A pre-study about using artificial intelligence for semantic segmentation of Swedish wetland types

Aleksis Pirinen, RISE Research Institutes of Sweden

1 Introduction

In this report we summarize a pre-study[1] about using artificial intelligence (AI) – more specifically deep learning – for semantically segmenting five different Swedish wetland types. Semantic segmentation refers to the task of classifying every individual pixel in a spatial input map. In plain language, this corresponds to indicating various connected regions within the spatial input, and specifying what those regions are (e.g. what wetland types they correspond to; see conceptual explanation in Figure 1). Code for this project is available at `https://github.com/aleksispi/ai-swetlands`.

Funding and acknowledgements. This study was conducted by RISE Research Institutes of Sweden (Aleksis Pirinen) for the Swedish Environmental Protection Agency (henceforth abbreviated NVV, contact person Matti Ermold) during September - November, 2022. It was NVV that funded this work. We would like to thank William Lidberg at the Swedish University of Agricultural Sciences for all the help with transforming model predictions to a format that works when visualizing in GIS.

2 Dataset overview

Matti provided Aleksis with the various types of spatial input and output data for the project. The full set of possible input data consists of the following 10 types:

- Base vegetation / hydrology type classes assessed by the satellite monitoring of Swedish wetlands
- Wetland type from the Swedish wetland inventory
- Soil moisture index (from NVV) and soil moisture (from SLU)
- National land use cover (from NVV)
- Bush height and cover (from NVV)
- Tree height and cover (from NVV)
- Height map (from Lantmäteriet)

[1] See also `https://www.naturvardsverket.se/om-oss/aktuellt/nyheter-och-pressmeddelanden/ai-teknik-testas-for-att-identifiera-vatmarker/`.

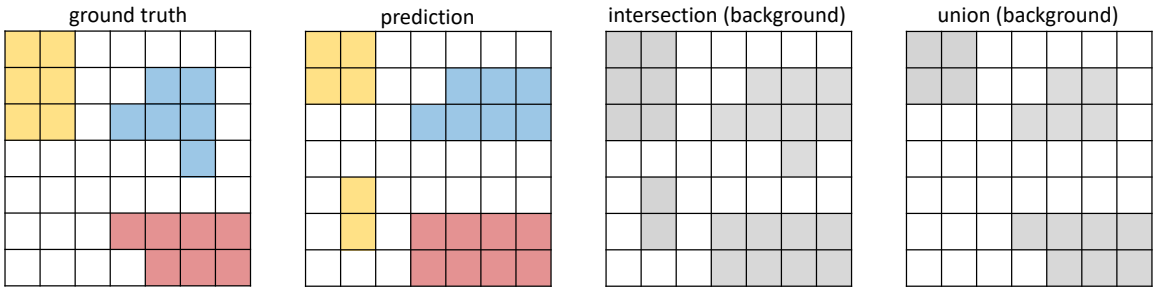

Figure 1: Conceptual explanation of semantic segmentation, as well as the main evaluation metric of mean intersection-over-union (mIoU). Column 1: Ground truth (GT) segmentation mask, where each grid cell corresponds to a pixel in the associated spatial map. There are three semantic categories (corresponds to wetlands) present in the spatial map; their pixel segmentations are shown in yellow, blue and red, respectively. White represents the background category (no wetland). Column 2: Predicted segmentation mask. None of the three regions of the GT are perfectly predicted and there is a fourth yellow region that is incorrectly predicted. The mIoU is the mean IoU over the four categories and is equal to $(4/8 + 5/8 + 7/8 + 25/33)/4 \approx 0.69$ (order: yellow, blue, red, white). Note that in practice the mIoU is computed over a dataset of spatial maps, not only a single map. Columns 3 - 4: The intersection and union, respectively, of background pixels in the GT and prediction masks. White and gray correspond to background and non-background (wetland) pixels, respectively. Figure borrowed from [4]. Visual examples on actual wetland predictions given in Figure 2 - 16.

The output data corresponds to the quantities that we want the AI model to predict. In this case, the output data corresponds to annotations of five different wetland types. The wetland types described here are used by the EU habitats directive:

- Högmosse (nature type code 7110, 7111)
- Rikkärr (nature type code 7230-7233)
- Öppna mosse (nature type code 7140-7143)
- Aapamyr (nature type code 7310)
- Källor (nature type code 7160)

The output data that contains the five different wetland types also contains numerous other nature types, and this set of various nature types is used during the training (calibration) process of the AI model (see Section 3.1).

All input and output data was provided as a spatially aligned set of raster maps, of size $H \times W = 146,258 \times 64,034$ pixels (10 aligned such layers for the input, and 5 for the output; see the specifics above). This corresponds to a 10×10 meter resolution[2] of Sweden from above. This data was split into rectangular regions of size $h \times w = 100 \times 100$ pixels (corresponding roughly 1000×1000 meters), with 3 different offsets relative to the origin $(0, 0)$ (one starting at $(0, 0)$, one at $(50, 0)$, and one at $(0, 50)$). These 100×100 regions are henceforth denoted *examples*. Then, only the examples that

[2]Exceptions: The height map obtained is of resolution 50×50 meters, and the base vegetation map is of resolution 25×25 meters. These were upscaled (interpolated) to correspond to the 10×10 resolution.

contain at least one wetland pixel were kept. Note that all other examples were entirely void of any annotated wetlands, even though they could contain wetlands that have not been annotated – we omitted these examples to not cause any potential confusion for the AI model.

The data was then split into training and validation (roughly 80% training, 20% validation) in such a way that both data splits cover parts across all of Sweden while having no overlap between any 100×100-regions between the respective splits (the last part ensures that a trained model will not have seen any validation data, and thus that the model's ability to generalize to unseen data can be assessed). This resulted in 107,966 training examples and 21,810 validation examples. In the full dataset (training and validation set taken together), we have

- 100.0% examples containing at least one background pixel,[3] and 85.1% background pixels on average across all examples;
- 2.9% examples containing at least one högmosse pixel, and 0.7% högmosse pixels on average across all examples;
- 7.7% examples containing at least one rikkärr pixel, and 0.5% rikkärr pixels on average across all examples;
- 89.7% examples containing at least one öppna mosse pixel, and 8.0% öppna mosse pixels on average across all examples;
- 34.6% examples containing at least one aapamyr pixel, and 5.7% aapamyr pixels on average across all examples;
- 1.4% examples containing at least one källor pixel, and 0.03% källor pixels on average across all examples.

If one considers the training and validation sets individually, the above statistics look the same, i.e., the characteristics of the two sets are very similar. Also note the high imbalance between categories in the dataset, where the background category is by far the most common one. There is extremely little 'källor' data in comparison.

3 Methodology

In this section we provide an overview of the approach that we have considered for tackling the wetland segmentation task. In Section 3.1 we describe the AI model that we have used and how it is trained, and in Section 3.2 we provide some details regarding data preprocessing.

3.1 Overview of semantic segmentation AI model

Since the task is to predict the spatial extents and classes of various wetland types based on spatial input of size $h \times w \times C_{\text{in}} = 100 \times 100 \times 10$, we have opted for one of the standard deep learning architectures, namely a fully-convolutional neural network (FCN) [3]. The approach is very straightforward – a spatial input is fed in at one end, and a spatial output is obtained at another end. In this case, the output has the same spatial size $h \times w = 100 \times 100$ but a different number of output channels $C_{\text{out}} = 5 + 1 = 6$, i.e., C_{out} is the number of wetland types considered, including a 'background' category which corresponds to pixels that do not represent any of the five wetland types.

[3] The 'background' category corresponds to pixels that do not represent any of the five wetland types.

Once the input-output mapping (i.e., AI model) has been selected, the parameters of the model are set by training (calibrating) the model on training data. We do this using the pixel-wise cross-entropy loss [1], which encourages the network's predicted probability distribution per pixel to be high for the correct-class entry (i.e. for the correct wetland type), and small for the rest. We only provide the loss in the *support area* of each training example, and this is defined by a binary mask that checks if any of the nature types (including the wetland types to predict, cf. Section 2) appear in a pixel. This support is used in the loss computation, because it is not certain what exists beyond the support – it could for example be that there are wetland pixels outside the support, but that they simply have not been annotated, e.g. because they are outside a protected area. If the support was not included, the AI model would always strive to predict 'background' outside the support – this would be incorrect in those cases where there in fact are wetlands but that have not been annotated.

