CNN sensitivity analysis for land cover map models using sparse and heterogeneous satellite data

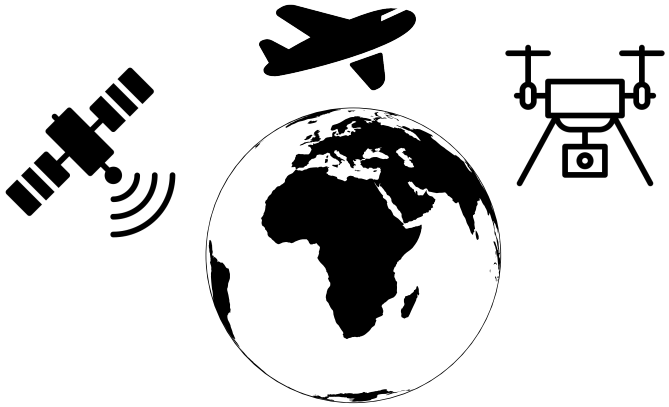
The Iberoamerican Congress on Pattern Recognition, 2024

Sebastian Moreno, Javier Lopatin, Diego Corvalán, Alejandra Bravo-Diaz.
Universidad Adolfo Ibáñez
Magister en Ciencias de la ingeniería
Doctorado en Data Science

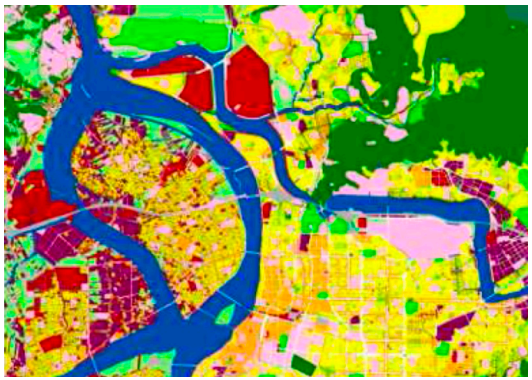
November, 2024

Context

- Currently, it is possible to **observe the Earth** with different **remote sensing** techniques such as satellites and drones.

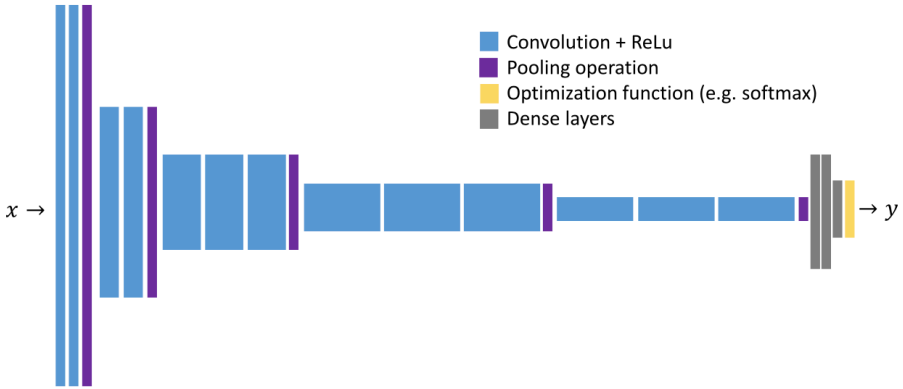


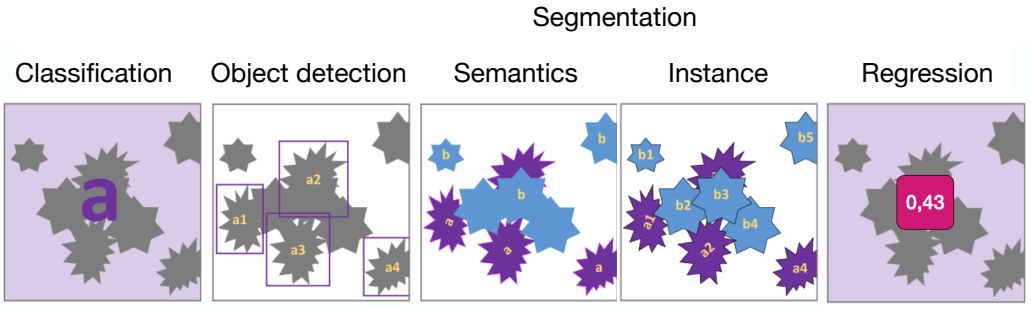
- **Land cover maps** are essential for characterizing the Earth's surface at a specific location and time.



