Rapport Computer Vision - Wildfire Prediction

Self-Supervised & Unsupervised Learning

ALLEMAND Fabien (Télécom SudParis), LAMURRI Pierre (ENSTA), PETITCOLIN Corentin (ENSTA), RIGAUD Charles (ENSTA)

24 février 2025

1 Introduction

Classical approaches for image classification relies on deep neural networks trained on large datasets of images and labels, a technique called Supervised Learning (SL). Eventhough these methods have proven their worth for many image classification tasks, their supervised nature prevent them from being used on tasks where there is few labelled training datas. Image labelisation is often time consuming and costly. For instance, simple tasks like object recognition require a lot of training samples to account for the variety of objects. Such data extensive methods sometimes requires experts to labelize data. It should also be noted that in both cases labelisation can be incorrect and negatively impact the training of the network. This is without mentioning the ethical issues hidden behind low income labelisation labour.

In this work, we try to reduce the use of labelised data by exploring unsupervised learning and Self-Supervised Learning (SSL) techniques that can leverage hidden knowledge contained in images without labels. The goal of this study is to assess the performance gain of SSL on downstream tasks that have large amount of unlabeled data available but few labeled data. We apply this technique to wildfire detection through satellite images as analysing satellite images is a domain where SL would require experts to annotate large databases of images to the best of their ability. A process that end up being costly, time consuming and unreliable. First, we introduce the dataset and do a quick overview of self-supervised techniques. Then, we experiment four different methods based on unsupervised and self-supervised learning, namely Gaussian Mixture Models (GMM), Auto-Encoders (AE), Variational Auto-Encoder (VAE) and the SimCLR SSL framework.

1.1 Wildfire Prediction Dataset

Throughout this study, we use the Wildfire Prediction Dataset (Satellite Images) [1]. This dataset contains 42850 satellite images of areas that previously experienced wildfires and areas that did not. These images are 350×350 pixels color images split in two classes : wildfire and no wild fire. Examples of images can be seen on Figure 1. This dataset is split into three subsets for training, validation and testing. Table 1 show that classes are balanced in all splits.

Note : This dataset contains labels for all images. However, in this study, we do not use the labels of the train split.

Class	Train split	Validation split	Test split	Total
Wildfire	15750	3480	3479	22709
No Wildfire	14499	2820	2820	22709
Total	30249	6300	6299	

Table 1 – Number of samples in each class of the dataset. Two corrupted images had to be removed from the dataset.

1. https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset/data

FIGURE 1 – Visualisation of images from the Wildfire Prediction Dataset.

1.2 Unsupervised Learning

Unsupervised learning is a machine learning paradigm where the model is trained solely on unlabeled data, aiming to uncover hidden patterns and intrinsic structures within the dataset. This technique relies entirely on the data's inherent properties. This approach can include clustering methods, dimensionality reduction, or generative models that learn the unknown distribution. Used in **Method 1**.

1.3 Self-supervised Learning

Self-supervised learning is a machine learning paradigm where the model is trained in two steps. First, there is an auxiliary or pretext task whose goal is to learn insights and capture essential features of the data itself. This task does not require labels and can be seen as initialising the weights of the neural networks in a way that suits the data. Next, the downstream task can be performed, in our case using SL with labelised data from the validation set.

There are different types of SSL. In this work we use :

— Autoassociative : The auxiliary task consists in reconstructing the input data. **Method 2, 3, 4**.

— Contrastive : The goal of the pretext task is to minimize the distance between data that are known to be similar and maximise the distance between those that are different. **Method 4**.

2 Method 1 : Gaussian Mixture Model

We tried using a GMM to fit a multivariate Gaussian distribution on the features extracted from the unlabeled train set with a pretrained *VGG16* model. For unsupervised learning, this method can leverage the ability of the model to identify features in the images at different levels and group them by the similarity of their features.

For our experiment with this method, we force the model to fit a bivariate Gaussian distribution on the test set. This produced satisfactory results which is why we kept it this way. However this could have led to misrepresentation of the set, either identifying irrelevant classes for the final task (eg : town

or forest instead of wildfire or no wildfire), or failing to identify clear differences between images and predicting a uniform distribution.
An approach that could help fix this type of issues would have been to add more components to the multivariate Gaussian (eg : 10 instead of 2) and use the computed responses from each component for the input image as input for a small multi-layer perception classifier we could train on the smaller, labeled dataset.