Some technical details that can be skipped if desired: For model parameter optimization, we resort to Adam [2] with batch size 64 (i.e., 64 training examples are used per model parameter update), learning rate $2 \cdot 10^{-4}$ (affects how much the model parameters are updated per step), and momentum 0.9 (a typical technique used in deep learning). The model is trained for $250,000$ batches, which takes about 48 hours on the GPU-equipped (Titan V100) work station that is used for experimentation. To improve model generalization towards unseen data, we resort to the customary deep learning training technique of *data augmentation*. In our case, we randomly flip each training example horizontally or vertically (independently, with 50% probability each), which effectively results in 4x more training examples.

3.2 Data preprocessing

Note: This subsection can probably be omitted unless one is specifically interested in some technical details of the approach.

As is customary within deep learning, the input data is preprocessed prior to feeding it to the model. For discrete input layers (i.e., such input data layers whose pixel values are integers) we perform the following preprocessing:

1. Figure out which value corresponds to 'background' (typically 0, 128 or 255) and ensure that it is set to 0.
2. Find all unique values $[u_1, u_1, \ldots, u_n]$, with $0 = u_0 < u_1 < \ldots < u_n$.
3. Set all u_1 to $1/N$, all u_2 to $2/N$, and so on, so that the unique values are $\{0, 1/N, \ldots, 1\}$.

For continuous input layers we perform the following preprocessing:

1. Figure out which value corresponds to 'background' and ensure that it is set to 0.
2. Divide the layer by the largest value, so that all values are in the $[0, 1]$-range.

Note that the preprocessing occurs for the large data layers (of size $H \times W = 146,258 \times 64,034$ pixels) prior to model training. Hence it is *not* performed on the basis of individual 100×100 input layers.

Model	mIoU	mIoU-no-k	IoU-bg	IoU-h	IoU-r	IoU-o	IoU-a	IoU-k
Main	0.51	0.61	0.92	0.71	0.30	0.50	0.60	0.03
No height	0.49	0.58	0.92	0.67	0.25	0.48	0.58	0.01
No bushes+trees	0.47	0.56	0.92	0.67	0.21	0.45	0.56	0.01
No land cover	0.48	0.57	0.92	0.67	0.23	0.47	0.56	0.00
No soil moist	0.48	0.57	0.92	0.68	0.23	0.48	0.56	0.01
No wetland inventory	0.50	0.60	0.92	0.69	0.28	0.50	0.59	0.03
No base vegetation	0.50	0.59	0.92	0.70	0.28	0.47	0.58	0.02
Only-h	0.89	-	0.90	0.88	-	-	-	-
Only-r	0.67	-	0.93	-	0.40	-	-	-
Only-o	0.70	-	0.92	-	-	0.48	-	-
Only-a	0.75	-	0.90	-	-	-	0.61	-
Only-k	0.67	-	0.99	-	-	-	-	0.38

Table 1: Results on the validation set of various trained semantic segmentation models for five Swedish wetland types. The top seven rows correspond to models which predict all wetland types using a single model, whereas the bottom five rows correspond to binary models (predicting only one wetland type per model). Omitting the height map from the model input slightly decreases the mIoU and individual IoUs, and more so if omitting bush and tree data from the input. The removal of the national land cover yields the same average result as if removing soil moist (both of which yield slightly worse results than the main model). Removing the wetland inventory or base vegetation as input hardly decrease performance at all. All models obtain the best results for the högmosse category, and the worst results for the källor category (the latter has extremely little training data in comparison to the other wetland types, cf. Section 2). As for the models predicting a single wetland type, the results improve significantly for the högmosse (+0.17), rikkärr (+0.10), and källor (+0.35) categories, while they remain roughly the same for öppna mosse (-0.02) and aapamyr (+0.01).

4 Experimental results

In this section we present the empirical results of our model experimentation. Results are always presented on validation data, i.e., data which the models have not seen during the model training process. We use the mean intersection-over-union (mIoU) as the evaluation metric. It measures how well the predicted segmentation mask matches the ground truth (GT) mask. See Figure 1 for a conceptual explanation.

We train and evaluate the following models:

- *Main* is the model described in Section 3.1, which simultaneously predicts six possible categories per pixel (five different wetland types, and a "no wetland", or background, category). It uses all the ten data layers described in Section 2 as input.
- *No height* is the same as *Main* but omits the height map in the input.
- *No bushes+trees* is the same as *Main* but omits the bush and tree heights and covers (four layers) in the input.
- *No land cover* is the same as *Main* but omits the national land cover map in the input.
- *No wetland inventory* is the same as *Main* but omits the wetland inventory map in the input.
- *No base vegetation* is the same as *Main* but omits the base vegetation map in the input.

- *Only-h*, *only-r*, *only-o*, *only-a* and *only-k* are single-category predictors. With this is meant that they are trained to predict one wetland category per model, so *only-h* categorizes each pixel as either högmosse or background (not högmosse), and similar for *only-r* (rikkärr), *only-o* (öppna mosse), *only-a* (aapamyr) and *only-k* (källor).

In Table 1 we show the experimental results. The main model, presented in the top row of the table, shows that the mIoU reaches 0.51 on the validation set, although it is significantly higher if omitting the källor category (see the *mIoU-no-k* metric). Predictions on the källor category are abysmall on average (with an IoU of 0.03), which is due to the fact that such data is extremely scarce in comparison to the other wetland types (cf. Section 2). When training a single-category model for källor only (*only-h*), results improve significantly, to an IoU of 0.38.

Omitting the height map from the model input slightly decreases the mIoU and individual IoUs, and more so if omitting bush and tree data from the input. The removal of the national land cover yields the same average result as if removing soil moist (both of which yield slightly worse results than the main model). Removing the wetland inventory or base vegetation as input hardly decrease performance at all, which could be because at many spatial locations there is no such data available (see the visual examples in Figure 2 - 16, where the base vegetation is not present in any example). All models obtain the best results for the högmosse category, and the worst results for the källor category (the latter has extremely little training data in comparison to the other wetland types, cf. Section 2). As for the models predicting a single wetland type, the results improve significantly for the högmosse (+0.17), rikkärr (+0.10), and källor (+0.35) categories, while they remain roughly the same for öppna mosse (-0.02) and aapamyr (+0.01).

Visual examples of how the main model performs are given in Figure 2 - 16. We also took the best single-category model, *only-h* (i.e. a model that predicts only the presence of högmosse) and evaluated it in every location in Sweden, to map all högmosse wetlands in Sweden. These results are visualized in Figure 17 - 18. The difference between the results in Figure 17 and Figure 18, is that in the latter the model was trained using significantly more negative examples. This was done to prevent the model to incorrectly predict högmosse in the northern parts of Sweden, where there is very little of this wetland type.

5 Conclusions

In this pre-study we have provided an early AI baseline model for performing semantic segmentation of wetlands in Sweden. Models which predict all five wetland categories at the same time are currently not good enough for practical application, while some single-category models provide quite strong results on validation data. One of the key issues was the severe imbalance between various wetland categories. Obvious directions for future work include:

- Use techniques that handle the extreme imbalance in wetland categories, so that better results can be obtained for e.g. källor.
- Add reliable uncertainty measures to the model outputs (it is often useful during verification to get an estimate of how certain the model is in its predictions).
- Explore other (and in particular larger) input sizes than 100×100 for the models.

