# Random Forest and Deep Learning for Glacier Mapping

## Finse Alpine Research Centre, Norway


Konstantin A. Maslov
k.a.maslov@utwente.nl

University of Twente,
Faculty of Geo-Information
Science and Earth Observation

Thomas Schellenberger
thomas.schellenberger@geo.uio.no

University of Oslo,
Faculty of Mathematics and
Natural Sciences


April 2023

# Outline

Image classification

Random forest

From convolutional networks to fully-convolutional networks

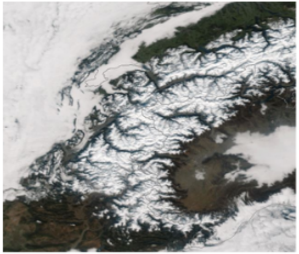
Project description

# Image classification

Image classification (in the context of remote sensing) or semantic image segmentation (in the context of machine learning/deep learning) is a problem of segmenting the whole digital image into semantically meaningful classes.

There are several types of features that help to classify objects in remotely sensed images:

- spectral
- textural
- shapes
- topological


# Image classification

In general, one can identify three approaches to address the classification problem:

- Pixel-to-pixel (the majority of the classification methods; can take into account only the spectral information)

$$\mathbf{y_{ij}} = f(\mathbf{x_{ij}}) + \epsilon \tag{1}$$

- Patch-to-pixel (textural analysis, convolutional neural networks, transformers)

$$\mathbf{y_{ij}} = f(\mathbf{X_{i:i+h,j:j+w}}) + \epsilon \tag{2}$$

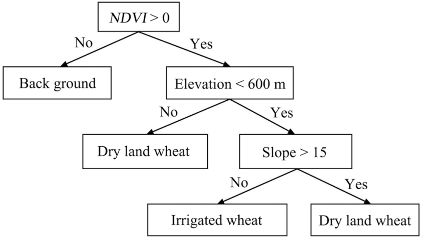
- Patch-to-patch (fully-convolutional neural networks, transformers)

$$\mathbf{Y_{i:i+h,j:j+w}} = f(\mathbf{X_{i:i+h,j:j+w}}) + \epsilon \tag{3}$$

# Random forest

Random forest—an ensemble of decision trees trained to solve one problem.

Decision trees:




Source: Meng et al., 2014

- No-leaf nodes are splits (rules to build decision boundaries)
- Leaf nodes are decisions

- To make a split, compare how every feature threshold reduces 'impurity', e.g., Gini impurity
- $gini_{leaf} = 1 - p_0^2 - p_1^2$
- $gini_{split} = \frac{N_{yes}}{N_{yes}+N_{no}} gini_{yesleaf} + \frac{N_{no}}{N_{yes}+N_{no}} gini_{noleaf}$
- At each split, choose a threshold to minimize $gini_{split}$ in a greedy manner
- After every leaf is 'pure', assign labels to the leaves

# Random forest

The output of a random forest model is the average prediction of the decision trees:

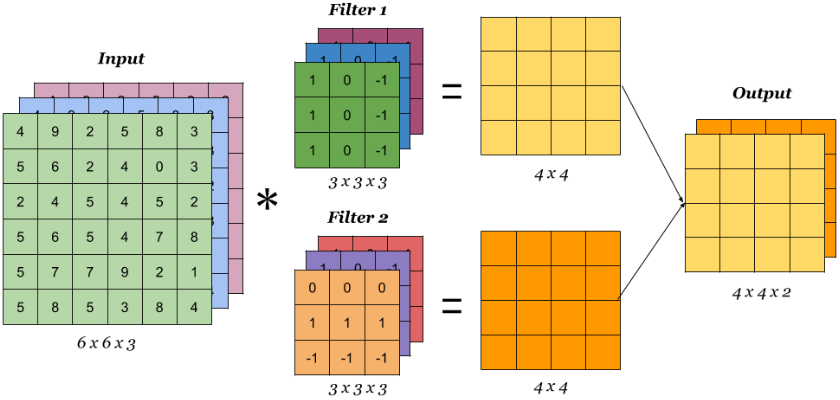
$$forest(\mathbf{x}) = \frac{1}{N_{trees}} \sum_i tree_i(\mathbf{x}). \tag{4}$$