Context

- **Convolutional neural networks** (CNNs) have transformed image processing by enabling models to learn patterns directly from images.
- **Segmentation** is one of the most common approaches for **land cover** analysis.
- **Pre-trained networks** are widely applied to address challenges in remote sensing.

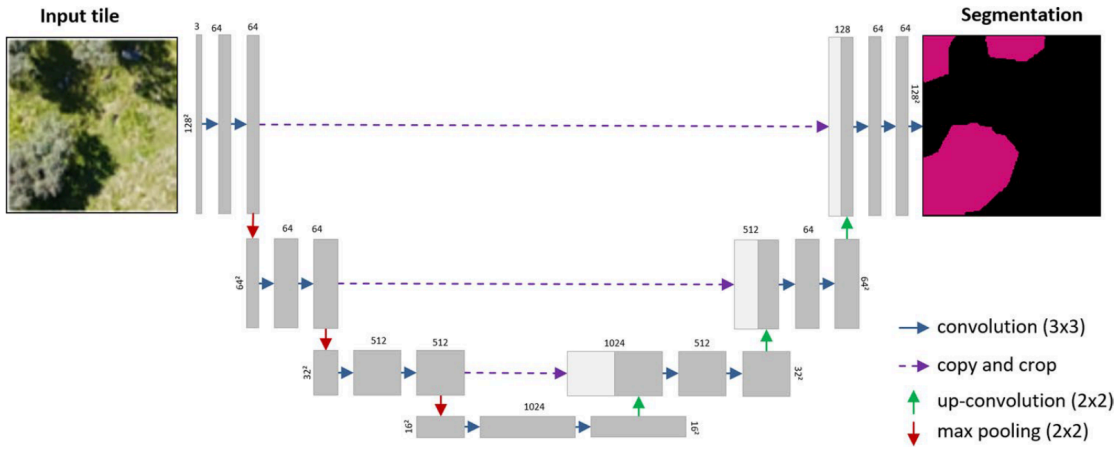




Problem

Is it feasible to use pre-trained models for land cover tasks using satellite data?

ResNet50 | Xception | InceptionV3 | Seresnet18 | EfficientNet



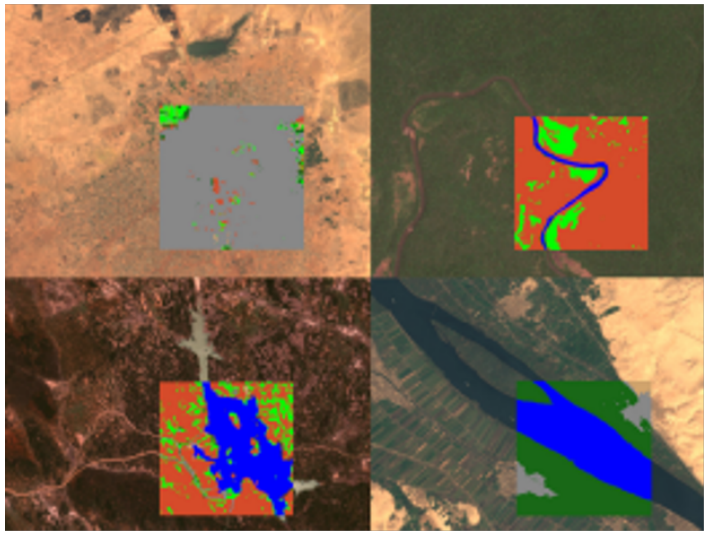
We analyze the segmentation capability of different models and architectures on the **LandCoverNet** dataset.

Methods - Data acquisition

- **LandCoverNet** dataset
- The dataset was reduced to 600 images of 256x256x3, divided among Africa, South America, and North America.

