# Abstract - FLAIR #1 Challenge

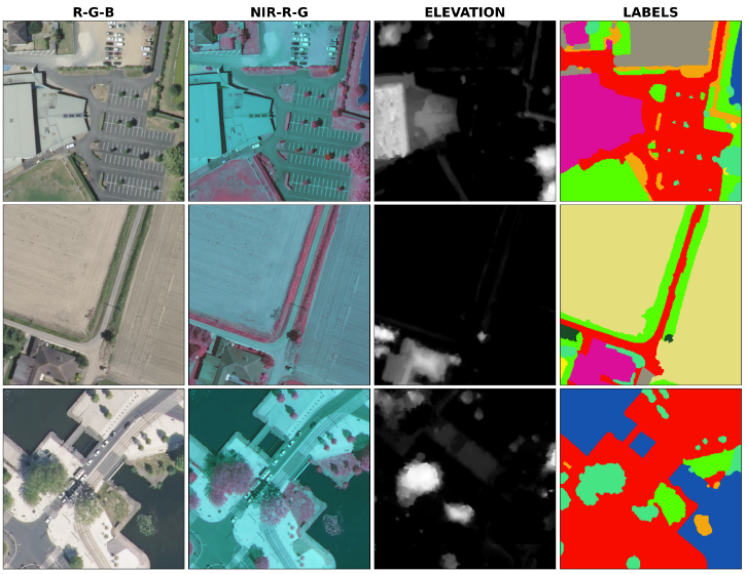
by Alan ENTEM & Vincent LARMET

---

**Context**

In order to map the occupation of the entire French territory, the IGN has compiled the FLAIR dataset (French Land cover from Aerospace ImageRy), one of the most important datasets on an international/European scale to train machine learning models to produce land cover maps.

FLAIR-one dataset consists of 77,412 high-resolution patches (0.2 m spatial resolution) with 13 semantic classes (19 original classes mapped out of 13). The dataset covers a total of approximately 800 km², with plots that have been sampled across continental France to illustrate the different climates and landscapes (spatial domains). The aerial images included in the dataset were acquired during different months and years (time domains).




Example of input data (first three columns) and corresponding supervision masks (last column).

**https://arxiv.org/pdf/2211.12979.pdf**

**AI task**

Image segmentation divides an image into segments where each pixel in the image is mapped to an object. This task has several variants such as instance segmentation, panoptic segmentation and semantic segmentation.

Semantic segmentation is the task of segmenting parts of an image that belong to the same class. Semantic segmentation models make predictions for each pixel and return class probabilities for each pixel. These models are evaluated on the mean intersection over the union (Mean IoU).

## Transfer learning, fine tuning & benchmark des modèles

### Semantic Segmentation & Transformers

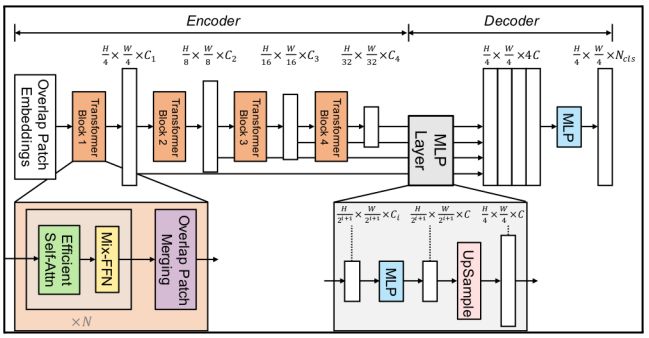
Since semantic segmentation is a type of classification, the network architectures used for image classification and semantic segmentation are very similar. In 2014,a founding article by Long et al. used convolutional neural networks for semantic segmentation. More recently, transformers have been used for image classification (e.g.YEAR ), and now they are also used for semantic segmentation, pushing the state of the art further.

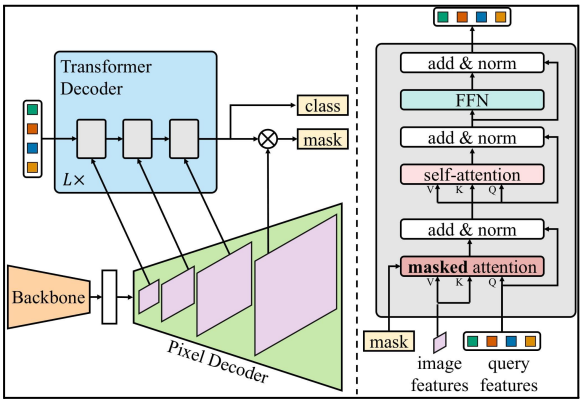
### Fine tuning

A pre-trained model is a machine learning (ML) model that has been trained on a large data set and can be fine-tuned for a specific task. Pre-trained models are often used as a starting point for ML model development, as they provide a set of initial weights and biases that can be refined for a specific task.

