ETH zürich

Building block vectorization of Atlas Municipal

Chunyang Gao, Emanuel Stüdeli

Research Topics in Cartography

28.05.2024

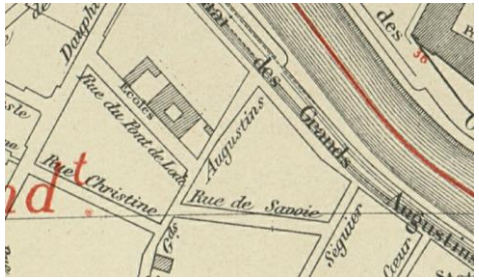
Content

1. Recap of project
2. Pipeline
3. Models
   a) U-Net
   b) ResUnet
   c) SwinUnet
4. Result
5. Challenges
6. Conclusion and future directions
7. Workload distribution

Recap

Problem Description

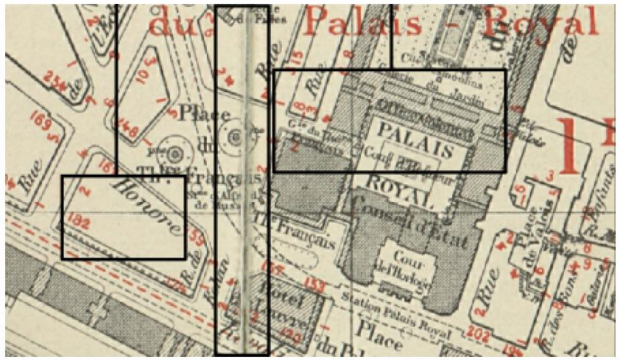
- Building blocks vectorization of Atlas Municipal, a series of city maps of Paris in the late 18th century
- Challenges:
  - Lack of color and texture information
  - Nested objects, Building blocks are highly overlapped with objects and text
  - Scanned Maps include skews, warps and different contrast



Excerpt of Input Image

Excerpt of Expected Output (in orange)

Scans with different contrast



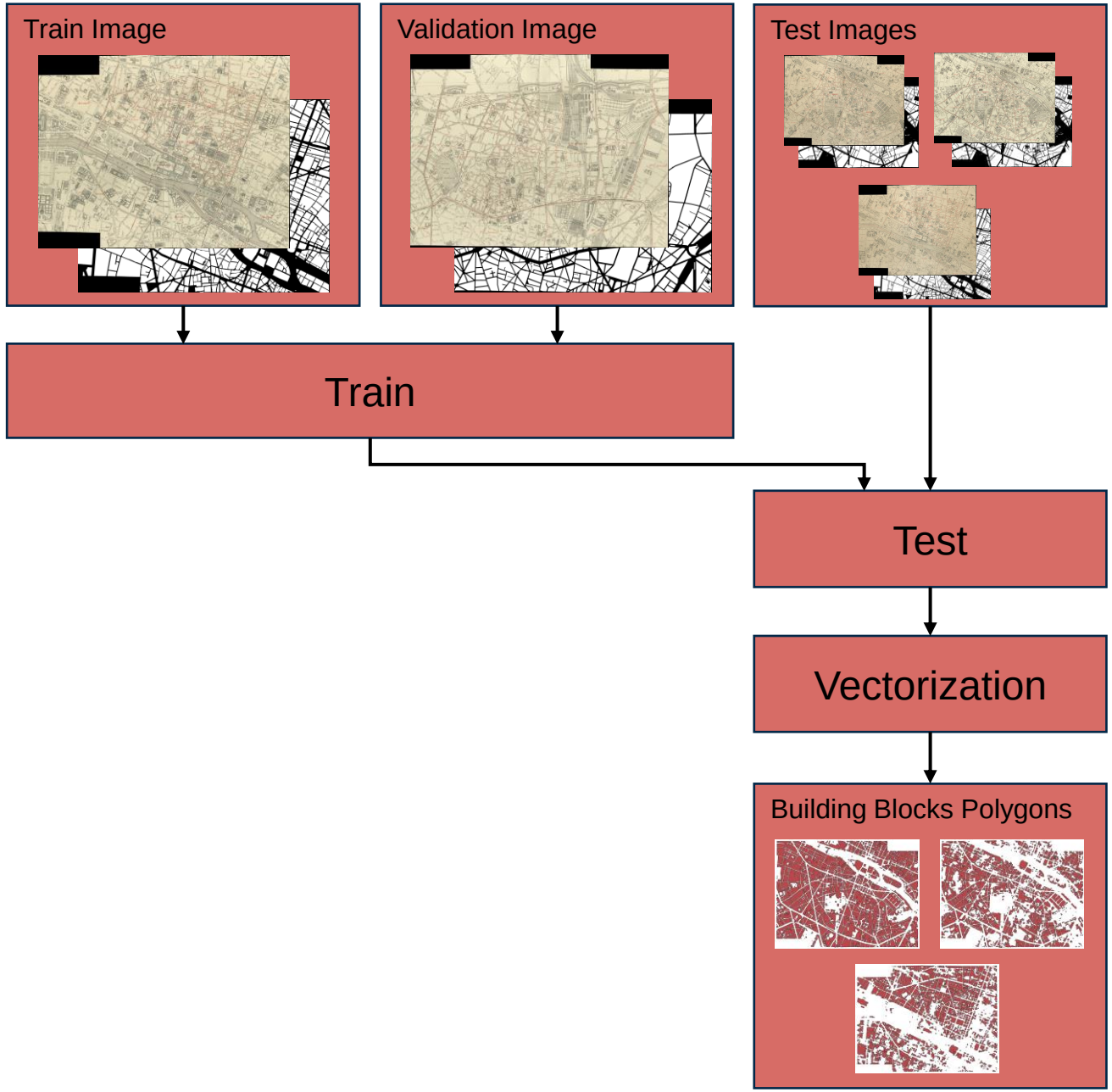
Zoom in with potential challenges marked in black