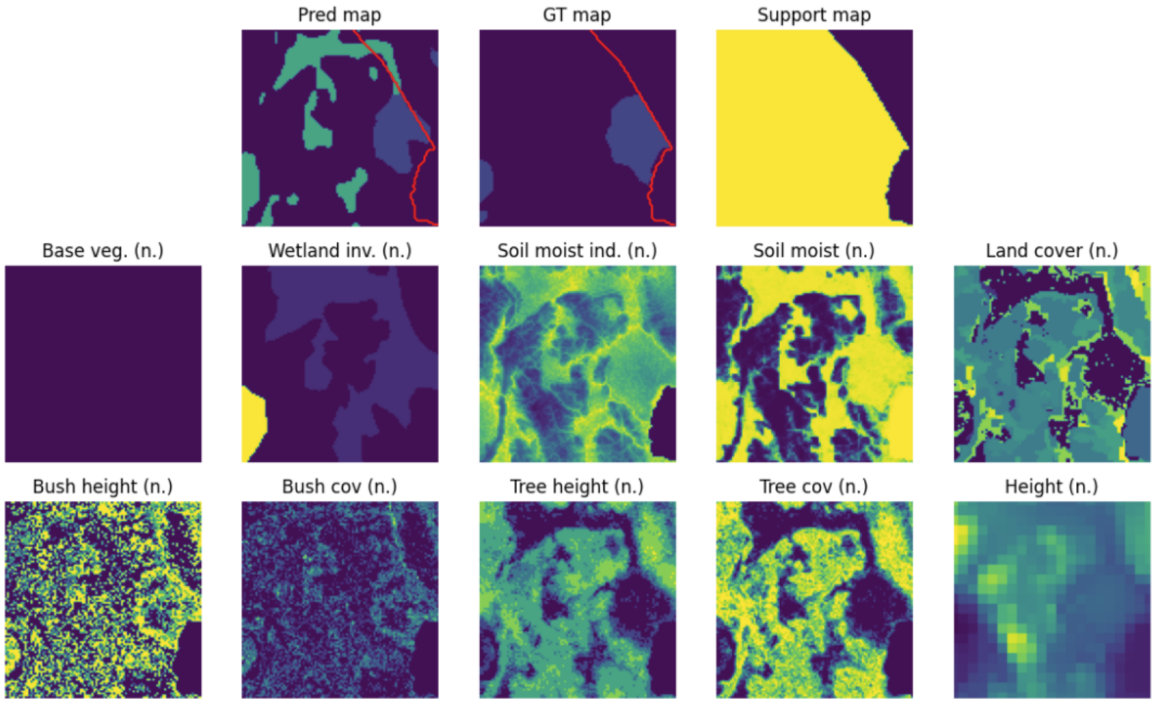

Figure 2: Qualitative example of the main model (which simultaneously segments all five wetland types and the 'no wetland' background category) on a högmosse patch in the validation set. Top row (from left to right): Model prediction, ground truth (GT) content, and support map (the latter is explained in Section 3.1 – it simply highlights the area where we would compute the loss had this been a training example). The middle and bottom rows contain all the ten model inputs described in Section 2. In this case, we see that the model correctly predicts most of the large högmosse region (shown in brighter blue on the right side of the GT map), but it misses the smaller högmosse region on the left. Also, the model incorrectly predicts that there is öppna mosse (shown in turquoise).

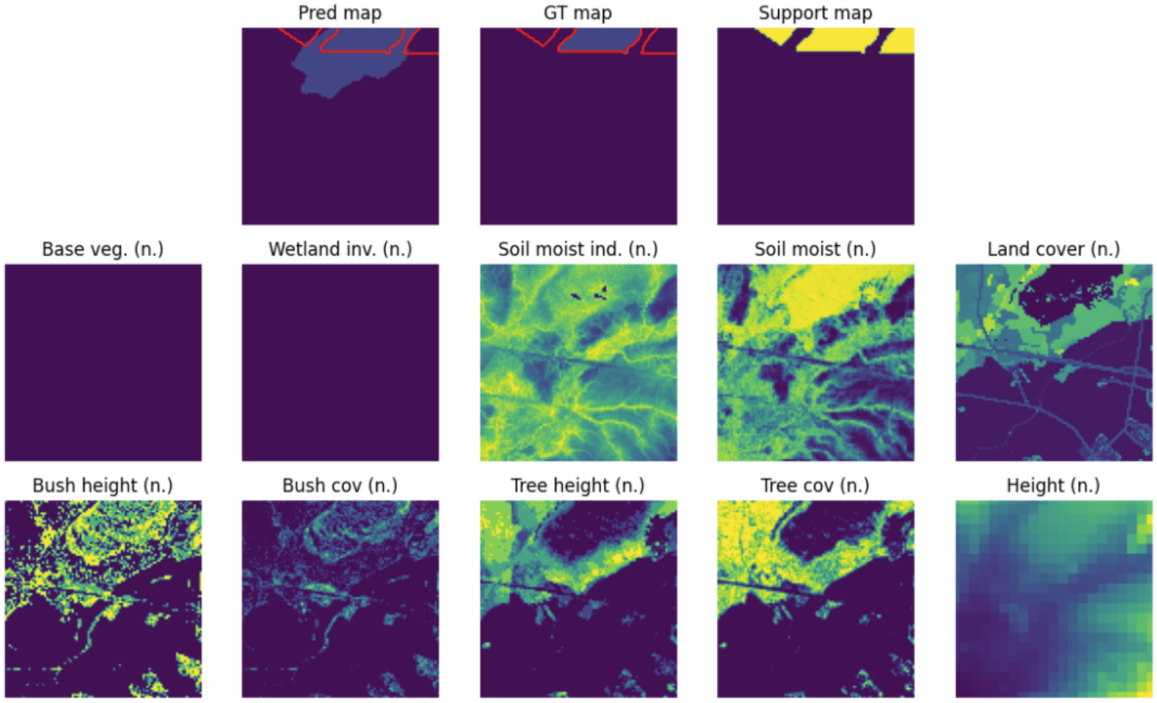

Figure 3: The model correctly predicts all of the högmosse region (shown in brighter blue on the top of the GT map). The predicted högmosse region is much larger than in the GT, but we cannot tell with certainty whether it is correct or incorrect outside the support area.

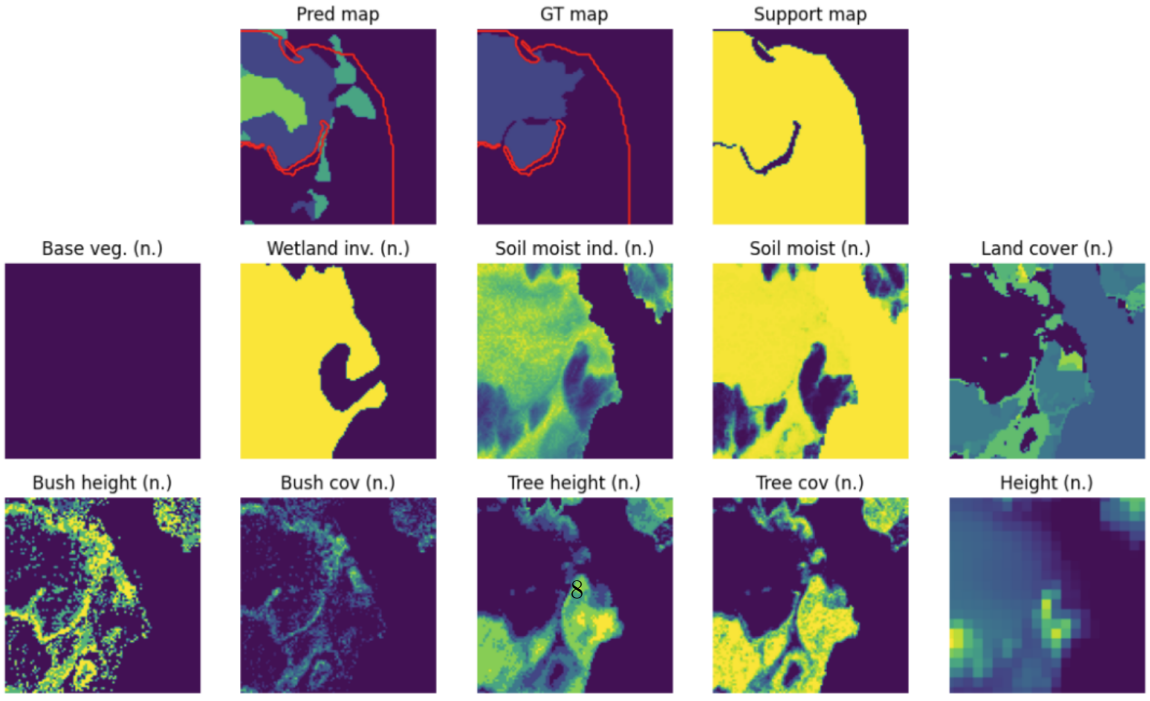

Figure 4: The model correctly predicts some of the högmosse region (shown in brighter blue in the GT map). Also, the model incorrectly predicts that there is öppna mosse (shown in turquoise) and aapamyr (shown in green).