But we do not want the decision trees to be correlated, so

- use data bootstrapping
- use feature bootstrapping at each split

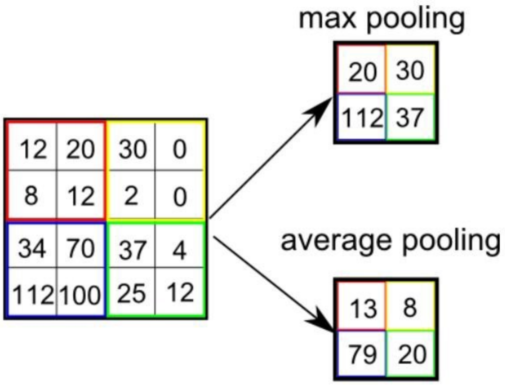
# Fully-convolutional networks: convolutional layers

Let's first recall what convolutional neural networks are.



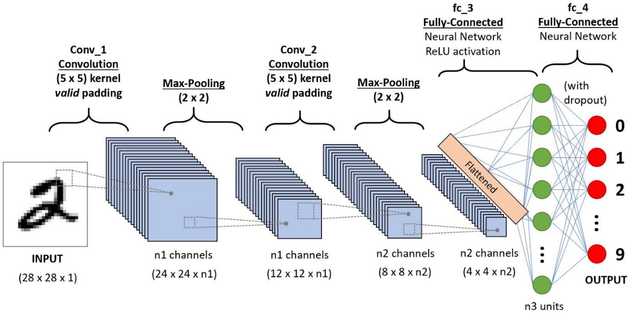

Source: `https://bit.ly/3eoHAJd`

# Fully-convolutional networks: pooling/downsampling layers




Source: `https://bit.ly/3L0BHhL`

# Fully-convolutional networks: typical convolutional network topologies

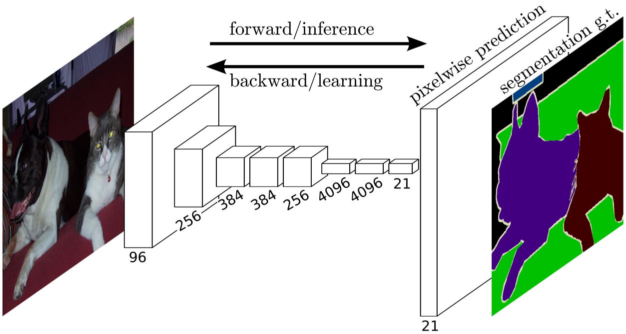



Source: `https://bit.ly/3wU3ryu`

- One inference to classify one pixel
- Computationally intensive
- Still can neglect some spatial correlations producing ‘noisy’ results

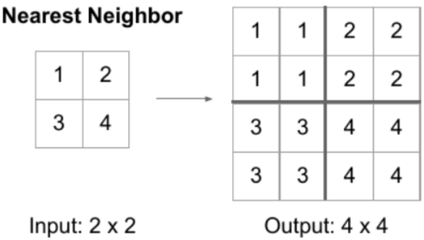
# Fully-convolutional networks

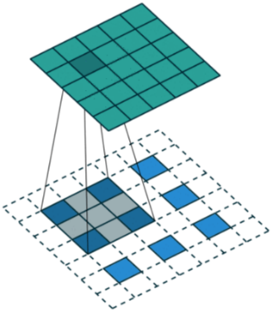
Can we do better? Yes!