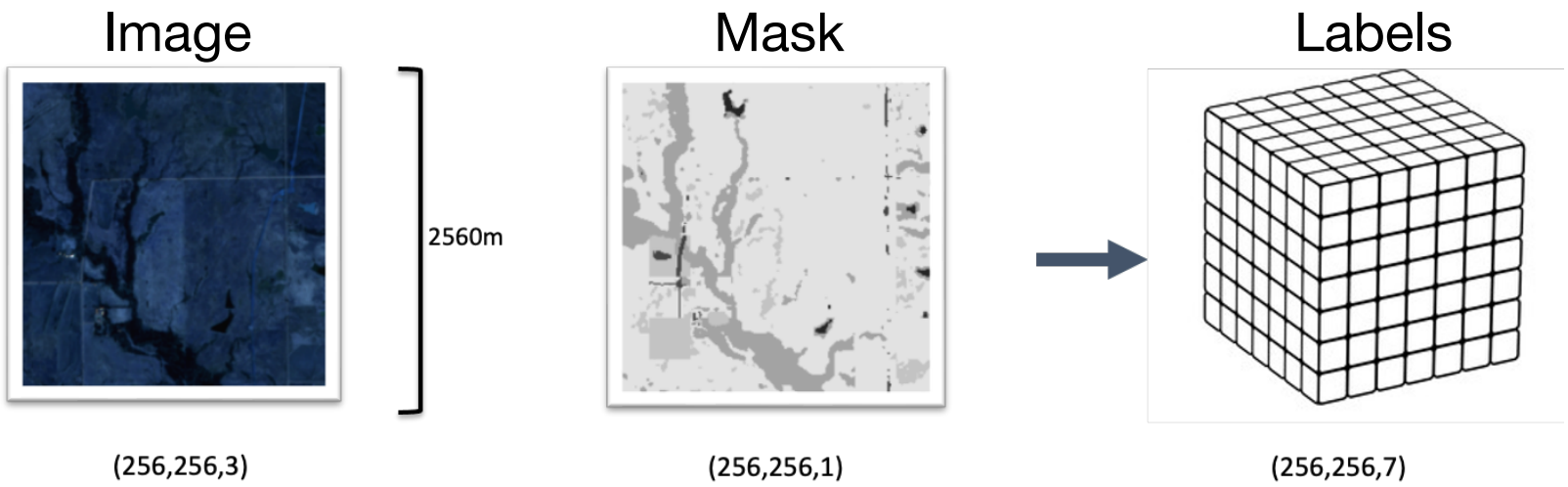
Class	%
• C1-Water	• 4.69%
• C2-Artificial bare soil	• 2.78%
• C3-Natural bare soil	• 10.19%
• C4-Snow/Ice	• 0.70%
• C5-Woody vegetation	• 24.07%
• C6- Cultivated vegetation	• 20.89%
• C7-Natural vegetation	• 36.68%