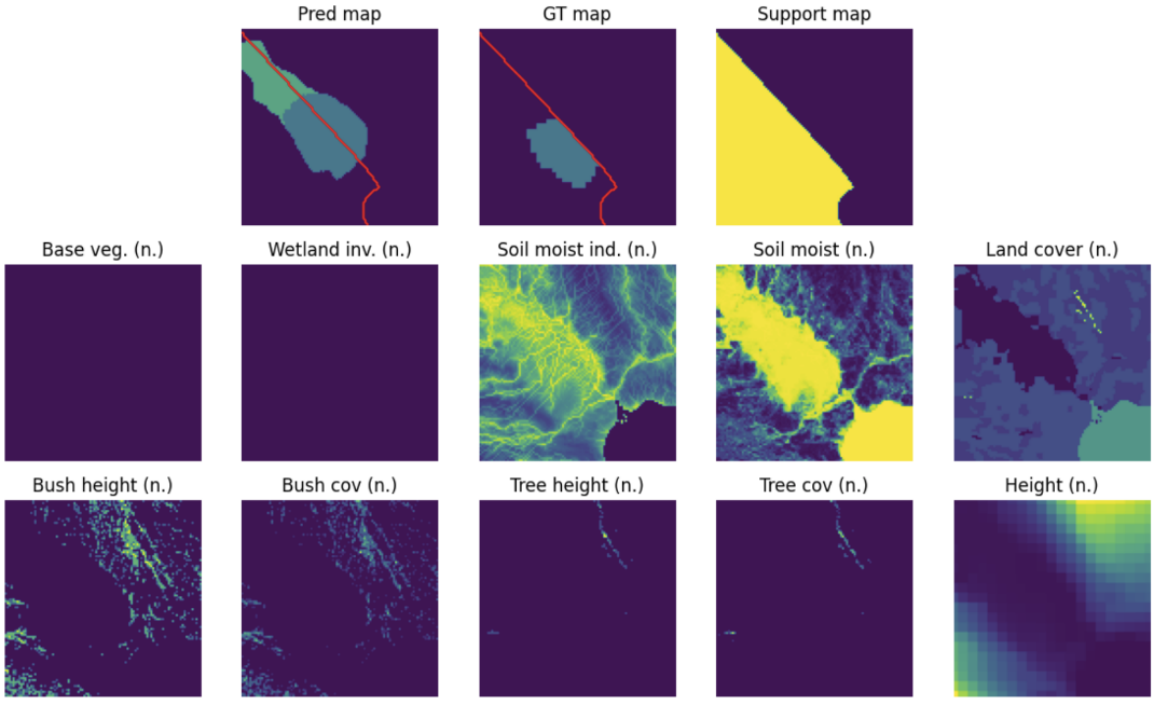

Figure 5: The model correctly predicts most of the rikkärr region (shown in brighter blue in the GT map), even though it is slightly too big. Also, there is some incorrectly predicted öppna mosse (turqoise), and the model predicts the existence of some wetlands outside the support area (which may or may not be correct).

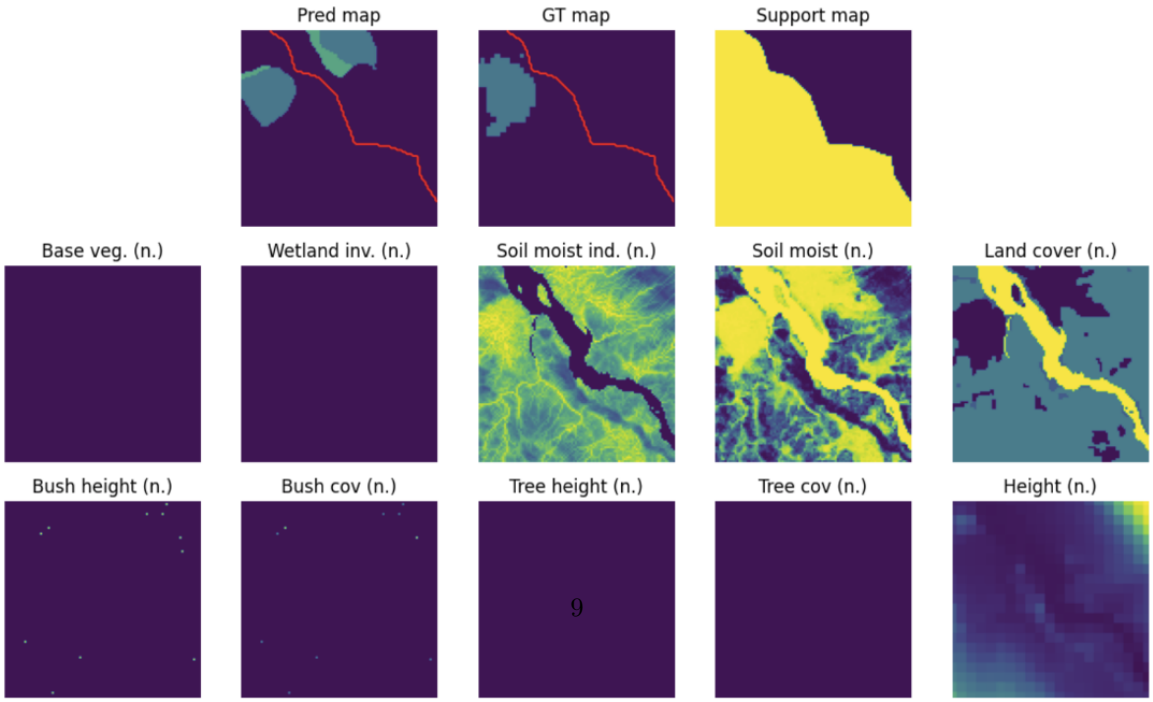

Figure 6: The model correctly predicts some of the rikkärr region (shown in brighter blue in the GT map), but it is a bit too small. Also, there is a little bit of incorrectly predicted öppna mosse (turqoise), and the model predicts the existence of some wetlands outside the support area (which may or may not be correct).

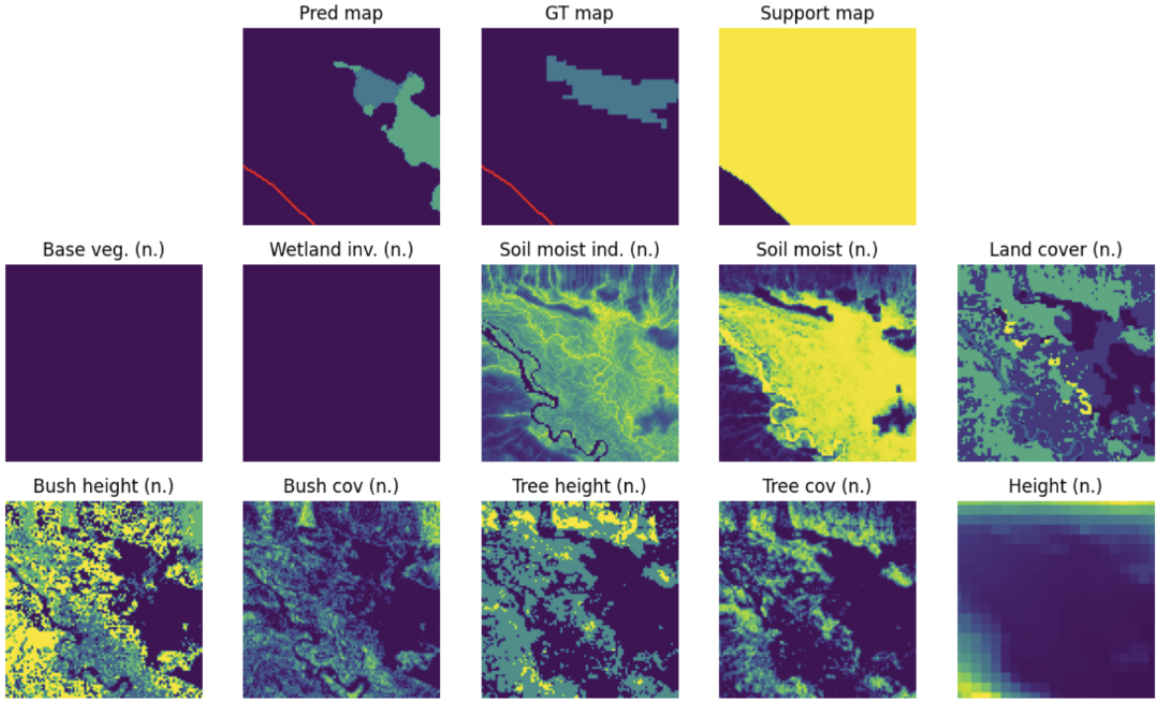

Figure 7: The model fails to predict most of the rikkärr region (shown in brighter blue in the GT map), and it incorrectly predicts quite a bit of öppna mosse (turqoise).