2.1 Results

Using this method, we obtained an accuracy of 89 % on the test set with more false positives than false negatives for the presence of wildfires (571 false positives and 213 false negatives over the 6300 samples.)

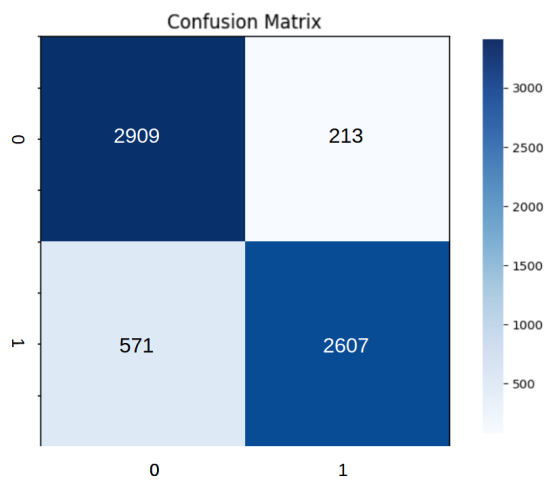


FIGURE 2 – GMM Confusion Matrix on the validation and test sets

This method managed to produce good classification over the distribution. However, it does not fully leverage the learned weights of a pretrained mode.

2.2 Consideration

We see that using pretrained model with unsupervised learning like GMM can have satisfactory results but we think we could leverage more the feature extraction capabilities of pre-trained models

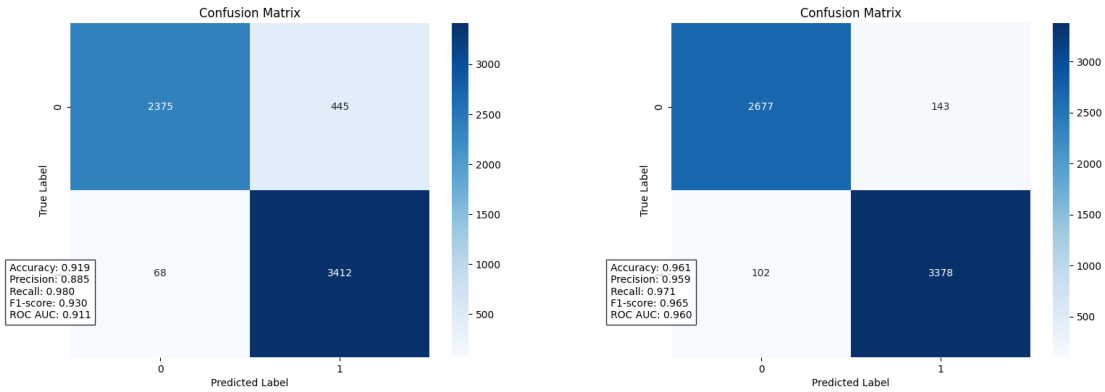
3 <u>Method 2</u> : $VGG16$ based auto-encoder

With this method we trained a pretrained-model $VGG16$ for an autoencoding task to try to create even better representations of the dataset images.
After the autoencoding part we extract and freeze the encoder. Then we plug it on 2 layers classification head. Which will be trained on the validation dataset to perform the classification task using Cross Entropy.
We want to measure the increase in accuracy of using the training data on the auto-encoder compared to the original $VGG16$.

3.1 Results

We see in figure 3 that both models performs well with an accuracy of more than 90 % every time. We also see that the $VGG16$ Only model performs better overall with a better accuracy but is more prone to false negative. We were not expecting these results as we expected the model trained with the training data to be better.

We tried modifying the hyperparameters but the results were the same. We think that the acting on the weights of the pretrained network with the autoencoding task could make it worse.