Recap

Tasks

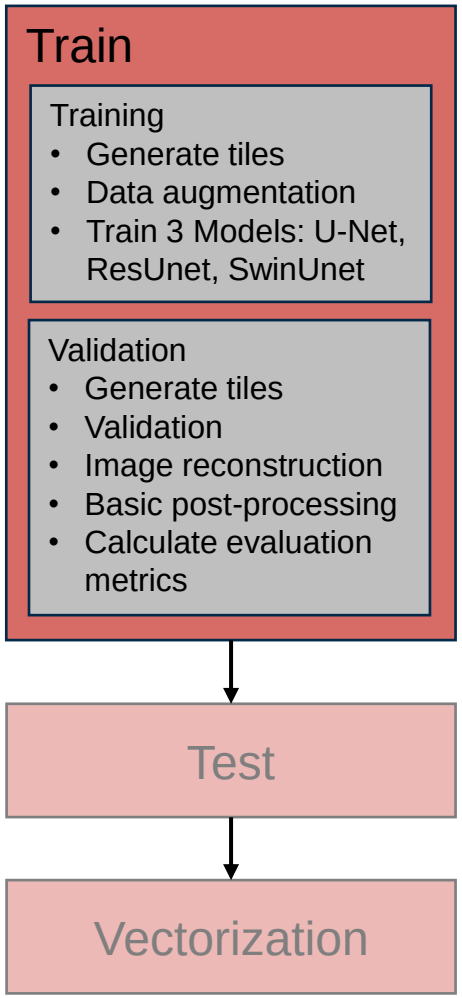
1. Find out feasible pipelines for vectorizing building blocks
2. Investigate three different model types. Evaluate the model performance with pixel- and instance-based metrics
3. Improve and optimize your pipeline and the models

Pipeline


Train Image
Validation Image
Test Images
Train
Test
Vectorization
Building Blocks Polygons

Pipeline

Train



- Three different Models: U-Net, ResUnet and SwinUnet

Training properties

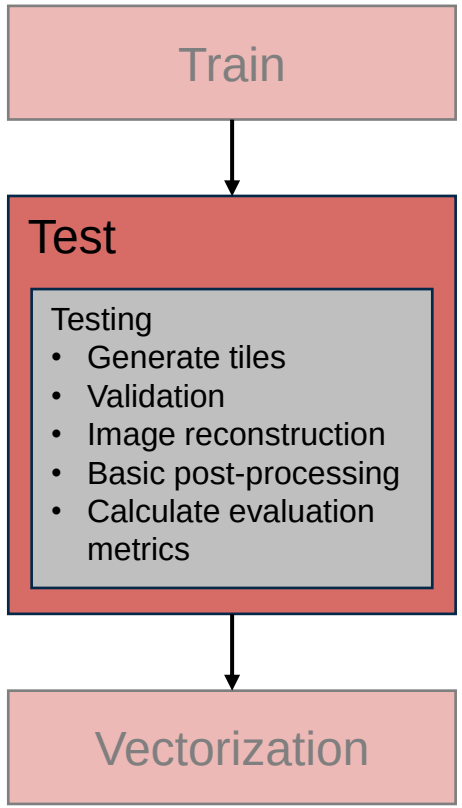
- Train loss: dice + BCE
- Regularization and gradient clipping to reduce overfitting
- Data augmentation to address various scan conditions (skews, warps, contrast)
  - Random contrast
  - Random rotation
  - Random thin plate spline

Validation properties

- Mask to mask out irrelevant areas
- Includes basic postprocessing steps (opening, closing, connected components)
- Pixel and instance-based evaluation metrics

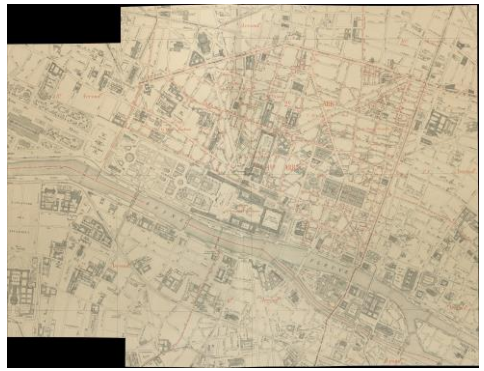
Pipeline

Test



Three test images for evaluation with different background shades, contrast and other scan properties