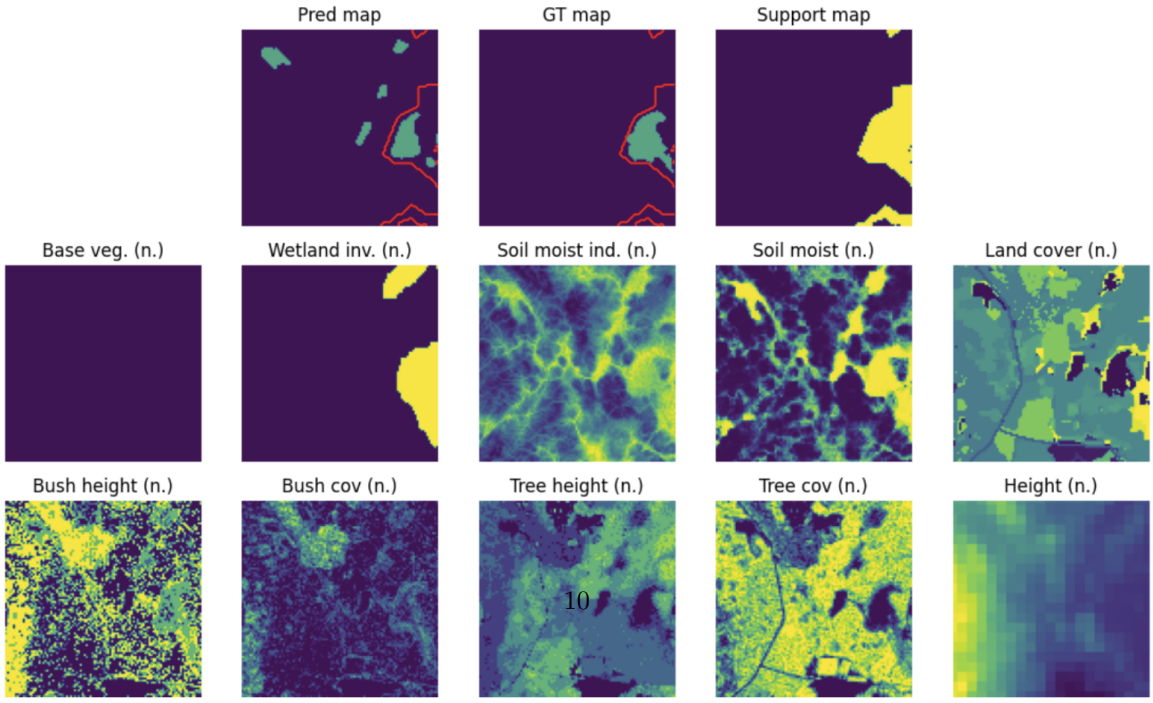

Figure 8: The model correctly predicts most of the öppna mosse region (shown in turquoise in the GT map). Also, the model predicts the existence of some öppna mosse outside the support area (which may or may not be correct).

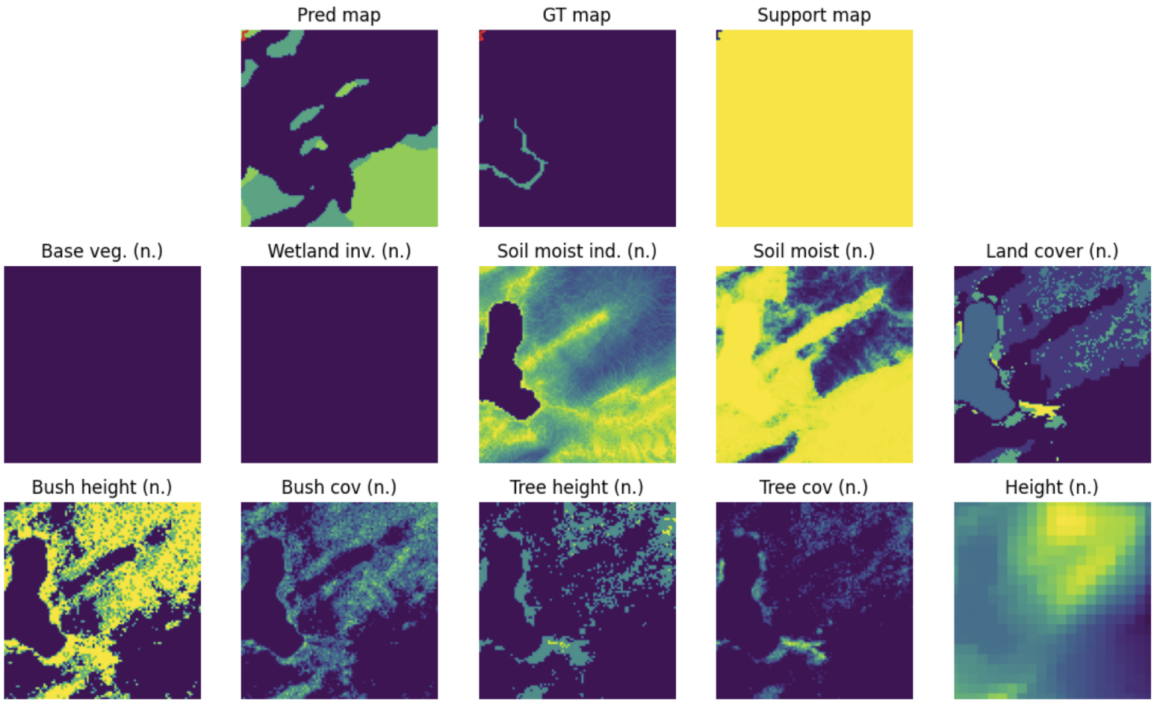

Figure 9: The model fails to predict the öppna mosse region (shown in turquoise in the GT map), and it incorrectly predicts quite a bit of aapamyr (green) and öppna mosse elsewhere.

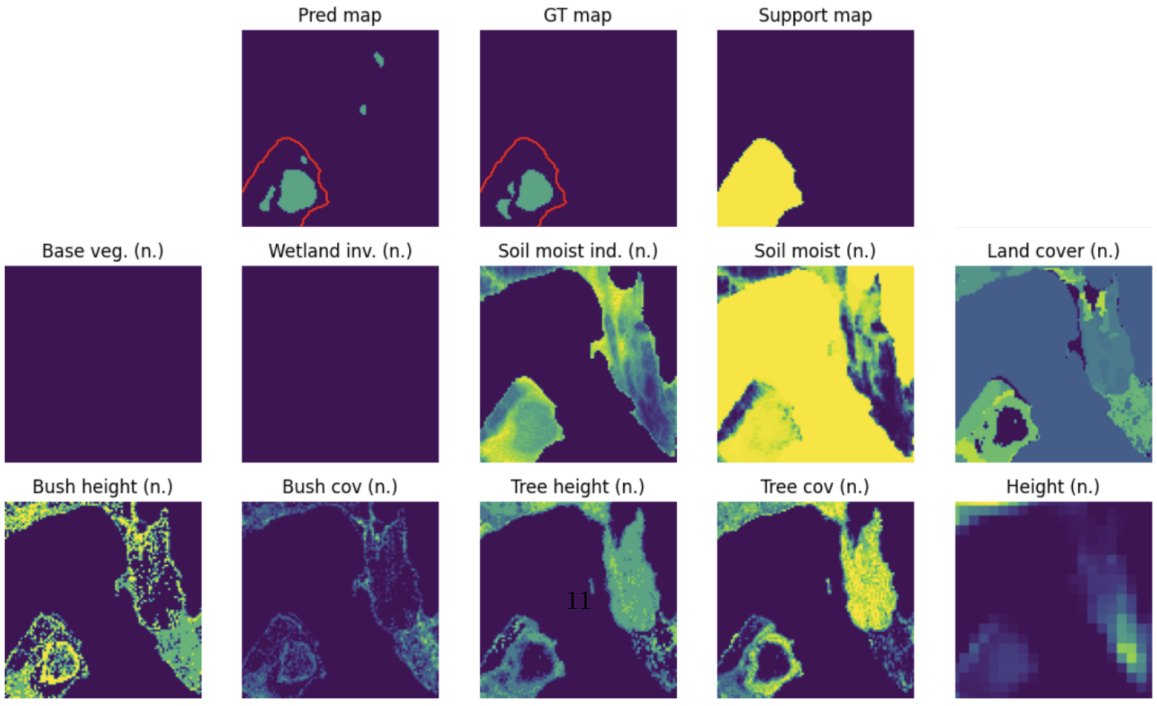

Figure 10: The model correctly predicts most of the öppna mosse region (shown in turquoise in the GT map). Also, the model predicts the existence of a tiny bit of öppna mosse outside the support area (which may or may not be correct).