(a) Auto-Encoder on the test set

(b) VGG16 Only on the test set

FIGURE 3 – Comparison of confusion matrices

3.2 Consideration

Maybe training with this method on a pretrained model "downgrades" its understanding of certain information. We will try another self-supervised method trained from scratch to see if it can match or outperform the accuracy of the off-the-shelf $VGG16$ encoder (with limited data).

4 <u>Method 3</u> : Variational Auto-Encoder (VAE)

We train a variational autoencoder from scratch on the training set to try another way to create a latent space that could potentially leverage more of the structure of the image.
We will do a reconstruction task on the original image.

4.1 Results

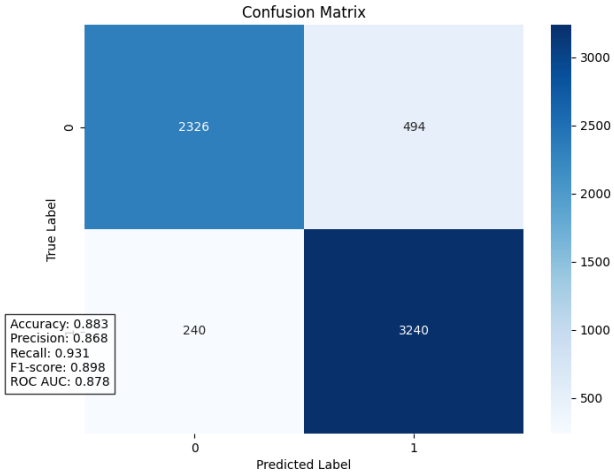


FIGURE 4 – VAE Confusion Matrix on the test set

4.2 Consideration

At first sight VAE results looks strong, but not better than method 2. We tried multiple LIME inference of our model to check its classification criterions. We see in Figure 5 that he model seems to focus on the top part of images to make its predictions. This disqualify the model and raises questions about potential biases in our dataset, why would our model focus that much on the top otherwise ?

This was a good way to check the difficulty of training a model from scratch versus the generalization capacities of off-the-shelf models.

The last type of model we have not tried yet are contrastive SSL trained models. Let's check if they can leverage better performance, especially, we would like to check that they focus on pertinent part of the images and can reach at least VAE accuracy performance.

We will also try to change of pretrained model architecture to try to beat the $VGG16$ of Method 2.

FIGURE 5 – Examples of LIME Inference

5 Method 4 : SimCLR

This method is based on the the SimCLR framework presented *A Simple Framework for Contrastive Learning of Visual Representations*[2]. This framework applies contrastive SSL to images. The goal of the pretext class is thus to minimize the distance between similar images on the latent space, while maximizing the distance between different images. Figure 6, represents the framework for the pretext task. For each batch of data, each image x undergoes two different transformations in order to create a positive pair. The goal of the neural network is to project these two transformed images close together in the latent space but far from all other images in the batch (considered as negative pairs).

2. https://arxiv.org/abs/2002.05709

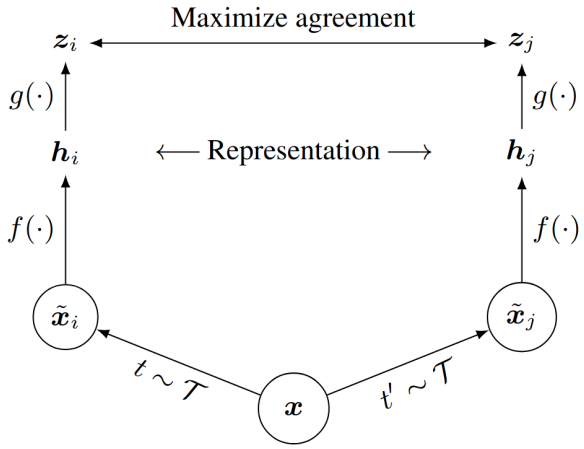


FIGURE 6 – SimCLR framework.

We tried using different portions of the validation split for training our neural network on the downstream task, in our case the image classification.

As recommended when using the SimCLR framework, the transformation to create positive pairs consists of randomized resized crop, randomized flip (horizontal and vertical), randomized color distortion and randomized Gaussian blur. Examples of images and their transformation are presented in Figure 7.

FIGURE 7 – Visualisation of images from the Wildfire Prediction Dataset before (top) and after (bottom) transformation.