- File 303 is most similar to the train image regarding this aspects



File 301

File 302

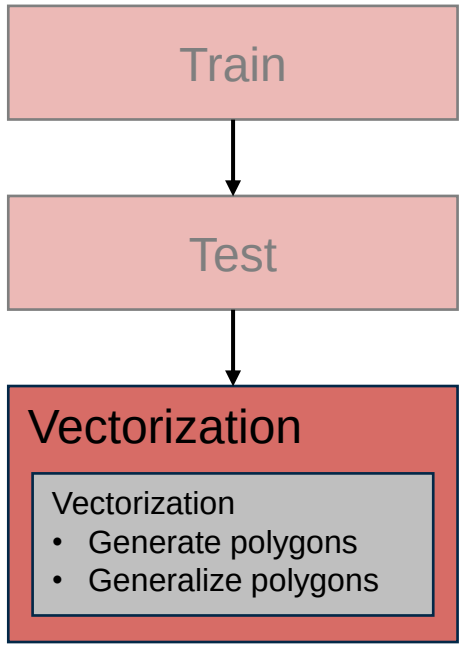
File 303

Testing properties

- Mask to mask out irrelevant areas
- Includes basic postprocessing steps (opening, closing, connected components)
- Pixel and instance-based evaluation metrics

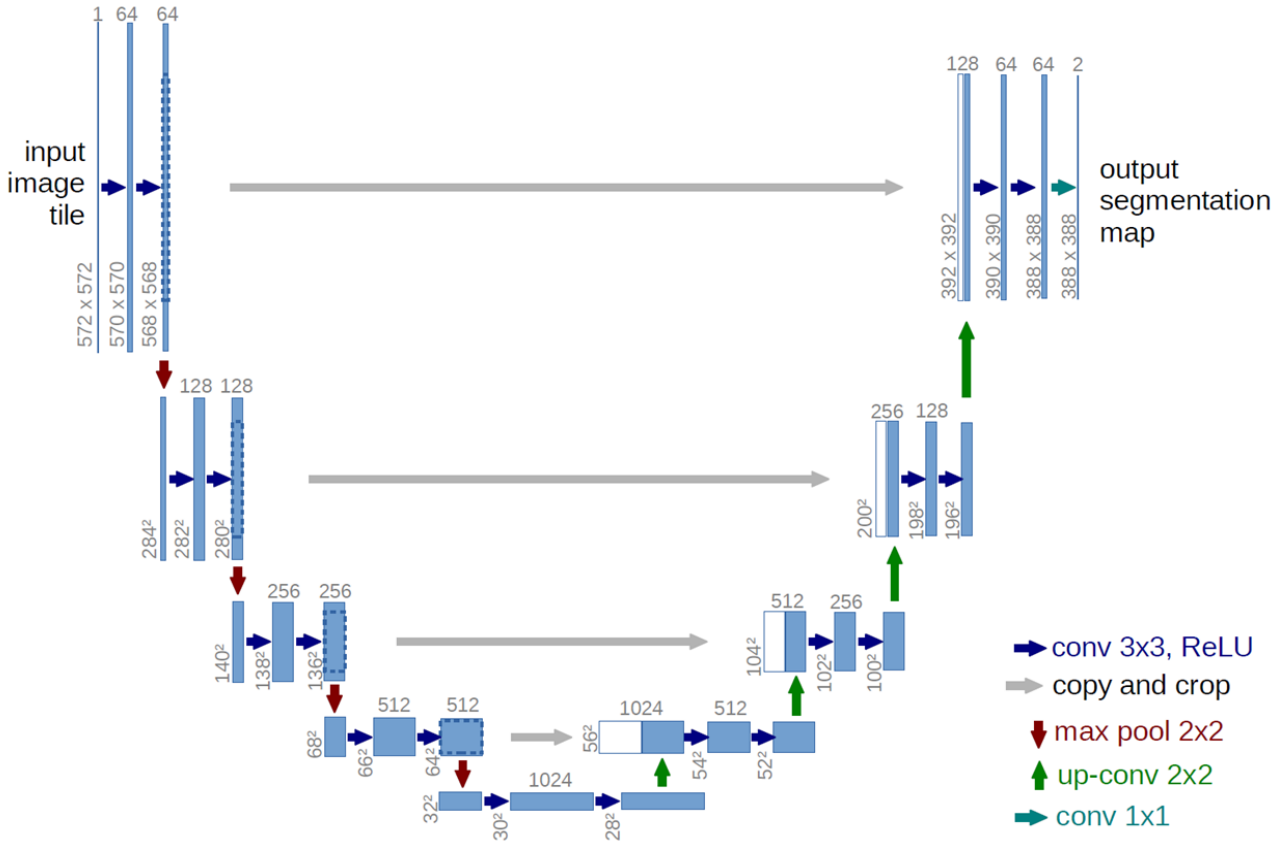
Pipeline

Vectorization



- Vectorization of building blocks
- Generalization with Douglas-Peucker

U-Net



U-Net architecture proposed by Ronneberger et al. [1]

