TILES"
metadata_format= "Classified_Tiles"
start_index = 0
classvalue_field = None
buffer_radius = 10
in_mask_polygons = None
rotation_angle = 0
reference_system = "MAP_SPACE"
processing_mode = "PROCESS_AS_MOSAICKED_IMAGE"
blacken_around_feature = "NO_BLACKEN"
crop_mode = "FIXED_SIZE"
```

```python
# Execute
ExportTrainingDataForDeepLearning(inRaster, out_folder, in_training,
    image_chip_format,tile_size_x, tile_size_y, stride_x,
    stride_y,output_nofeature_tiles, metadata_format, start_index,
    classvalue_field, buffer_radius, in_mask_polygons, rotation_angle,
    reference_system, processing_mode, blacken_around_feature, crop_mode)
```

Contrast between the ArcGIS User Interface and the equivalent Tool expressed as Python Code

Sources

- Esri. (2022a). *How feature classifier works?* Abgerufen am 6. April 2022 von ArcGIS API for Python: https://developers.arcgis.com/python/guide/how-feature-categorization-works/
- Esri. (2022d). *How U-net works?* Abgerufen am 5. April 2022 von ArcGIS API for Python: https://developers.arcgis.com/python/guide/how-unet-works/
- Esri. (2022e). *How PSPNet works?* Abgerufen am 5. April 2022 von ArcGIS API for Python: https://developers.arcgis.com/python/guide/how-pspnet-works/
- Esri. (2022f). *How DeepLabV3 Works*. Abgerufen am 7. April 2022 von ArcGIS API for Python: https://developers.arcgis.com/python/guide/how-deeplabv3-works/
- Esri. (2022g). *Edge Detection with arcgis.learn*. Abgerufen am 6. April 2022 von ArcGIS API for Python: https://developers.arcgis.com/python/guide/edge-detection-with-arcgis-learn/
- Esri. (2022h). *How Multi-Task Road Extractor works ?* Abgerufen am 7. April 2022 von ArcGIS API for Python: https://developers.arcgis.com/python/guide/how-multi-task-road-extractor-works/
- Esri. (2022i). *How ChangeDetection Works?* Abgerufen am 7. April 2022 von ArcGIS API for Python: https://developers.arcgis.com/python/guide/how-change-detection-works/
- He, J., Zhang, S., Yang, M., Shan, Y., & Huang, T. (2019). *Bi-Directional Cascade Network for Perceptual Edge Detection.* Peking University, Computer Vision and Pattern Recognition, Peking. doi:10.1109/TPAMI.2020.3007074
- Hurni, L., Heitzler, M., Jiao, C., & Xia, X. (2021). Raumbezogene Ingenieurwissenschaften – Bachelor-Arbeiten FS 2022. Zürich.