For the pretext task, we use a $ResNet-18$ with two dense layers with ReLU activation function to

project data in a 128 dimensions latent space. After the 100 epochs of training are complete, we freeze the encoder and change the head on the neural network (three dense layers with ReLU activation) to train the neural network for binary classification during 50 epochs.

We also perform some other autoassociative SSL experiments with a $UNet$ neural network to see if we can find differences with the $VGG16$ one. The $UNet$ is trained for 100 epochs to reconstruct images. We then keep only the encoder and add a three layers head to perform binary classification and train again for 50 epochs with the encoder frozen.

We compare our downstream neural networks with similar ones trained only for the downstream classification task. That is to say neural networks that were not trained with any auxiliary task but only for classification with the same amount of data as our downstream neural networks. However, we use a pre-trained $ResNet - 18$ encoder.

The goal of this study on the last method is to assess the performance gain of SSL on downstream tasks that have large amount of unlabeled data available but few labeled data, in our case, satellite images to predict past wildfires. Keeping the same encoder trained on the auxiliary task, we train different models for the downstream classification using different sizes of datasets. This way, we can evaluate to what extend large amount of unlabeled data can compensate for a lack of labeled data.

5.1 Results

First and foremost, Figures 8 and 9 show that all models achieve reasonably good performance on the classification task. The accuracy is always above 90 %. As expected, the amount of data used for supervised training impact performance. For both $ResNet - 18$ and $UNet$ based classifiers, the accuracy and F1-score increase with the number of training samples.

Interestingly but unsurprisingly, the models trained with SSL are more efficient than their supervised counter parts. In Figures 8 and 9, self-supervised models are consistantly above supervised models in both accuracy and F1-score. What is definitly interesting is how this performance gap evolves with the number of training samples. The less training samples the larger the gap. This means that self-supervised trained neural networks benefit from the auxiliary training and are less impacted by the lack of labelised data.

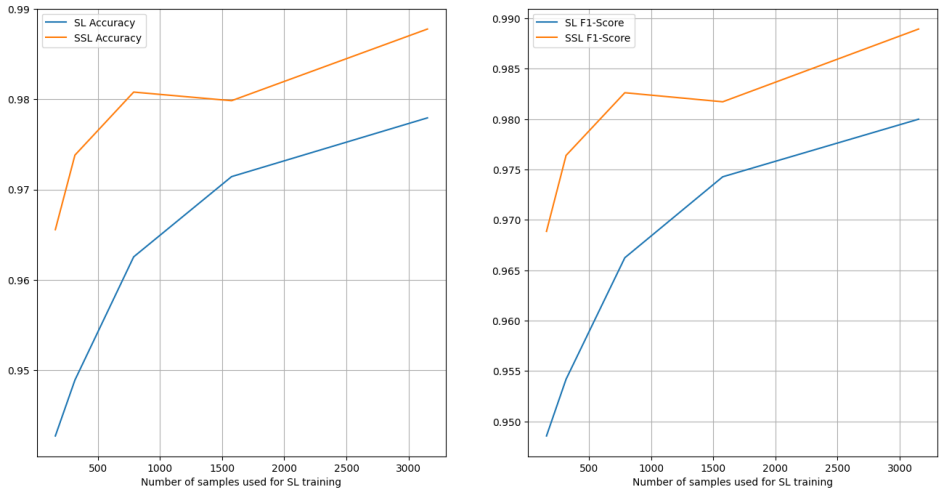


FIGURE 8 – ResNet based classifier classification performance based on training samples.