U-Net our implementation:

- Image tile size 512
- Kernel size 5
- Input channels 3, Output class 1

ResUnet

Related Work: Road Extraction by Deep Residual U-Net [2]

- ResUnet for road extraction from high resolution remote sensing images
- Combines the strengths of residual learning and U-Net

Input Image

Ground Truth

ResUnet prediction

ResUnet

Related Work: Road Extraction by Deep Residual U-Net [2]

Architecture

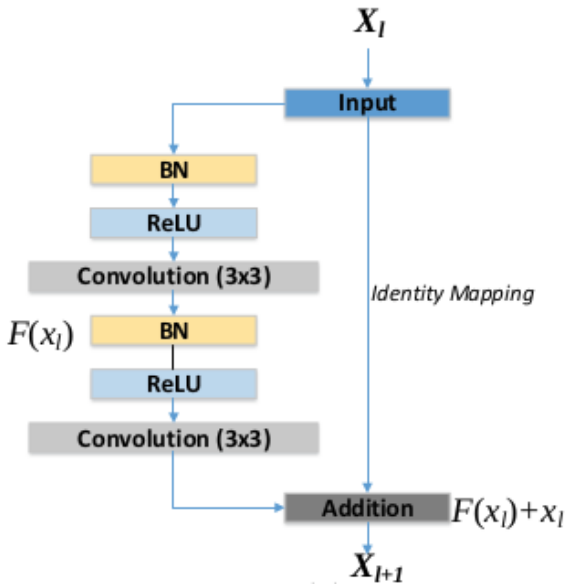
Encoding

- Residual units as basic blocks, which consits of:
  - Two 3x3 convolutional block
  - An identity mapping
- Stride 2 is applied for downsampling insead of pooling layer

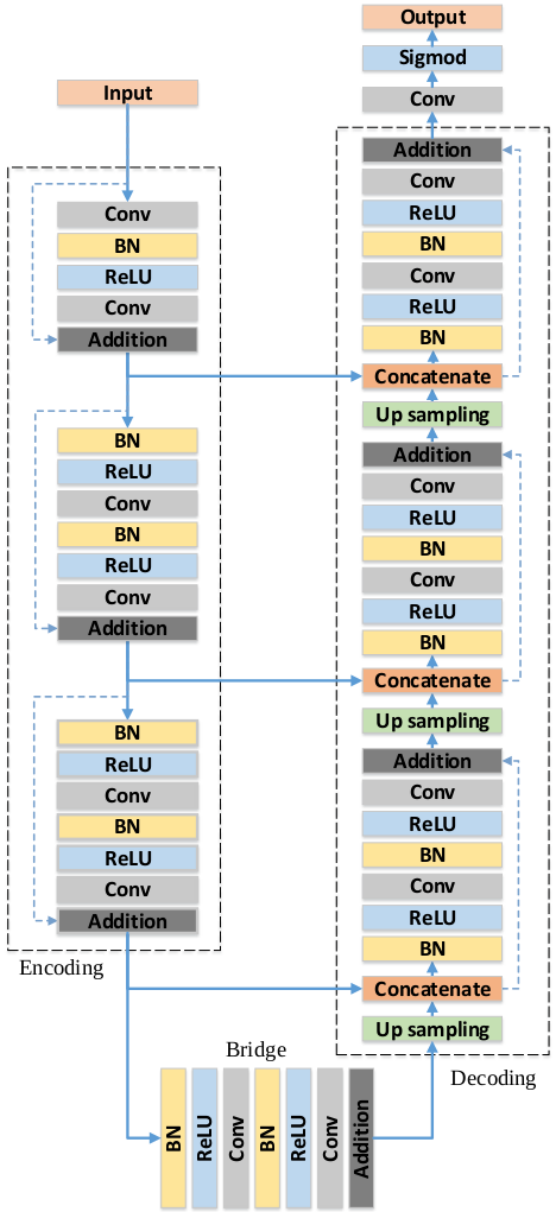
Bridge

Decoding

- Up-sampling
- Concatenation with the feature maps from the corresponding encoding path (skip connections)