Radiant MLHub
EARTH IMAGERY FOR IMPACT

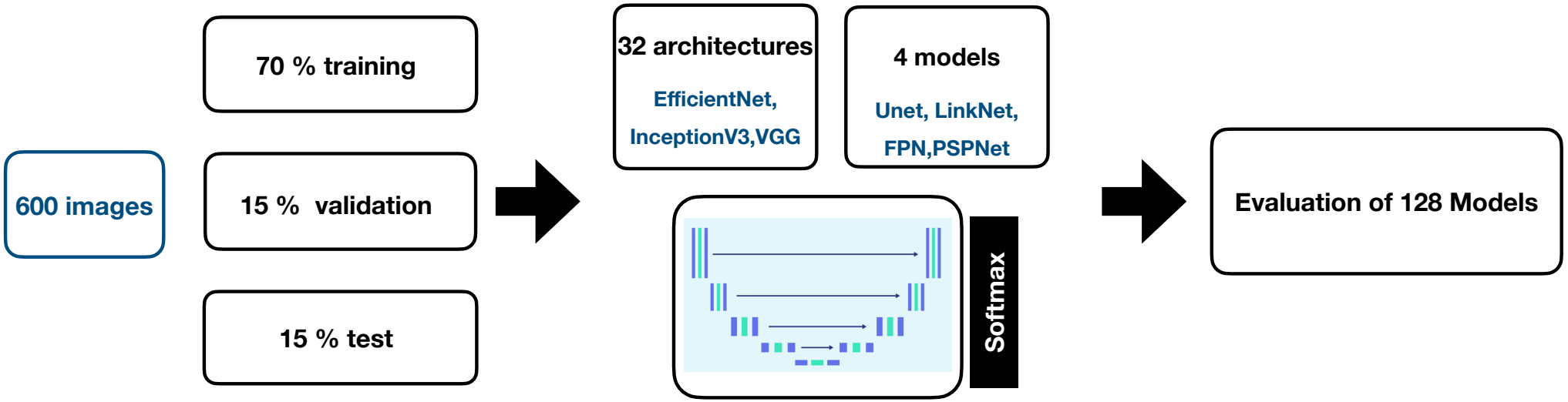


Methods - Image processing

- RGB channels were used.
- **Clouds** were **filtered** using the CLD channel.
- A total of **600 images** were selected (200 per continent).
- **One-hot encoding** was applied to the masks (for 7 classes).



Methods - Modeling

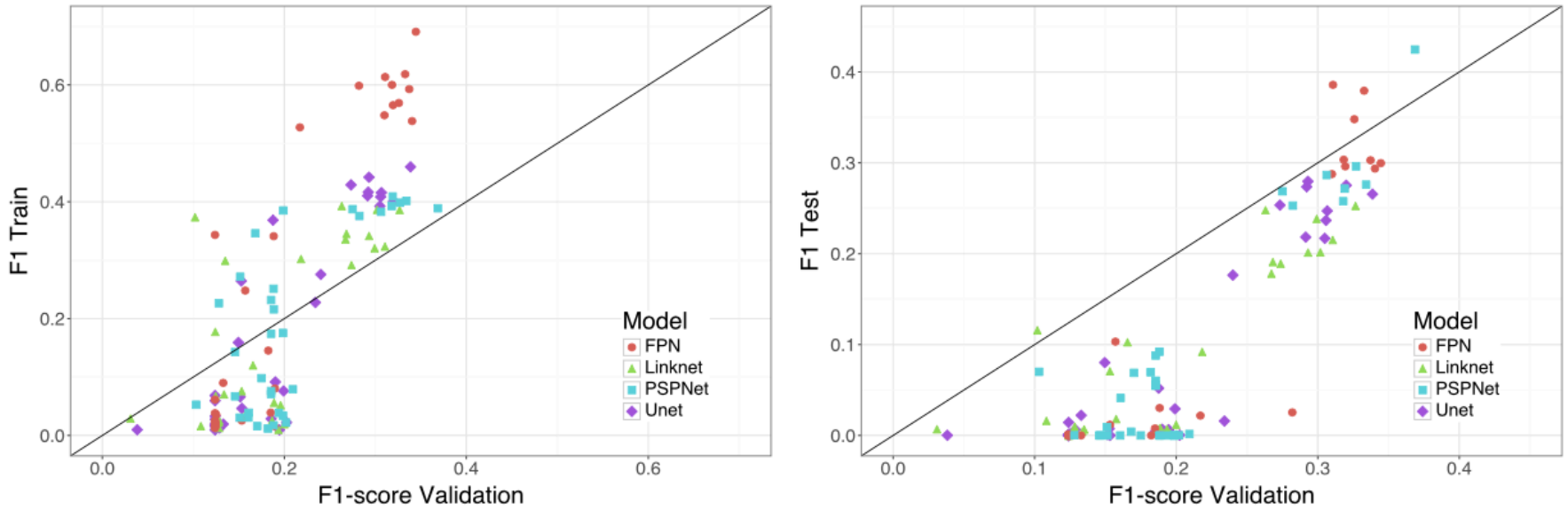


- **128 models** were evaluated for land cover segmentation.

  (4 models and 32 architectures per model).
- The **F1-score metric** was used to evaluate model performance.

Results - F1-scores

Several models from the FPN architecture exhibited overfitting, **PSPNet and LinkNet showed less overfitting**, with **PSPNet** performing the best.