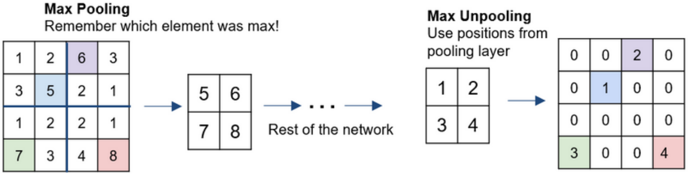



Source: Long et al., 2014

# Fully-convolutional networks: upsampling layers

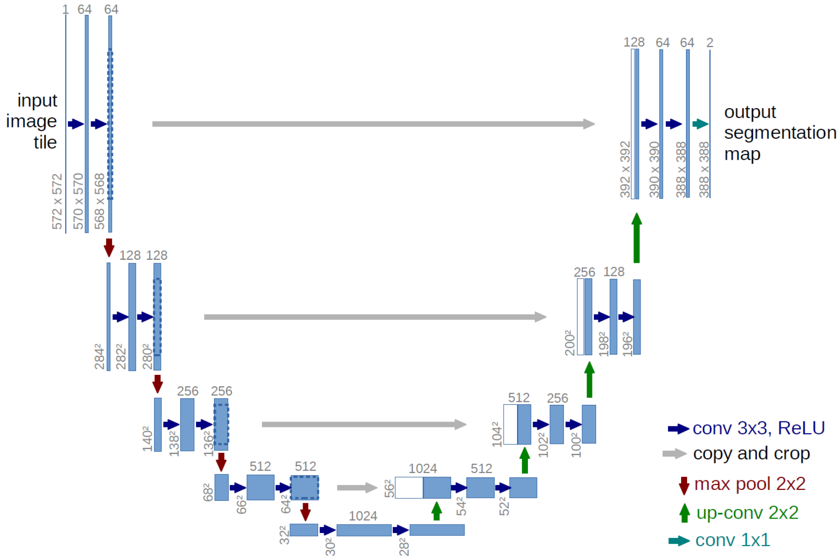




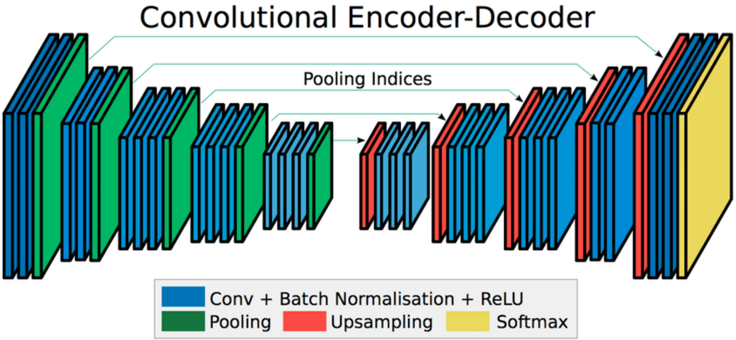

Source: `https://bit.ly/3RkuCL6`

# Fully-convolutional networks: U-Net




Source: Ronneberger et al., 2015

# Fully-convolutional networks: SegNet




Source: Badrinarayanan et al., 2015

# Fully-convolutional networks: DeepLab

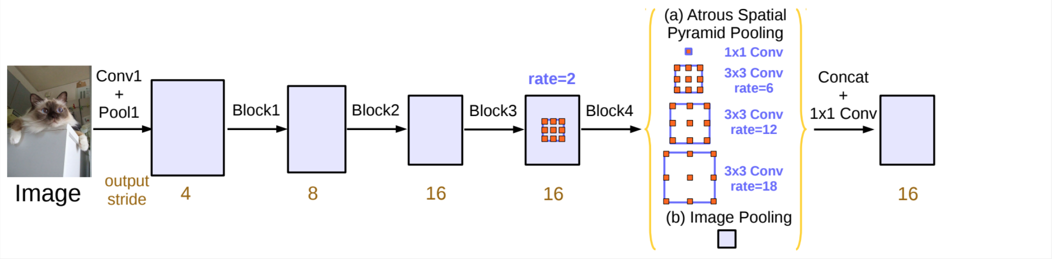
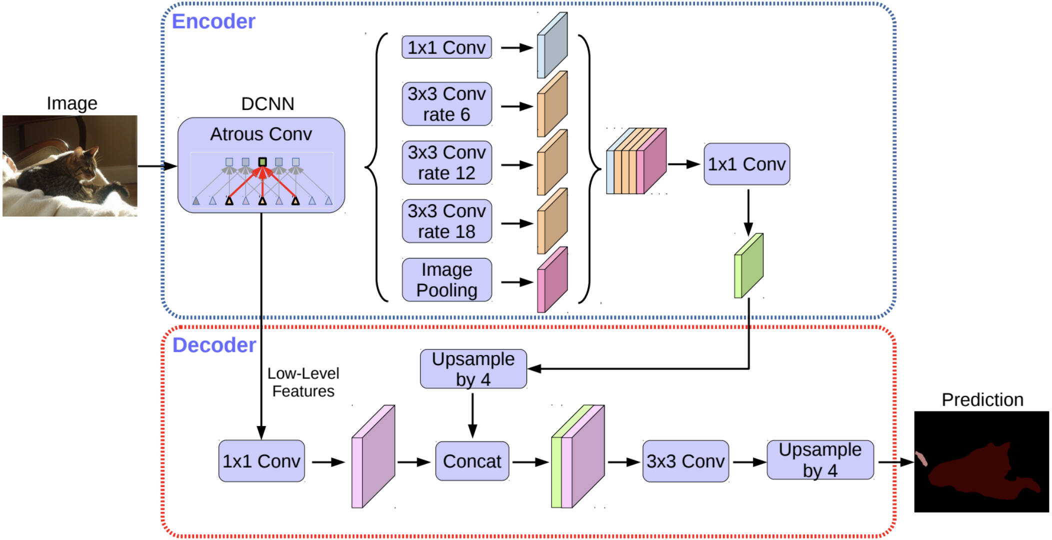



Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

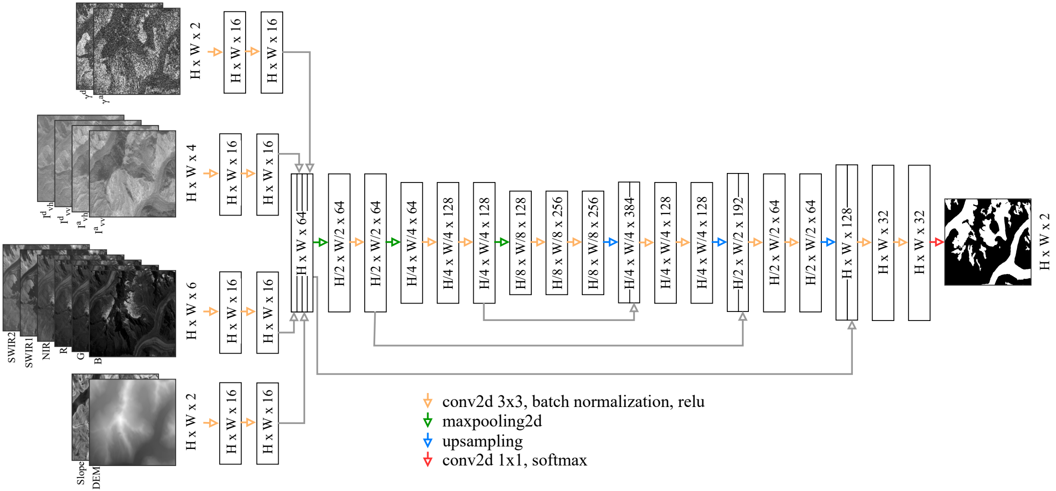
Source: Chen et al., 2017

# Fully-convolutional networks: DeepLabV3+

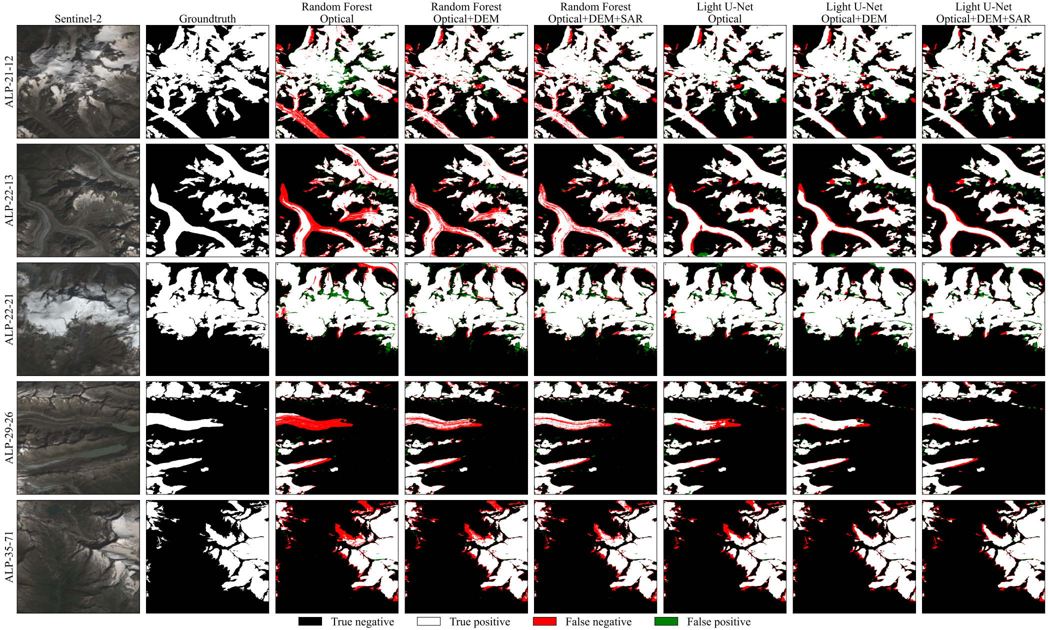



Source: Chen et al., 2018

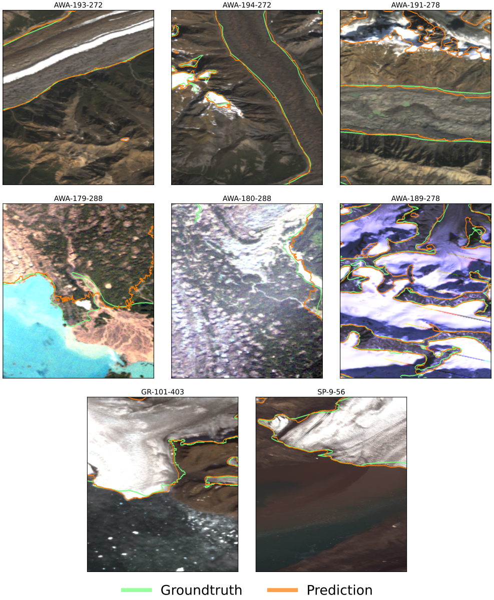
# Case studies: glacier mapping in the Alps



# Case studies: glacier mapping in the Alps



# Case studies: multi-regional glacier mapping in the Arctic

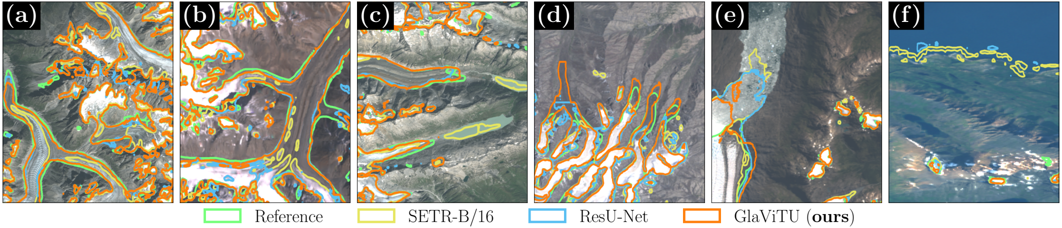



- In general, it nicely classifies debris-covered ice
- Surprisingly, it gives reasonable boundary estimates for vegetation-covered glaciers (but far from ideal)
- The model is robust to Landsat 5 artifacts at the scene boundaries
- Predictions for calving fronts are even better than groundtruth