Residual Unit

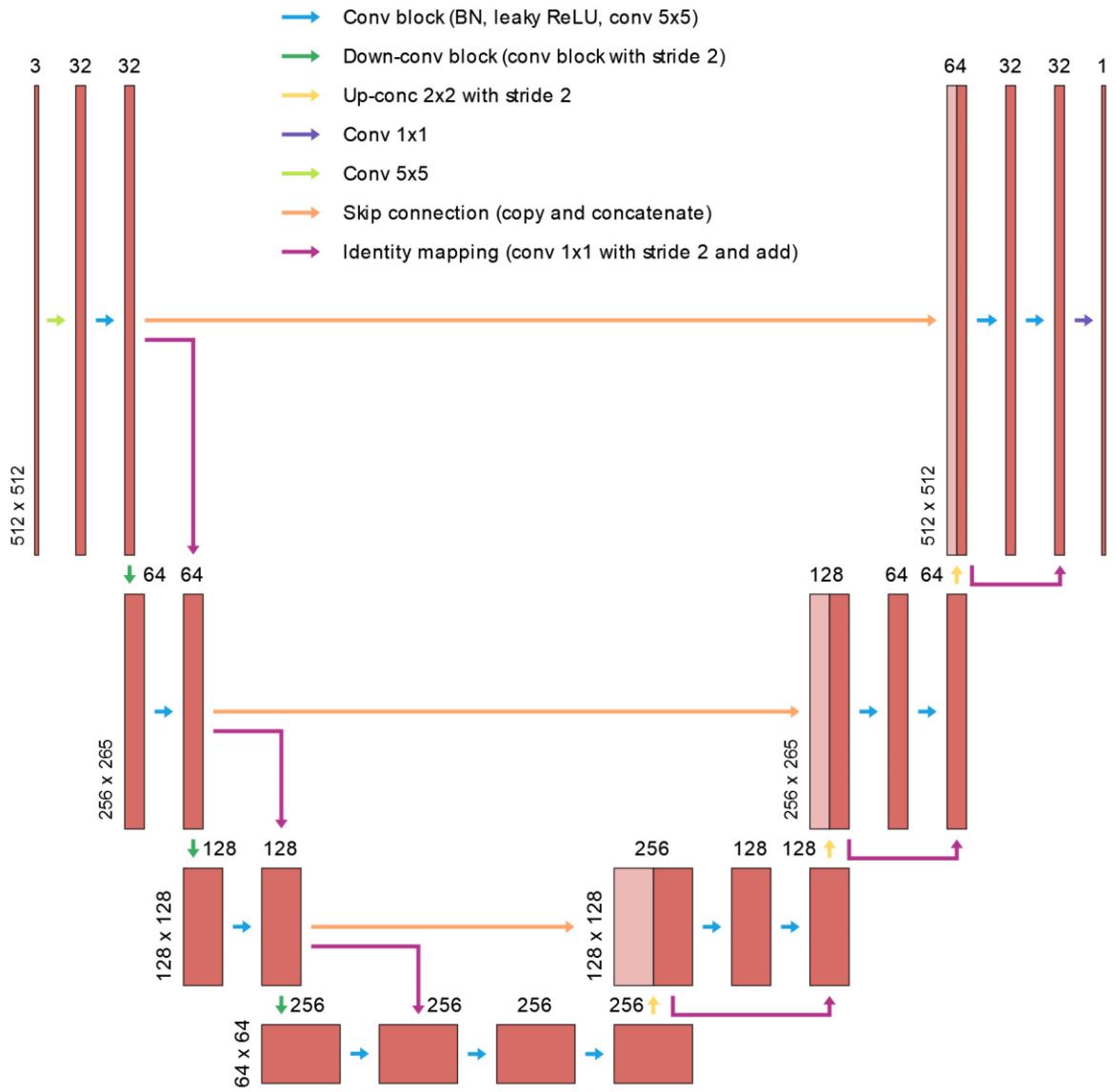


Complete Architecture

ResUnet

Our Implementation

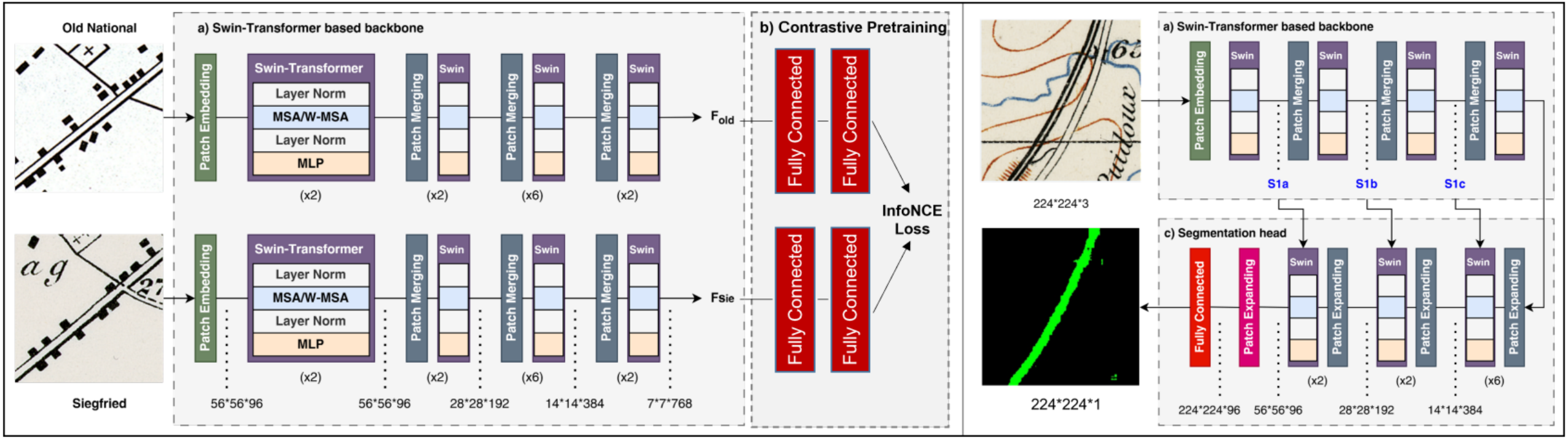
- Proposed model was trained with various adaptions of the model architecture
- Best model performance with
  - Seven Levels with two convolution blocks in each level
  - Kernel size 5
  - 32 channels for first convolution
  - Activation layer leaky ReLU instead of ReLU