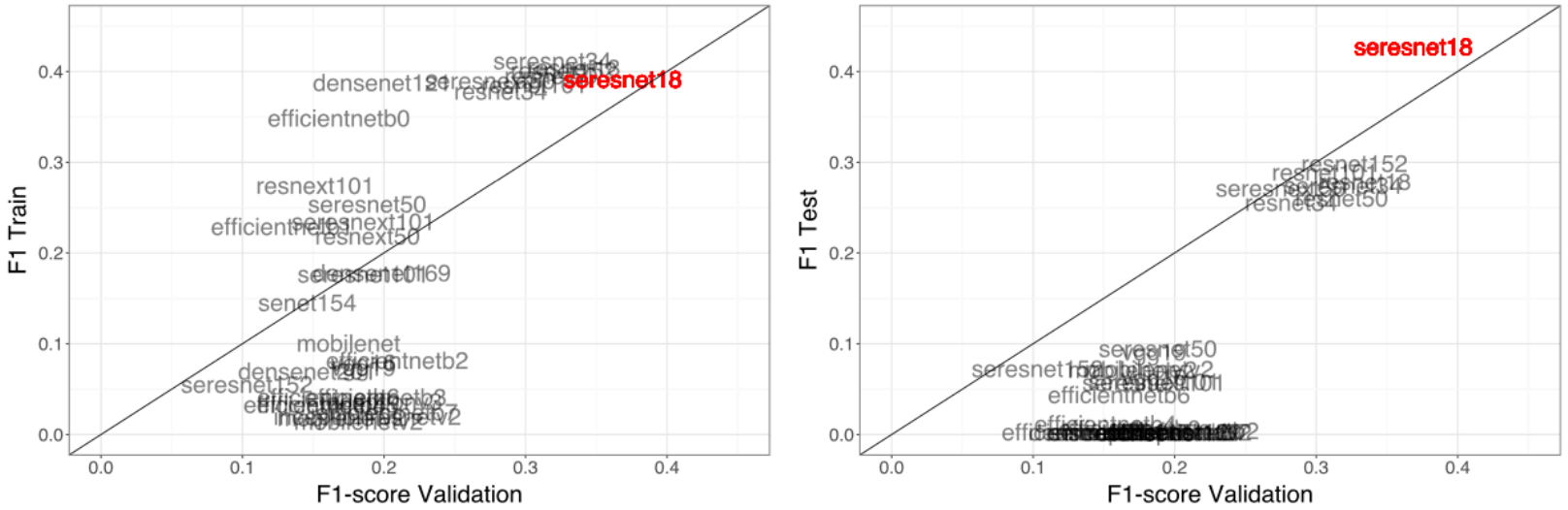


Comparing the **128 trained models**.

Left: Training versus validation data. Right: Test versus Validation data.

Results - F1-scores => PSPNet model

seresnet18 has the highest F1-score in the validation data with 0.3752, and close values for training (0.3889) and test (0.4247).