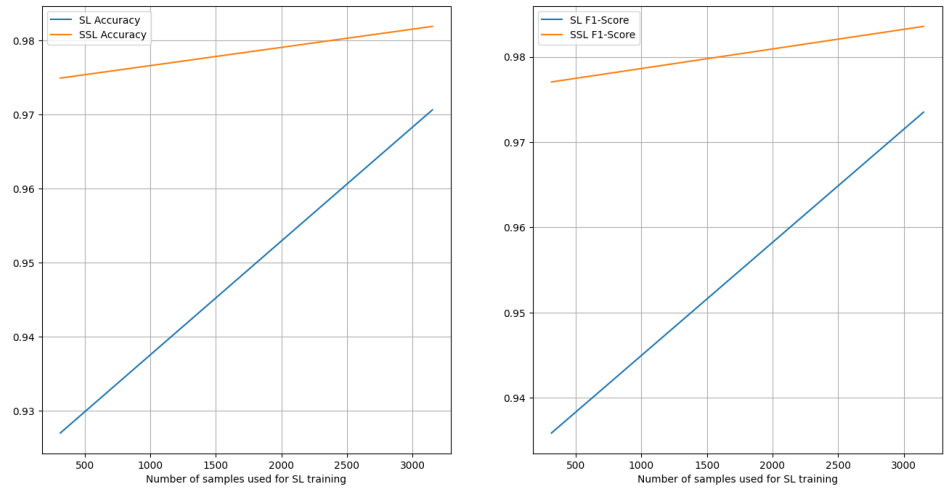
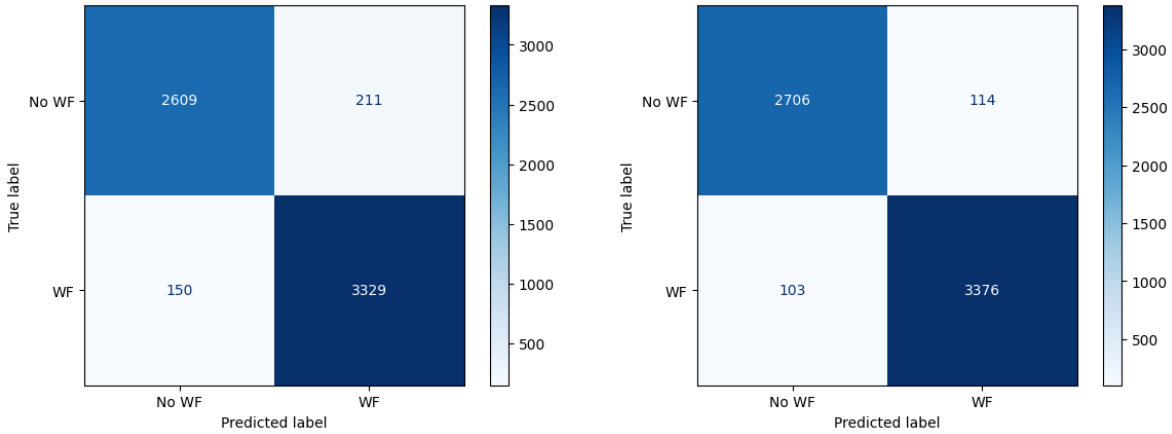


FIGURE 9 – UNet based classifier classification performance based on training samples.

Let's imagine we worked on a different task such as predicting current wildfires through real time satellites images. With wildfire prediction, false positive but mostly false negative can have important consequences. The former can lead to wasting firefighters resources, the latter to disastrous losses. Figure 10 show that with only 160 labeled training samples, both SL and SSL models perform similarly. There is no model that is clearly better in one way or another (preventing useless actions or avoiding fires).