ResUnet architecture with best performance

SwinUnet

Related Work: Contrastive Pretraining for Railway Detection: Unveiling Historical Maps with Transformers [3]

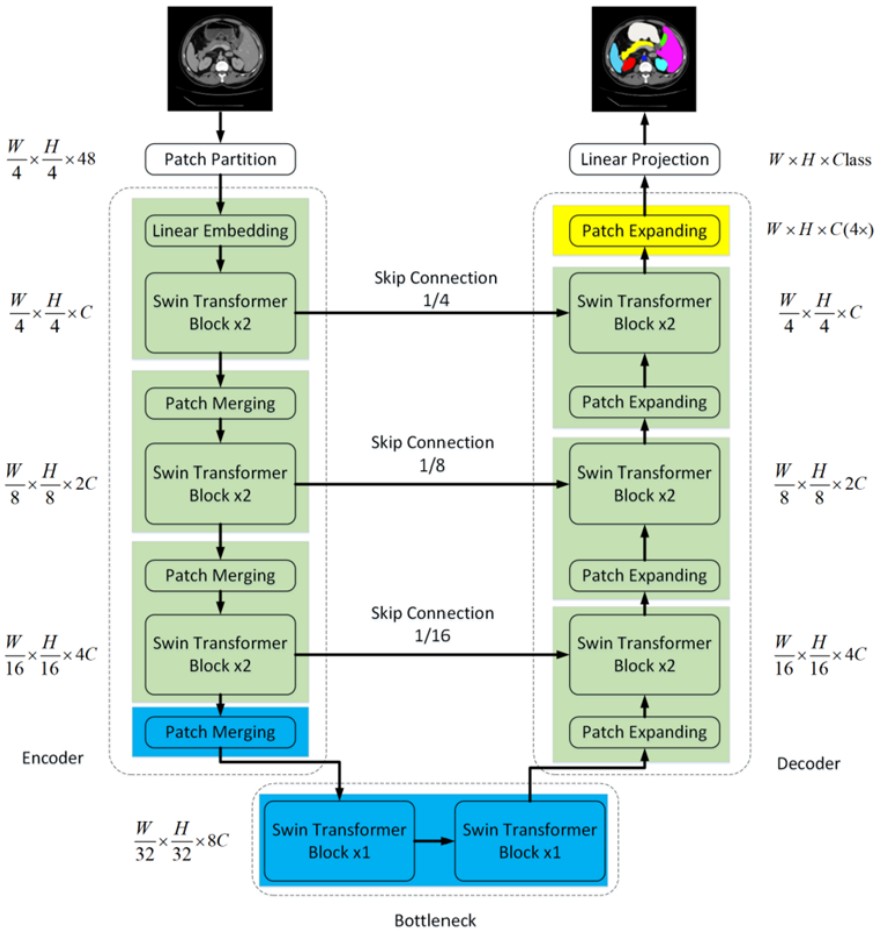


The overall approach. a shows the architecture of the Transformer encoder; c is the segmentation head used in the downstream railway detection task. The training contains two stages. a + c forms the model architecture for the downstream task, where the pretrained encoder for the Siegfried map along with the segmentation head are combined and fine-tuned to perform railway detection.

- Finetune the entire transformer network for the downstream railway detection task. Experimental results demonstrate that the method achieves comparable performance to fully supervised approaches.

SwinUnet

Our Implementation



The architecture of Swin-Unet, which is composed of encoder, bottleneck, decoder and skip connections. Encoder, bottleneck and decoder are all constructed based on swin transformer block.