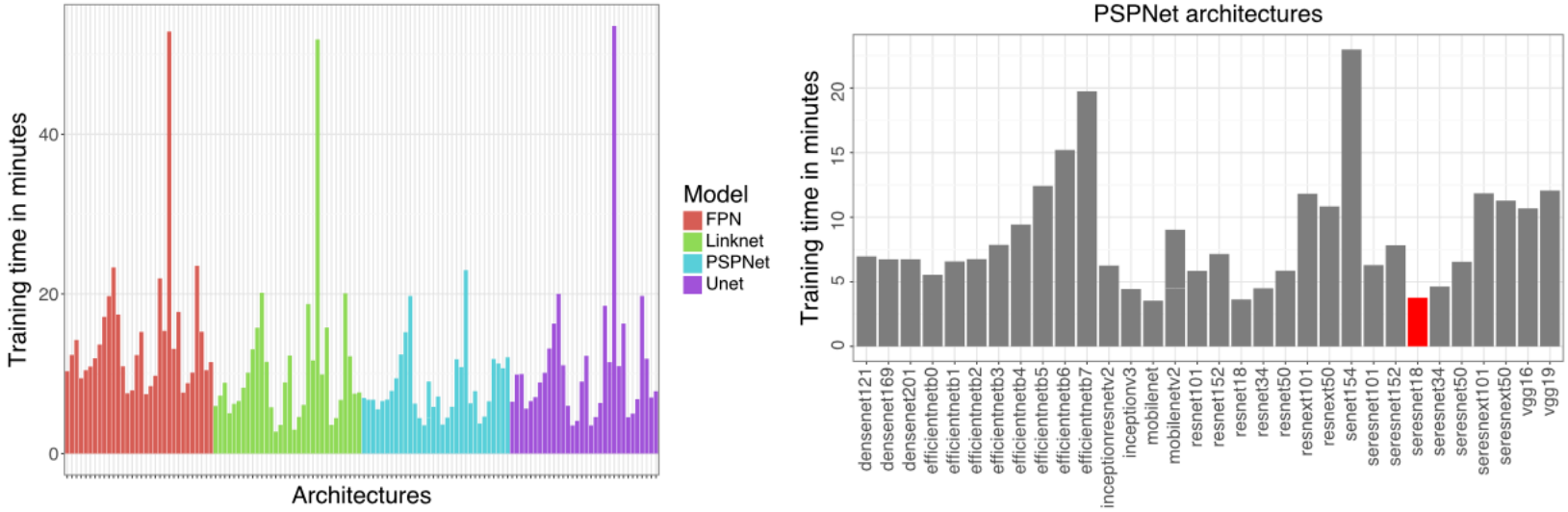
F1-scores results for the **32 architectures of the PSPNet model.**

Left: Training versus validation data. Right: Validation versus Test data.

Results - training times

Training time depends on the architectures rather than the model.

PSPNet-seresnet18 (red bar) is the fastest model and has the best F1-score.



Training times in minutes for the 128 models.

Left: Training time, we get rid of architecture names to avoid cluttering.

Right: training time of the 32 PSPNet architectures.

Results - fine tuning

The highest F1-score for validation/test was achieved with **80 unfrozen layers**.

Increasing the number of unfrozen layers could led to overfitting.

Unfrozen layers	Training time (mins)	F1-score Train	F1-score Validation	F1-score Test
0	3.701	0.3889	0.3752	0.4247
5	5.506	0.4944	0.4281	0.4266
10	5.194	0.5187	0.4439	0.4472
20	5.025	0.5391	0.4332	0.4303
40	4.683	0.5049	0.4607	0.4551
60	4.326	0.5366	0.4583	0.4550
80	**4.094**	**0.5253**	**0.4798**	**0.4786**
100	2.927	0.4752	0.4537	0.4541

Fine-tuning results for the **PSPNet-seresnet18**.

The experiment unfreezes the last encoder layers and gradually increasing the number of unfrozen layers up to 100 layers.

Results - data augmentation

Data augmentation increased the training time, and **improved the performance**.

As expected, fine-tuning and data augmentation obtained the best results.