### Segformer

is a semantic segmentation model introduced by Xie et al. in 2021. It features a hierarchical Transformer encoder that does not use positional encodings (unlike ViT) and a simple multi-layer perceptron decoder. SegFormer achieves peak performance on several common datasets.
https://github.com/NVlabs/SegFormer




### Mask2former

We have moved from separate architectures to what researchers now call universal image segmentation architectures, capable of solving any image segmentation task. Interestingly, these universal models all adopt the “mask classification” paradigm, discarding the “pixel classification” paradigm entirely. A figure illustrating the architecture of Mask2Former is shown below (taken from article original ).
https://github.com/facebookresearch/Mask2Former

An image is first sent through a backbone (which in the article could beResNet orSwin Transformer ) to get a list of low-resolution feature maps. Then these feature maps are enhanced using a pixel decoding module to achieve high resolution features. Finally, a Transformer decoder takes a set of queries and transforms them into a set of binary masks and class predictions, conditioned by the capabilities of the pixel decoder.

### Convnets

Convolutional neural networks, mainly of UNET encoder-decoder architecture, were also tested during the challenge. These models give significantly worse results than transformers, developed more recently in the literature. However, the Convnext encoder gives results similar to transformers. Finally, two “traditional” Convnets were used in the final submission: UNET++ (encoder: convnext Tiny) and UNET (encoder: efficientnet v2 Small).

### Learning strategy

Here are the main learning parameters and some notable observations:

- **Weighted cross entropy loss** (weight 0 to class 13)
- batch size set from **2 to 8** depending model size and hardware constraints
- **a reduced validation set** (and therefore a bigger training set) allows a better generalization on the test set. Our best models were trained with validation sets of 600 to 1000 images. This may seem counter-intuitive regarding the risk of overfitting.
- **Data augmentation** (geometric, brightness, contrast) marginally increases performance. Convergence, on the other hand, is less stable and requires much more epochs.
- **Test-Time Augmentation** (3 rotations) significantly improves the performance of Convnets, and marginally those of transformers.
- **RGB images** yield equivalent (or even better) results than 5-channel images. It would be necessary to deepen by looking at the results by class, we can expect that the near infra-red and the elevation improve vegetation classes and buildings, respectively.
- The use of **BigEarthNet** dataset pre-trained weights gives poor results and a chaotic learning curve. It would be interesting to investigate this issue because of the similar nature of images.

### Results of the experiments

All training notebooks can be found here and the weights of the different driven models here. Here are the results by individual models:

| ID | models | channels | batch | epochs | test IoU | GPU |
|---|---|---|---|---|---|---|
| 1 | segformer b0 ** | rgb | 8 | 8 | 59.9 | T4 |
| 2 | segformer b5 ** | rgb | 8 | 4 | 61.5 | V100 |
| 3 | segformer b5 ** | rgb,nir,dsm | 8 | 3 | 61.2 | V100 |
| 4 | segformer b0 ** | rgb,nir,dsm | 8 | 4 | 59.9 | T4 |
| 5 | mask2former-swin-large-ade ** | rgb | 8 | 4 | 61.5 | V100 |
| 6 | mask2former-swin-base-ade-INK 21** | rgb | 2 | 4 | 61.3 | T4 |
| 7 | mask2former-swin-large-coco * | rgb | 2 | 5 | 59.36 | RTX 3060 |
| 8 | unet++ convnext_tiny * | rgb,nir,dsm | 4 | 7 | 59.9 | RTX 3060 |
| 9 | unet efficientnetV2_S* | rgb,nir,dsm | 8 | 15 | 59 | RTX 3060 |
| 10 | segformer b2** | rgb,nir,dsm | 4 | 3 | 59.2 | T4 |
| 11 | segformer b0 * | rgb,nir,dsm | 8 | 14 | 60 | RTX 3060 |
| 12 | segformer b1 * | rgb,nir,dsm | 3 | 18 | 59.3 | RTX 3060 |
| 13 | segformer b0 * | rgb | 8 | 39 | 60.1 | RTX 3060 |
| 14 | segformer b1 * | rgb | 3 | 34 | 61 | RTX 3060 |
| 15 | segformer b2* | rgb | 3 | 28 | 61.3 | RTX 3060 |
| 16 | segformer b3 * | rgb | 3 | 17 | 61.3 | RTX 3060 |
| 17 | segformer b4 * | rgb | 2 | 15 | 61.4 | RTX 3060 |