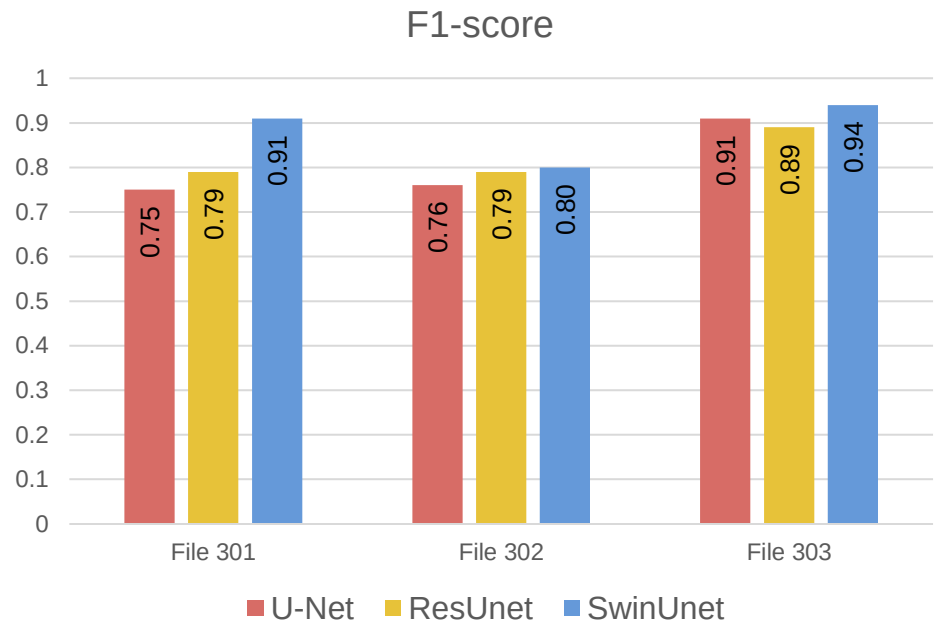
- SwinUNet adapts the U-Net design by replacing traditional convolutional layers with Swin Transformer blocks. Swin Transformers are based on self-attention mechanisms that address long-range dependencies within images.
- SwinUNet structure with pre-trained weights from the original paper was trained on our dataset
- Model setting
  - Image size 224 × 224 (same as original paper)
  - Class 1

Result

Pixel-based evaluation metrics

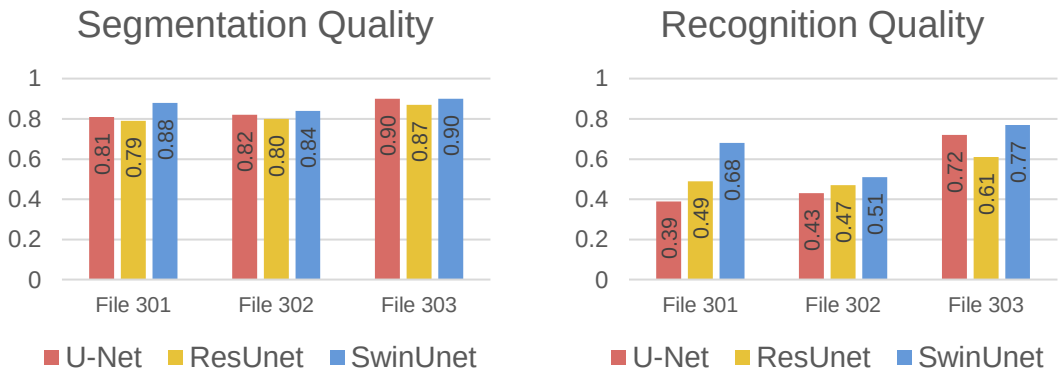
Evaluation with F1 score

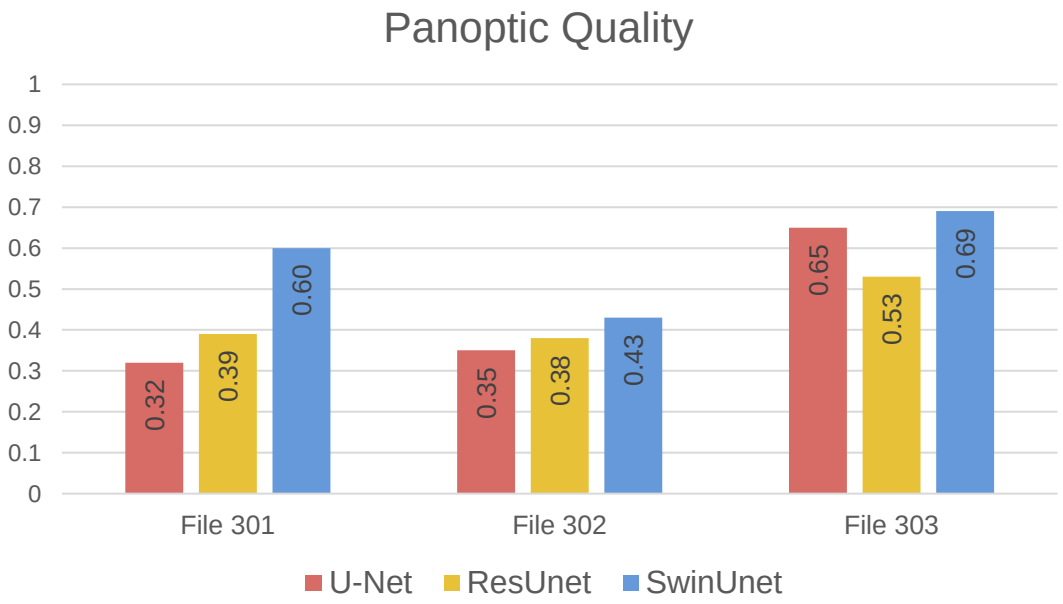
- All models achieve the highest performance on file 303
  - File 303 has similar contrast and background shade as train image
- SwinUnet generally achieves best F1 score