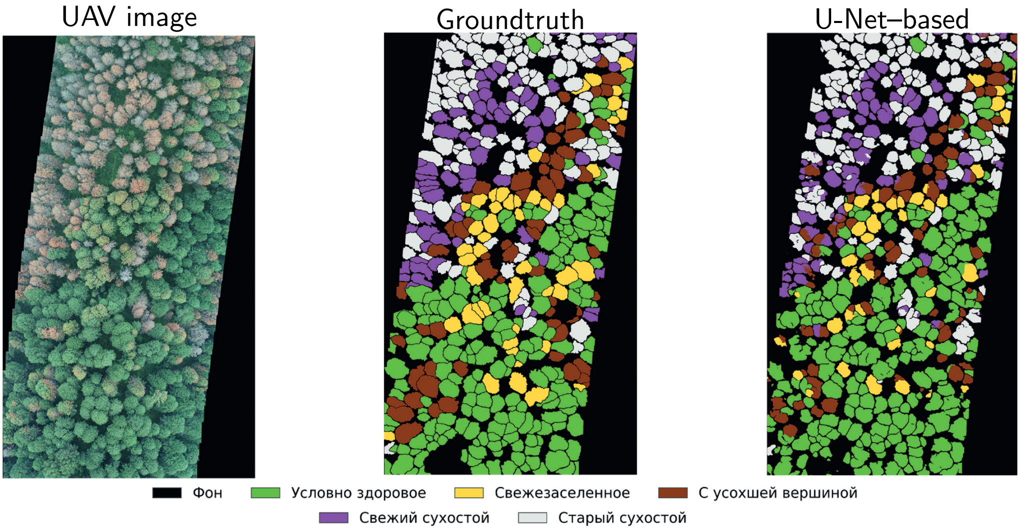
# Case studies: multi-regional glacier mapping with CNN-transformer hybrids

- Recently, we proposed a hybrid CNN-transformer model (GlaViTU) for multi-regional glacier mapping
- It has fewer parameters compared to ResU-Net and SETR-B/16 but shows higher performance and generalizes better

| Method | Params | IoU of different regions | | | | | | IoU mean | IoU std.dev. |
|---|---|---|---|---|---|---|---|---|---|
| | | ALP | HMA | LL | NZ | SA | SC | | |
| SETR-B/16 | 102M | 0.678 | 0.689 | 0.635 | 0.699 | 0.908 | 0.702 | 0.718 | 0.088 |
| ResU-Net | 33M | 0.843 | 0.803 | 0.837 | 0.833 | **0.955** | 0.829 | 0.850 | 0.049 |
| GlaViTU (**ours**) | **10M** | **0.844** | **0.812** | **0.864** | **0.855** | 0.952 | **0.866** | **0.866** | **0.043** |



# Case studies: mapping forests damaged by pests in Western Siberia




- Just to note that the methods can be relatively easily adapted for solving other problems

# Project

In the project, we suggest going deeper with random forests and several deep learning models for glacier mapping in different world regions.

What we can offer for the beginning:

- A Jupyter notebook with the data preparation process
- A Jupyter notebook showing how to classify images with random forest
- A Jupyter notebook with fully-convolutional networks, from training to producing vector outputs
- Two pretrained models—U-Net, DeepLabV3+ (slightly modified)
- $\approx$ 200 GB dataset with optical imagery, SAR and DEM data that covers some of the Alps, Southern Andes, New Zealand and two areas in the Himalayas

# Project: options

- Applying the pretrained models to other regions
- Tweaking the deep learning models
- Modifying the training routines
- Hyperparameter tuning for random forest
- Labelling snow cover and training multi-class classification models
- Exploring how different feature sets affect classification performance

  And more...

- Do not hesitate to suggest your ideas!