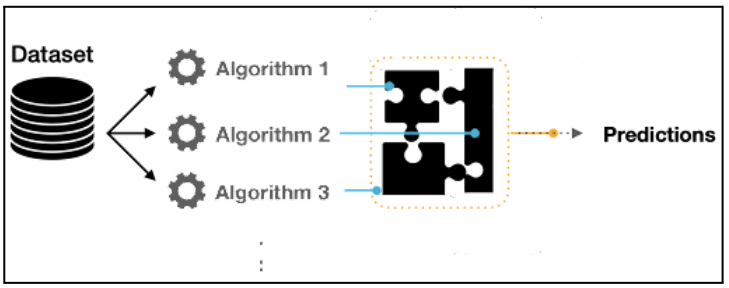
* models trained by Vincent Larmet

** models trained by Alan Entem

# Inference

## Ensembling

In statistics and machine learning , ensemble methods use several learning algorithms to obtain better predictive performance than those that could be obtained from any of the constituent learning algorithms alone.




Empirically, ensembles tend to perform better when there is significant diversity between models. Many ensemble methods therefore seek to promote diversity among the models they combine.
In practice, we have done the **averaged probabilities per pixel** of several models.

## Zone reconstruction

Using georeferencing and projection information from each image, we can easily create VRTs (virtual rasters) of the test set areas. The test set is made up of 193 large areas.

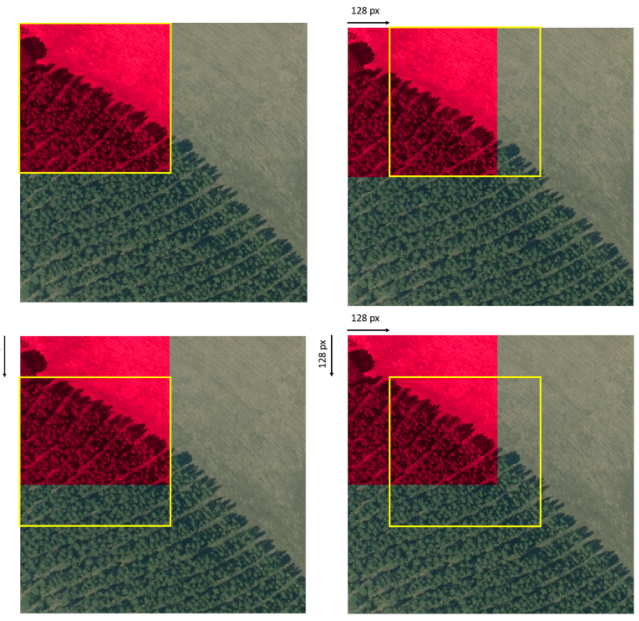
The reconstructed D012_2019/Z2_UF zone. In red, an original patch from the test set(512*512 pixels)

We predict each area multiple times with a 128 pixel shift on x and y. In this way, we get 4*4=16 predictions for each pixel. The final prediction of each pixel is the majority class.

In red, an original 512*512 patch from the test set. In yellow, the predicted patches shifted by 128 pixels.

Finally, we disaggregated the final predictions into original 512*512 patches.

This technique makes it possible to improve the predictions at the edge of the original patches, because due to the architecture of the learning models, the predictions are generally better at the center of the images.
Another method exists by using weights according to pixel position within a patch (larger weight in center), then predictions are aggregated through a weighted mean of probabilities. But this technique requires much more storage space and IO operations.(see https://github.com/Vooban/Smoothly-Blend-Image-Patches).




etc...