Result

Instance-based evaluation metrics (Coco Panoptic Quality score)





Segmentation quality (SQ): mean IoU between matching shapes (matching shapes in reference and prediction have an IoU > 0.5). SQ∈[0,1], higher is better.
Recognition quality (RQ): detection F-score for shapes, a predicted shape is a true positive if it as an IoU > 0.5 with a reference shape.
Panoptic Quality (PQ): PQ=SQ∗RQ. PQ∈[0,1], higher is better.

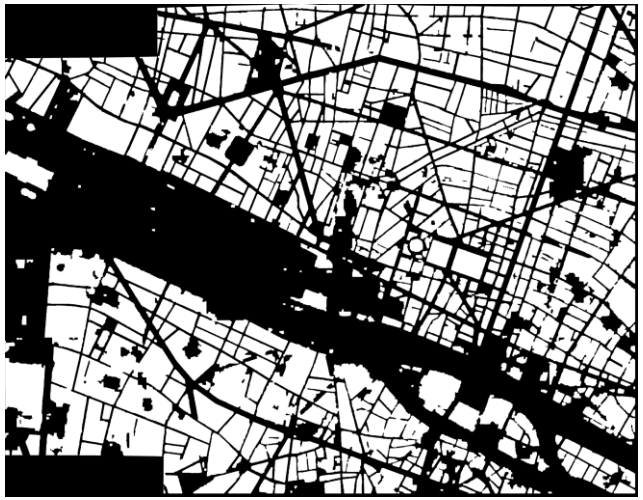
- Segmentation Quality is high for all models on all files (between 0.79 and 0.90)
- Recognition Quality is limiting factor and varies from 0.39 and 0.77
- All models achieve the highest performance on file 303
- SwinUnet generally performs best in instance-based evaluation metrics

Result

Example: SwinUnet prediction and vectorization for file 301

Full image

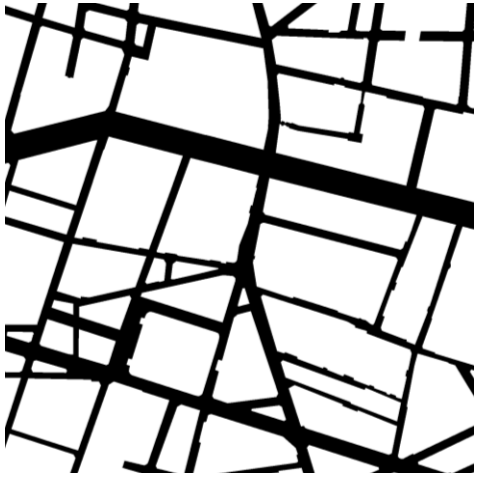
True labels



Prediction

Vectorized building block polygons

Zoom in



Challenges

- Capability of model to generalize
  - ➢ Implement Data Augmentation
  - ➢ Use pretrained model

- Find model with stable learning rate
  - ➢ Implement and optimize regularization method and gradient clipping

- Time and memory constraints
  - ➢ Choose appropriate batch size to fit the memory
  - ➢ Team organization

Conclusion and future directions

Conclusion

- Our pipeline is feasible for vectorizing building blocks.
- SwinUnet performs best for both pixel- and instance-based metrics, followed by ResUnet and U-Net.
- Since input data is limited, data aug. helps to achieve better performance and improves the generalization abilities of the model.
- For UNet and ResUNet, larger kernel size can help increase the model performance a little
- Limitations:
  - SwinUnet is the only model which uses pretrained weights, limited comparability with other models
  - Not all model parameters and design choices for the model architecture are evaluated

Future directions

- Integrate additional data sources for training
- Use pretrained weigths as well for U-Net and ResUnet
- Adapt model for other segmentation tasks for the Atlas Municipal

Workload distribution

	Emanuel Stüdeli	**Chunyang Gao**
Implement pipeline	✓	(✓)
U-Net implementation	(✓)	✓
ResUnet implementation	✓	
SwinUnet implementation		✓
Post processing and evaluation metrics	✓	
Vectorization	✓	

Reference

[1]
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv, abs/1505.04597.

[2]
Zhang, Z., Liu, Q., & Wang, Y. (2017). Road Extraction by Deep Residual U-Net. IEEE Geoscience and Remote Sensing Letters, 15, 749-753.

[3]
Xue Xia, Chenjing Jiao, and Lorenz Hurni. (2023). Contrastive Pretraining for Railway Detection: Unveiling Historical Maps with Transformers. In Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery (GeoAI '23). Association for Computing Machinery, New York, NY, USA, 30–33. https://doi.org/10.1145/3615886.3627738

ETH zürich

Questions