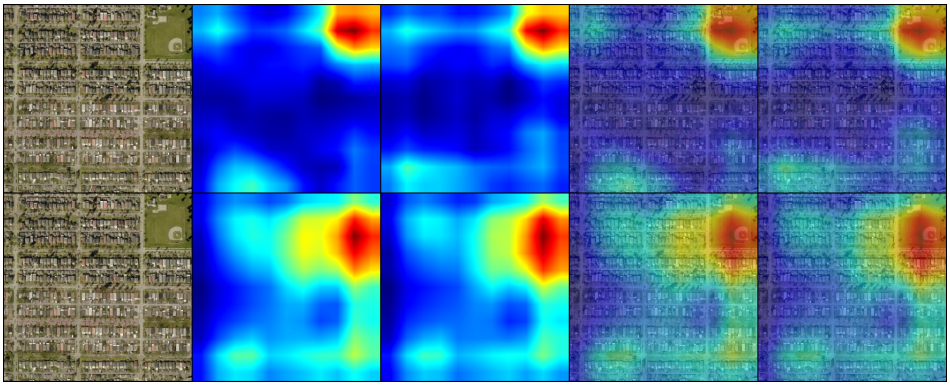


(a) SL

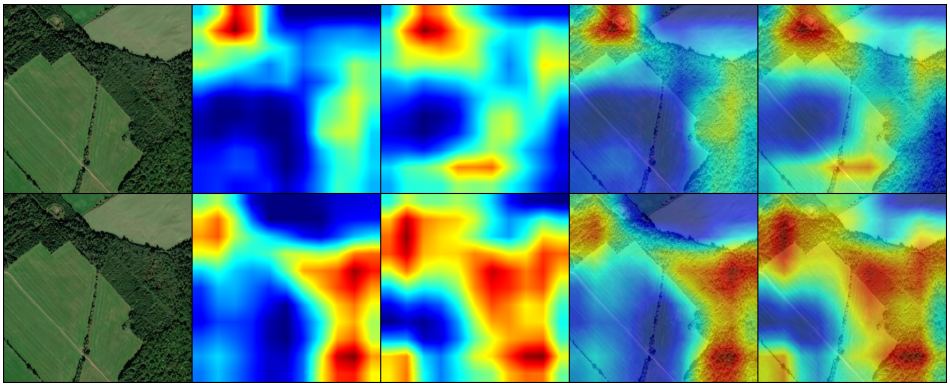
(b) SSL

FIGURE 10 – Confusion matrices for classification task with ResNet based classifiers trained with 160 labeled samples.

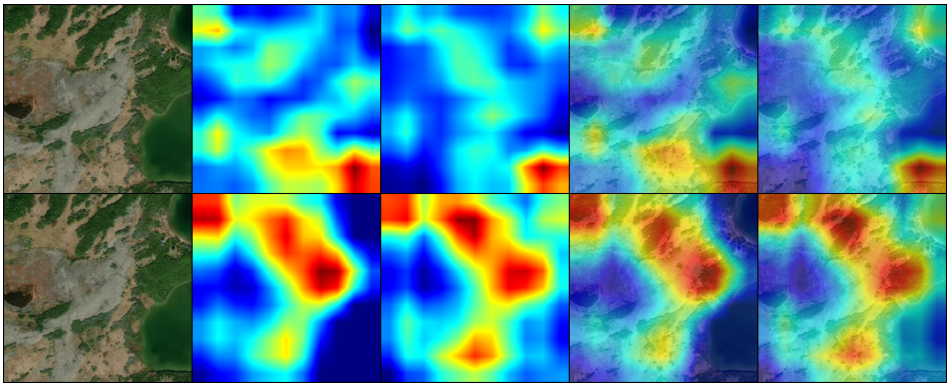
It is also interesting to notice that SL and SSL models do not classify images based on the same criteria. Figure 11 show visual explanation of classification using GradCAM. Both neural networks features to classify. This is noticeable in Figures 11b and 11c



(a) No wildfire sample



(b) Wildfire sample



(c) Wildfire sample

FIGURE 11 – Visual explanation of classification on ResNet based classifier using GradCAM (top row : SL neural network, bottom row : SSL neural network) **both from Method 4**.

6 Conclusion

Many possible classification tasks suffer from a lack of high quality labelised data due to cost and time constrains. Self-supervised learning is a machine learning paradigm that leverage insights contained into unlabeled data through an auxiliary task before performing the downstream task. The goal of this study was to assess the performance gain of SSL on downstream tasks that have large amount of unlabeled data available but few labeled data. We focused on satellite images to predict areas that experienced wildfires and found that SSL can indeed compensate for a lack of labeled data. Despite being considerable, it should however be mentioned that the gain was not enormous. Partly due to the fact that the neural networks without self-supervised training always achieved great

performance. As pointed out by Kaggle users [3], the dataset could contain biased data that induce neural networks to do the wildfire classification based on other criteria such as rural/urban areas which would dramatically simplify the task. We performed quick experiments (notably GradCAM analysis, Figure 11) and found no clear demonstration of such bias in our latest models but further experiments should be carried out.

3. https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset/discussion/436007