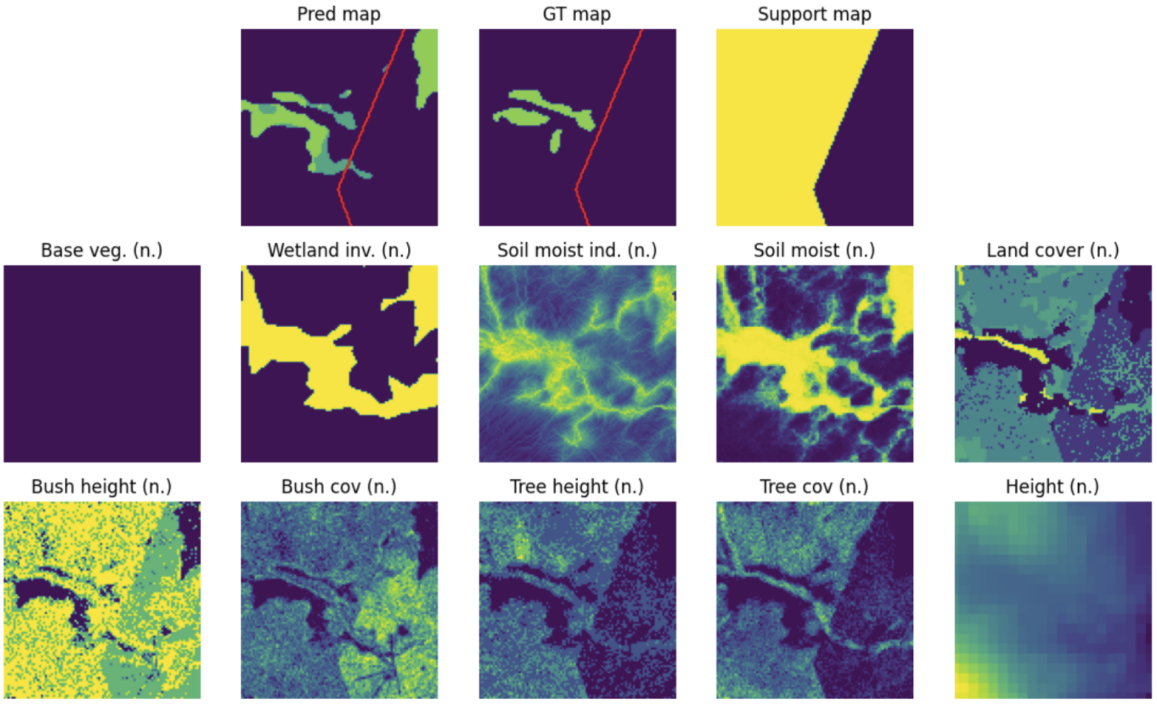

Figure 11: The model correctly predicts some of the aapamyr region (shown in green in the GT map), and it incorrectly predicts the existence of some öppna mosse (turqoise). Also, the model predicts the existence of some aapamyr and öppna mosse outside the support area (which may or may not be correct).

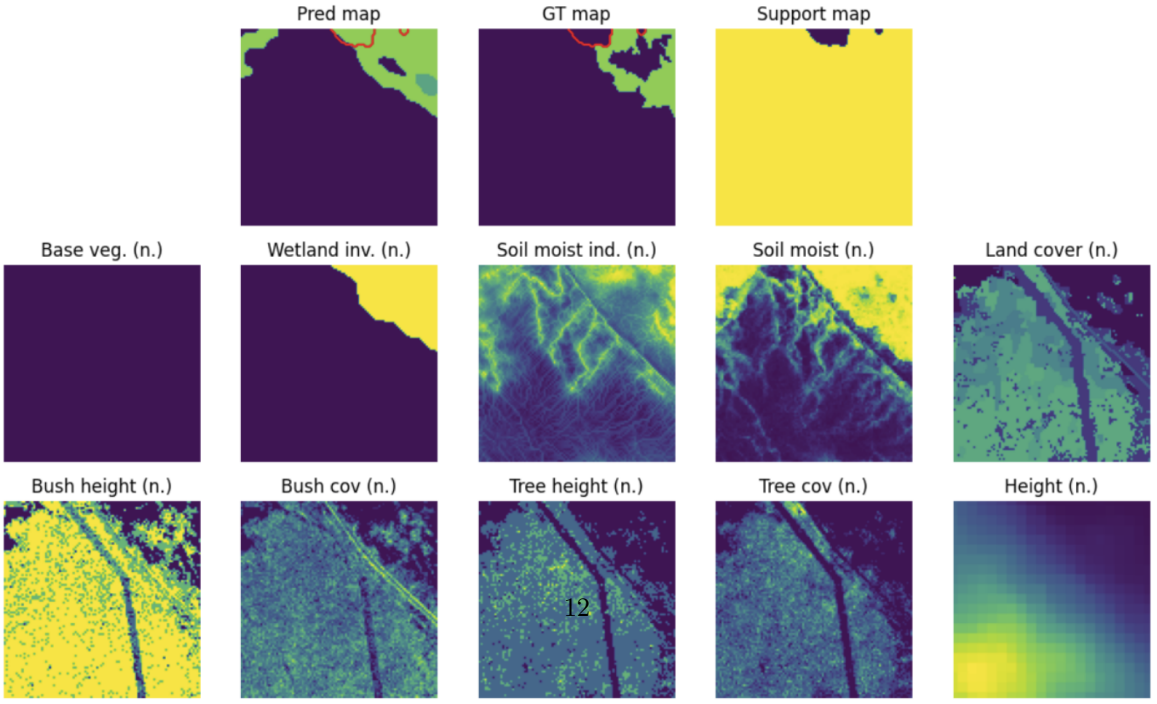

Figure 12: The model correctly predicts quite a bit of the aapamyr region (shown in green in the GT map), even though its extent is exaggerated (e.g. the non-wetland 'hole' is mostly predicted to be aapamyr). The model also incorrectly predicts some more aapamyr, and also predicts some aapamyr outside the support area.

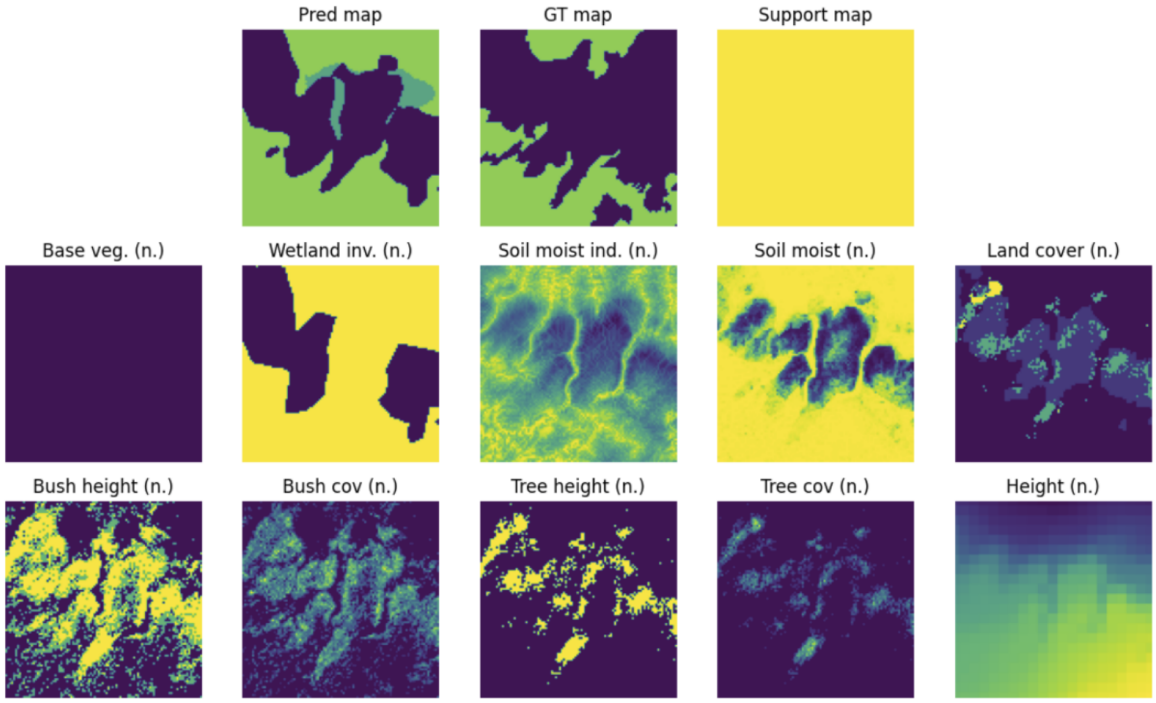

Figure 13: The model correctly predicts quite a bit of the aapamyr region (shown in green in the GT map), and in particular the overall structure of the aapamyr areas. The model also incorrectly predicts some öppna mosse (turquoise).