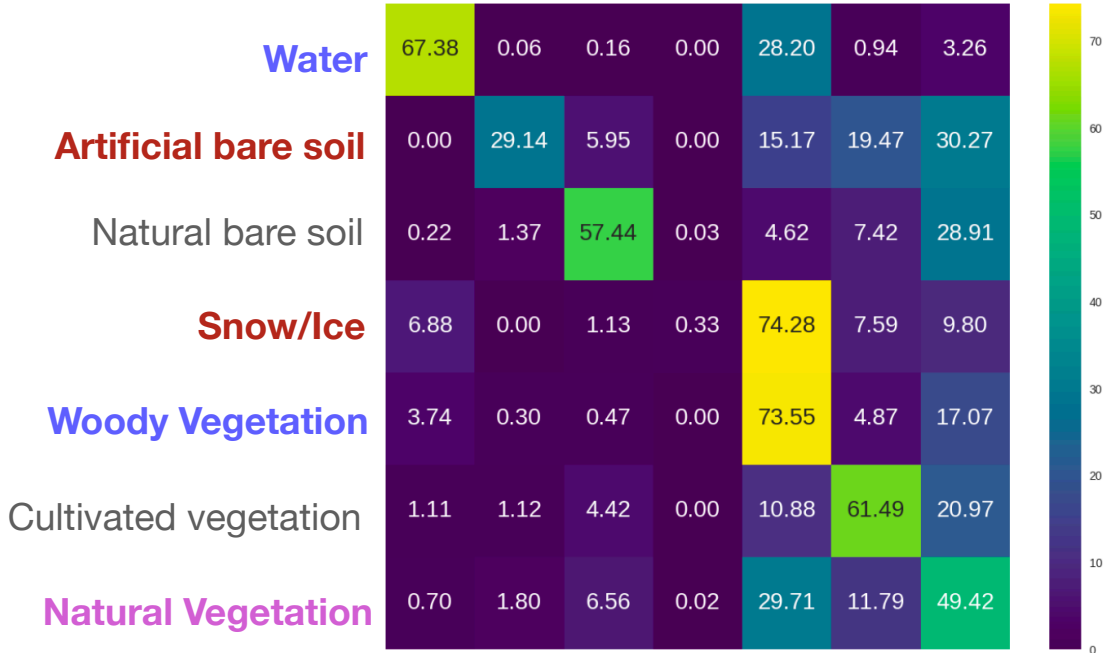
Model	Training time (mins)	F1-score Train	F1-score Validation	F1-score Test
Original	3.701	0.3889	0.3650	0.4247
Augmentation	9.920	0.3728	0.3552	0.4620
Fine-tuning	4.094	0.5253	0.4798	0.4786
Augmentation + Fine-tuning	**9.023**	**0.5314**	**0.5157**	**0.4950**

Data augmentation and fine-tuning results for the **PSPNet-seresnet18.**

The presented models are Original, Augmentation (Original + data augmentation), Fine-tuning (Original with fine-tuning), and Augmentation + Fine-tuning.

Results - confusion matrix

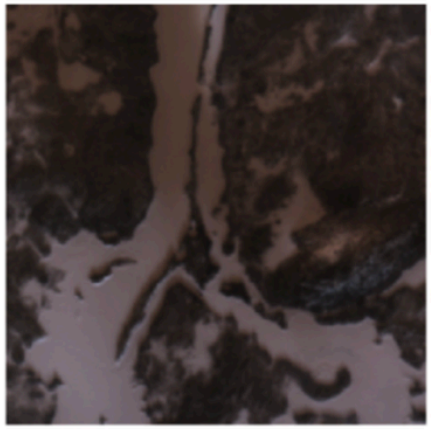
- "Water" and "Woody vegetation" have the best performances.
- Natural Vegetation is mainly confused with Woody vegetation.
- "Artificial bare soil" and "Snow/Ice" have few training data points (less than 3%) and show the worst performances.



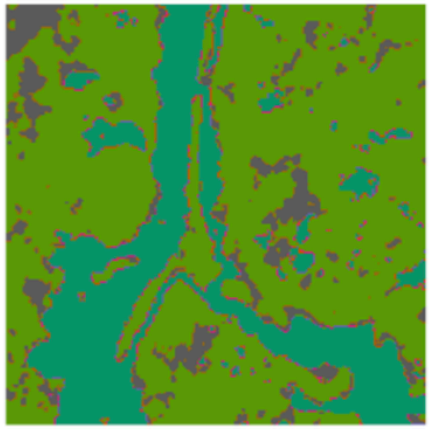
Confusion matrix for **PSPNet-seresnet18 (80 unfrozen layers** and **data augmentation)**.

Results - segmentation example

PSPNet-Seresnet18 **classifies correctly most of the water and snow/ice** (dark green and pink pixels), but **fails to classify the many small patches of natural vegetation** (gray pixels)



Original RGB image



Original mask

Final classification

PSPNet-Seresnet18

Conclusions

- **PSPNet-seresnet18** outperformed other models based on the F1-score.
- **Fine-tuning** and **data augmentation** improved the performance of the selected model (PSPNet-seresnet18).
- The confusion matrix revealed significant differences in the **performance across classes.**
- These results **are lower than other Sentinel-2-based land cover** analyses using semantic segmentation techniques.

Future work

- In remote sensing is common to use a model learned in a specific domain and applied in another domain.

- Given the poor performance obtained in the model, can we use a model learned from another domain and applied on another dataset?