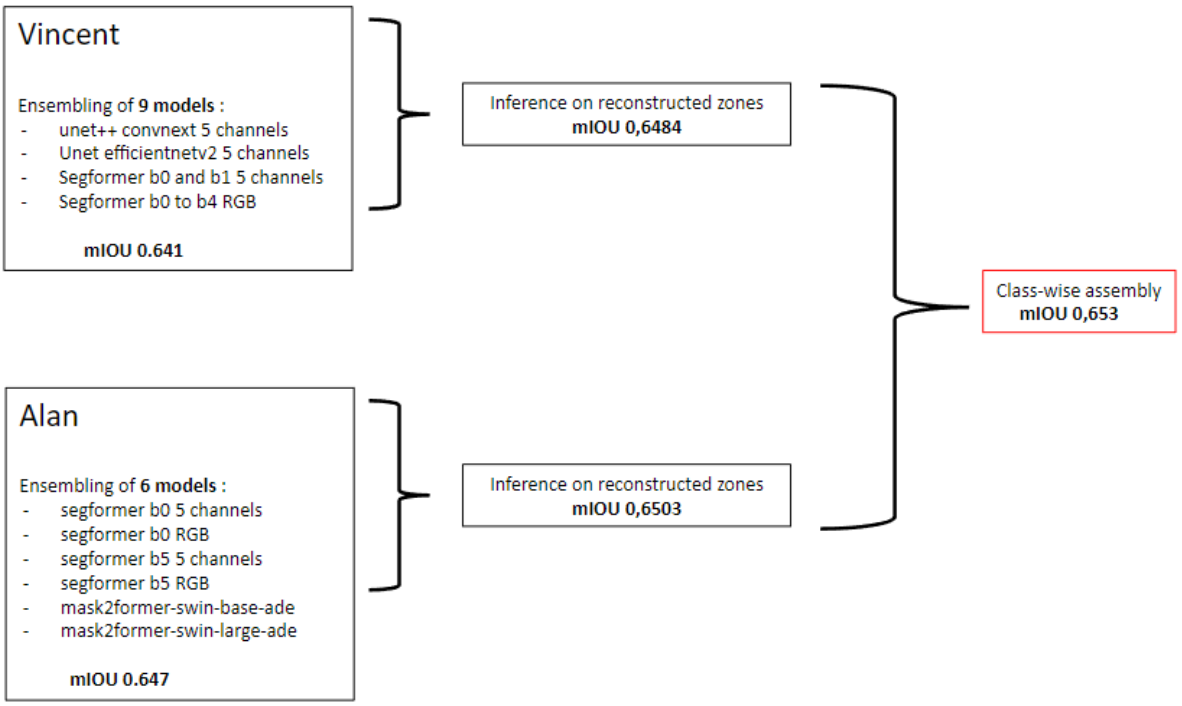
## Ensembling results

| ID | Meta | Patch rebuilding | test IoU | tps (min) | GPU |
|---|---|---|---|---|---|
| A | 1 + 4 | No | 62,2 | 100 | T4 |
| B | 1 + 4 | Yes | 63,3 | 480 | T4 |
| C | 2 + 3 | No | 62 | 120 | T4 |
| D | 1 + 2 + 3 + 4 | No | 63,52 | 240 | T4 |
| AND | 1 + 2 +3 + 4 | Yes | 64,4 | 960 | T4 |
| F | 1 + 2 + 3 + 4 + 5 | No | 64,4 | 240 | T4 |
| G | 1 + 2 + 3 + 4 + 5 | Yes | 64,8 | 1440 | T4 |
| H | 1 + 4 + 6 | Yes | 64,37 | 720 | T4 |
| I | 1 + 2 + 3 + 4 + 5 + 6 | No | 64,7 | 300 | T4 |
| J | 1 + 2 + 3 + 4 + 5 + 6 | Yes | 65,03 | 2040 | T4 |

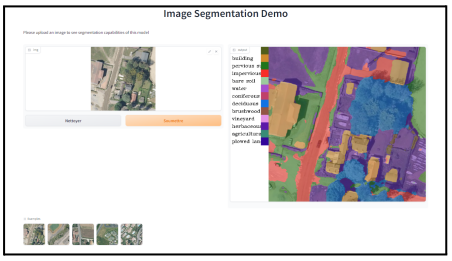
## Class assembly

Finally, we used the class-disaggregated results from the test set to collate the submissions to get the best mIOU. This step is naturally inherent to the nature of the challenge, and does not seem relevant in terms of generalization and added value for IGN.

## Our final submission



## Demo application




Demonstrating a Machine Learning project is easy withBuilt & Hugging Face:

https://huggingface.co/spaces/alanoix/ign-flair-one

## Development suggestions

Pre-training of an encoder on a task of denoising aerial images on 3 and/or 5 channels. When pre-trained on large amounts of data and transferred to several medium or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) achieves excellent results compared to state of the art convolutional networks while requiring far fewer computational resources for training.https://arxiv.org/abs/2010.11929

The idea is to pre-train a ViT from scratch for modeling hidden images on the full IGN aerial image dataset.

The list of models to test below is non-exhaustive:

- VIT-adapter,
- BEIT,
- swinTransformer V2,
- DEIT,
- …

## Sources

- https://huggingface.co/blog/fine-tune-segformer
- https://huggingface.co/blog/mask2former
- https://huggingface.co/blog/gradio-spaces
- https://deepai.org/publication/training-a-vision-transformer-from-scratch-in-less-than-24-hours-with-1-gpu
- https://github.com/huggingface/transformers/tree/main/examples/pytorch/image-pretraining
- VERY HIGH RESOLUTION LAND COVER MAPPING OF URBAN AREAS AT GLOBAL SCALE WITH CONVOLUTIONAL NEURAL NETWORKS