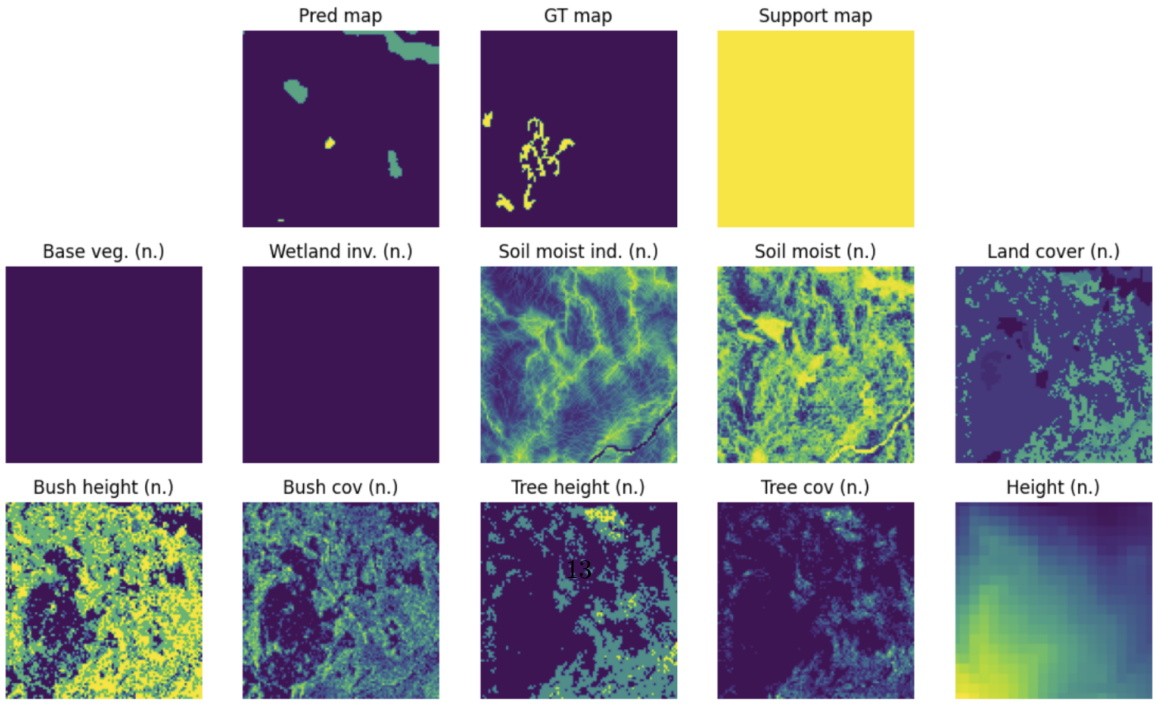

Figure 14: The model fails to predict the källor regions (shown in yellow in the GT map). It also incorrectly predicts some öppna mosse (turquoise).

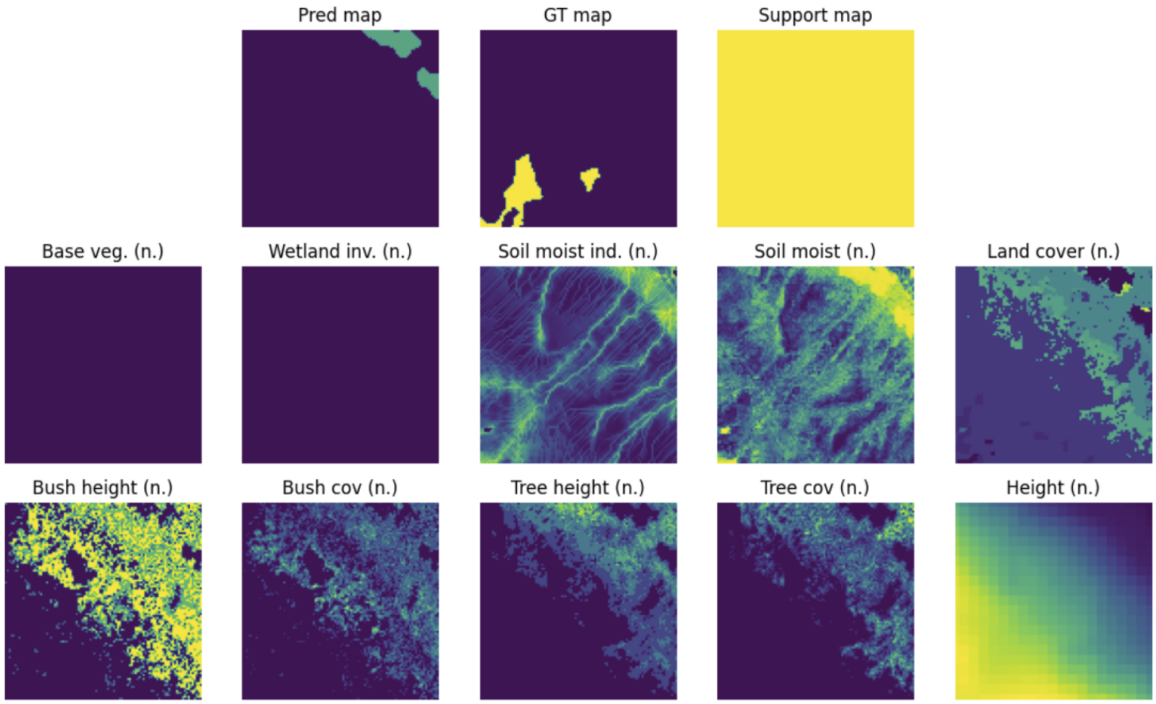

Figure 15: The model fails to predict the källor regions (shown in yellow in the GT map). It also incorrectly predicts some öppna mosse (turquoise).

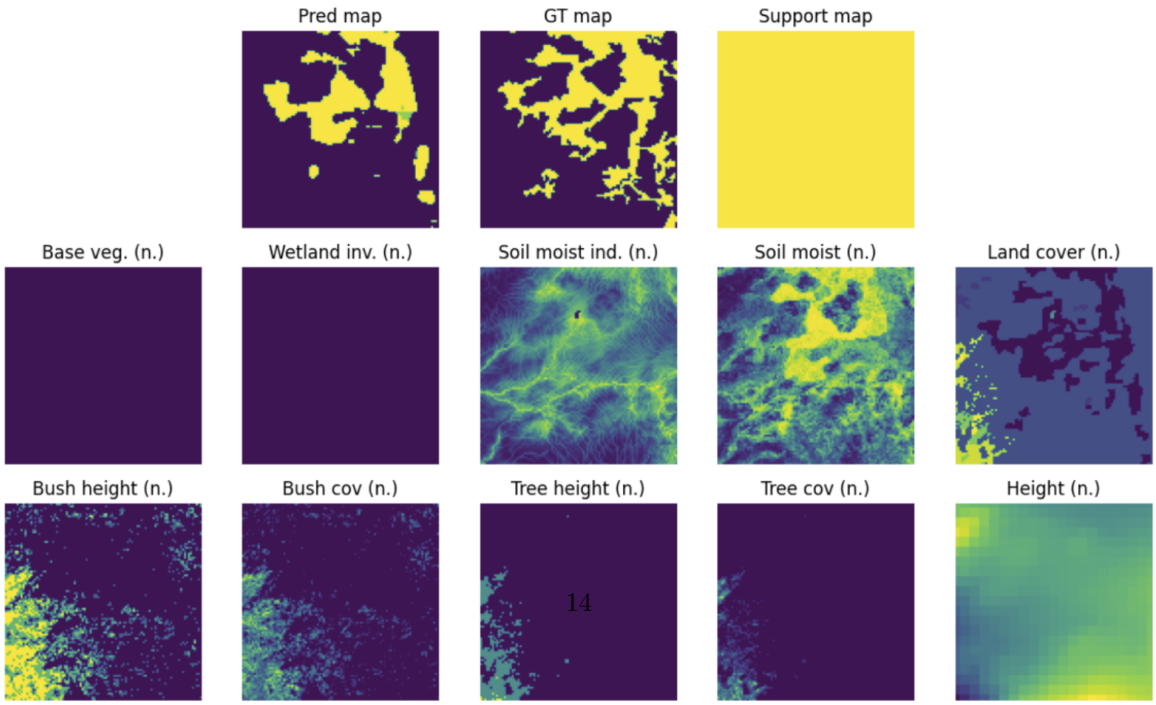

Figure 16: The model correctly predicts quite a bit of the källor regions (shown in yellow in the GT map), and in particular a lot of the overall structure is captured by the model. The model however quite rarely succeeds with källor examples, as also seen in Table 1.

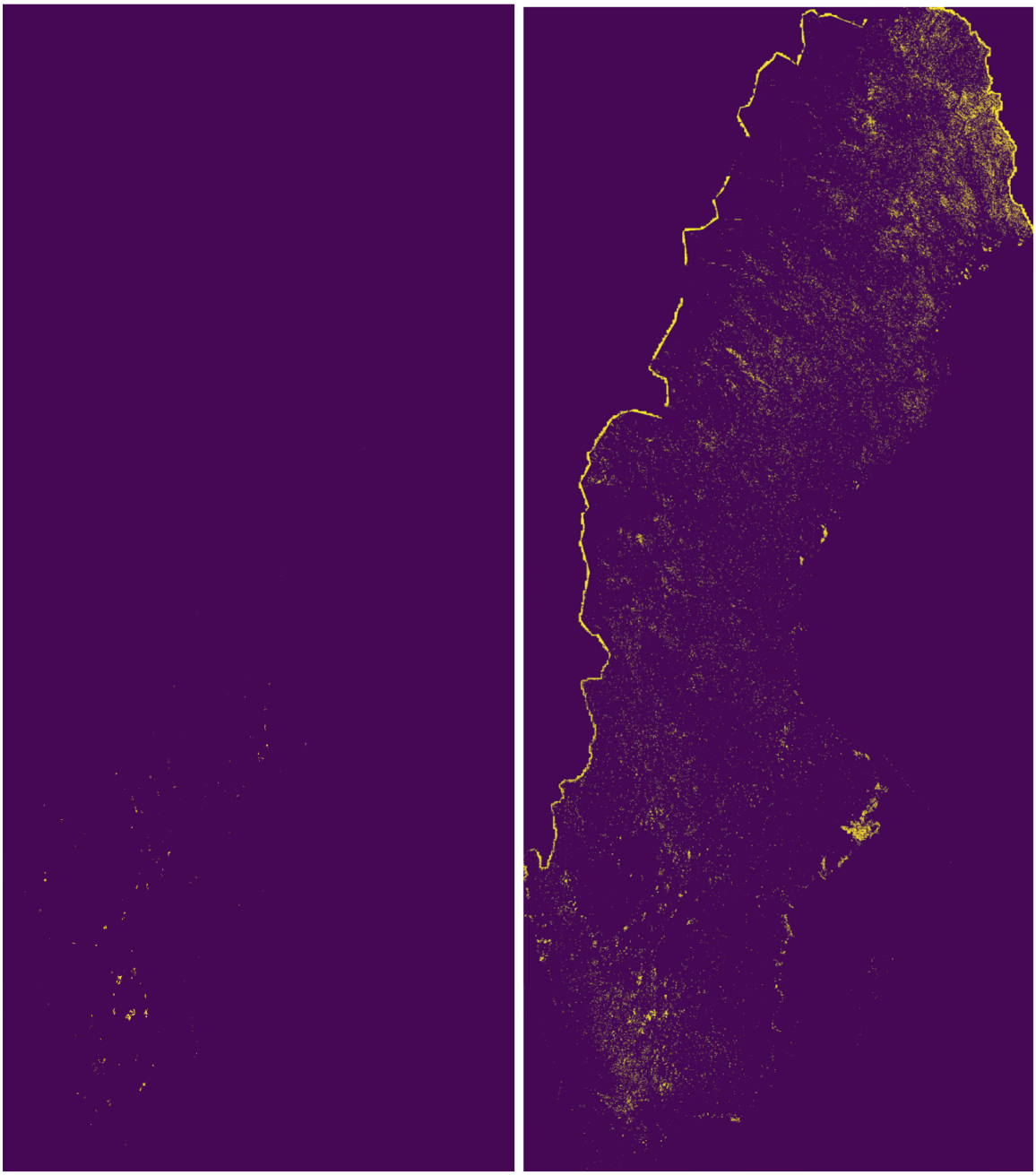

Figure 17: Left: Annotated högmosse in Sweden. It exists mostly in the southern parts of Sweden. Right: Predicted högmosse by a högmosse-only AI model. The model predicts excessive amounts of högmosse, in particular in the northern parts. The border of högmosse is a glitch that probably stems from there not being relevant input data exactly in the border regions. The model correctly predicts högmosse in those parts where the annotations contain högmosse.

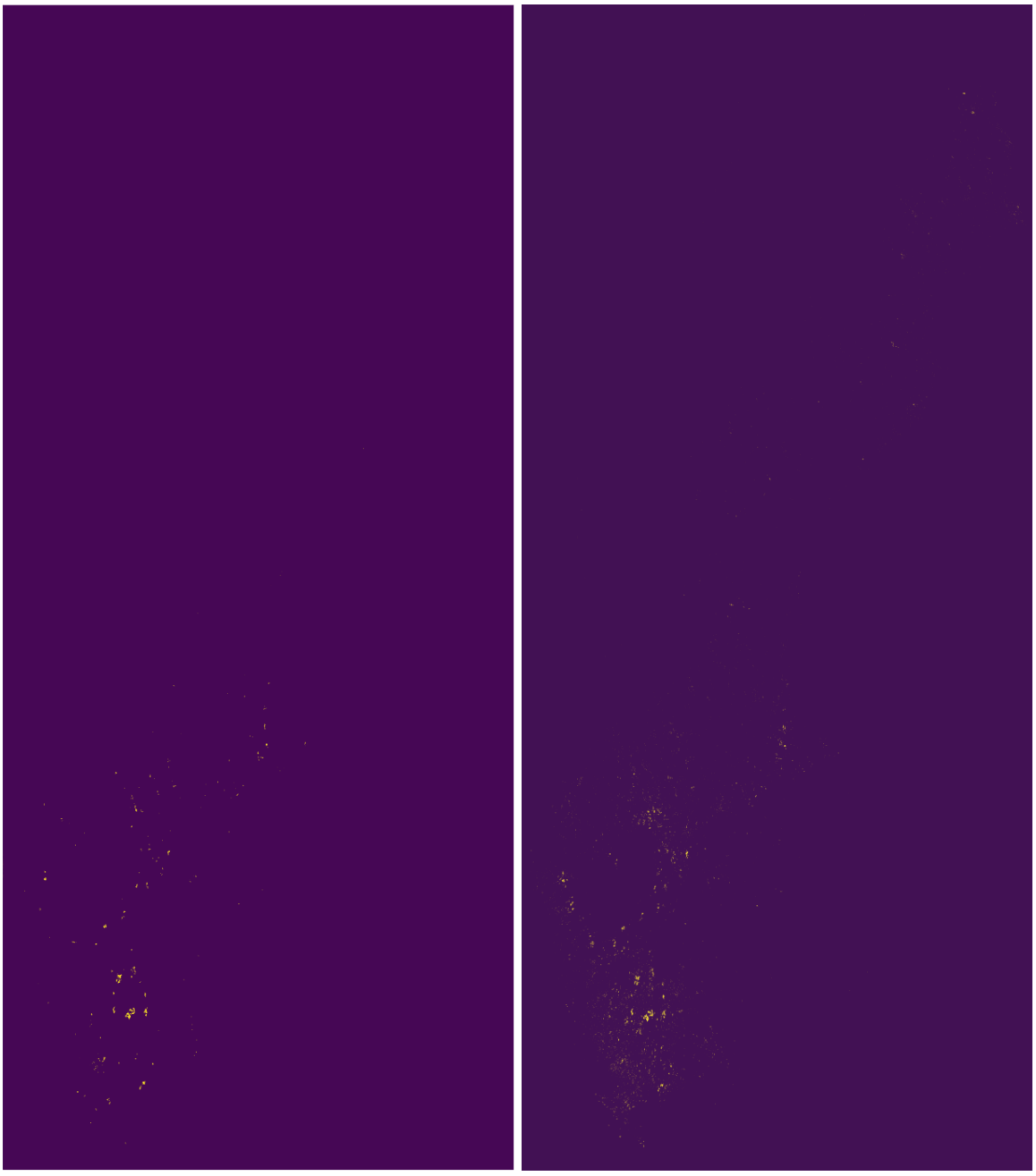

Figure 18: Left: Annotated högmosse in Sweden. It exists mostly in the southern parts of Sweden. Right: Predicted högmosse by a högmosse-only AI model, which during training has received training data also from the northern parts of Sweden which does not contain högmosse. The model predicts significantly less högmosse, as desired.

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[4] Aleksis Pirinen. *Reinforcement Learning for Active Visual Perception*. Lund University, 2021.