Seismic Monitoring and Analysis Challenge - A Quantiles and Zero-differences based solution

Tribel Pascal[1][0009−0006−3701−1223]
Gianluca Bontempi[1][0000−0001−8621−316X]

Machine Learning Group, Université Libre de Bruxelles, Belgique
http://mlg.ulb.ac.be/

Abstract. In the framework of the SMAC, we propose a solution for classifying earthquake-affected and unaffected regions, as well as for performing earthquake magnitude regression. Additionally, we analyze the robustness of our approach. We show how most unaffected regions see no change in the satellite imagery, and how this information helps simplifying the classification and regression methods. We propose the use of `Gradient Boosting` methods to complete this approach.

Keywords: Satellite imagery classification · Magnitude regression · Machine learning

1 Introduction

In the occasion of ECML-PKDD 2024, the Seismic Monitoring and Analysis Challenge has been proposed. In this competition, it is asked to classify a set of satellite images depending on whether or not the represented area has been hit by an earthquake, and, if so, to determine the magnitude of this earthquake. The satellite data is obtained by SENTINEL-1 technique (which works like a radar), emitting vertically polarized signal and receiving the vertically and horizontally polarized reflection. Those two reflections give different information about the structure and shape of the target area. More specifically, their evolution in time give information about the ground deformation. Designing a fast and reliable pipeline for the classification and the regression of events could improve and fasten the rescuing and helping people in areas affected by an earthquake.

2 Data

Three datasets are provided for the competition, implemented in the TorchGeo `QuakeSet` class:

Training set : 2266 samples: 1319 affected and 947 unaffected
Validation set : 550 samples: 309 affected and 241 unaffected
Testing set : 511 samples: 278 affected and 233 unaffected

Dataset	% of affected areas	% of unaffected areas	#affected/#unaffected
Train	58.2%	41.8%	1.393
Validation	56.2%	43.8%	1.282
Test	54.4%	45.5%	1.193

Table 1: Ratios of affected/unaffected regions for each provided dataset

which leads to a total of 3327 samples. The ratios of affected and unaffected regions in the different datasets are given in table 1. Each item in those datasets is a set of four images, a label (affected/unaffected, `0/1`) and an earthquake magnitude (`0` if the region is unaffected, and a value between `4` and `10` otherwise). The four images are Sentinel-1 images (polarized *vertical emission - vertical reflection* `VV` and *vertical emission - horizontal reflection* `VH`) taken respectively at most 13 days before ($\texttt{VV}_{t_0}$ and $\texttt{VH}_{t_0}$) and 13 days after ($\texttt{VV}_{t_1}$ and $\texttt{VH}_{t_1}$) a possible earthquake event. Each of those images represent a 20×20 km^2 area, with a resolution of 512×512 pixels. A pixel has therefore a coverage of 39×39 m^2. The exact time interval between the two time steps is not provided. Figure 1a shows an example of item [1]. The earthquakes in the three datasets are related to events for which the epicenter is distributed across the world. The geographic distribution of those epicenters is shown in figure 1b.

3 Tasks

For each sample, two predictions are performed:

- The classification of the items: *affected* (positive) or *unaffected* (negative). This task is evaluated using the `f1-score`: $\frac{TP}{TP+\frac{1}{2}(FN+FP)}$, where TP, FN and FP denotes respectively the True Positives, False Negative and False Positive.
- The regression of the event magnitude (if any). This is evaluated by the Mean Absolute Error (`MAE`): $\frac{1}{N}\sum_{i=0}^{N}|y-\hat{y}|$, where y and $\hat{y}$ are respectively the expected and the predicted magnitude.

In order to assess the scalability of the pipeline, it is required to compute the number of *floating point operations* (FLOP) for the inference for one item, including the data pre-processing. Unfortunately, the recommended tool, `python-papi`, neither runs on our MacOS laptop nor on our lab GPU server. Therefore we propose a complete computation of this quantity, based on the definition given in [1], which states:

> *A floating-point operation is an addition, subtraction, multiplication, or division operation applied to a number in a single or double precision floating- point representation.*

More importantly, this does not consider comparison nor data access as FLOPs.

[1] See section 10 for the figures

4 Preprocessing

Our solution implies two different kinds of features:

- Quantile-based features
- Zero-differences

Quantiles Quantiles computation only requires comparisons. This lightweight operation is computed on the pixel constituting the four images in an item, and on the pixel-wise differences between $\texttt{VV}_{t_0}$ and $\texttt{VV}_{t_1}$, and between $\texttt{VH}_{t_0}$ and $\texttt{VH}_{t_1}$. We compute every quantile of 2% from 0% up to 98% included. This leads to a total of 300 features.

Zero-differences At the scale of the provided samples (1 pixel covering $39 \times 39\ m^2$), changes have to be massive enough to imply a difference in the satellite imagery. An unaffected area could exhibit no difference between the two time steps. Therefore, if one of the two channels difference is all-zero, then the image can be directly labelled as *unaffected* and its magnitude can be set to zero[2]. This tool allows to correctly classify $\frac{709}{947}$ samples in the training set, $\frac{152}{241}$ in the validation set and $\frac{113}{233}$ in the test set. Checking this for one sample requires $2 \times 512 \times 512$ FLOPs (for the computing the pixel-wise differences).

Noisy samples There may be some noise present on the different images. Our way to process this noise can be explained by looking at the histogram shapes of the two item classes. We show the shape of the histogram of the difference between `VV` images, and of the difference between `VH` images, if the area is affected in figure 2a and if the area is unaffected in figure 2. In the case of a normally-random noised image, the differences will be normally distributed around 0. We decide to manually threshold the differences to 0.75 in absolute value for `VV` images and to 0.05 for `VH` images. The number of FLOP for this operation is $1 \times 512 \times 512$, implied by the difference between the channels. The threshold and the quantiles are obtained by comparisons and therefore require 0 operations.

Standardization We standardize the extracted features using the norm and the variance of the training set features. Each feature has to be standardized, so this implies $300 * 2$ FLOPs.

[2] We show in the appendix code that on the whole `train+val+test` set, only one item has a zero difference despite the fact it is labelled as `positive` and has a magnitude of 5.49. This item is discarded from the training set. Aside from this outlier, the zero-difference rule is correct in all the cases.

5 Model

The classification and regression tasks are decomposed in three steps:

1. A first estimation of the item class is computed. We predict as *unaffected* the ones having at least one channel difference being all-zeros. For the remaining ones, we propose to use the lightgbm's `LGBMClassifier`, trained on a dataset from which half of the zero-difference samples, randomly drawn, are excluded (to reduce the bias implied by the zero-difference samples). `GradientBoosting` methods are based on decision trees, which are made of comparisons. The number of FLOPs required by the inference step is then 0.
2. We compute the regression. We add the class predicted in the previous step as a feature in the input dataset, and we use lightgbm's `LGBMRegressor` to compute the magnitude. The number of required FLOPs is also 0.
3. The magnitude value is finally used to determine the class: `positive` if the magnitude is non-null, `negative` otherwise. This ensures that the regressed magnitude and the associated class are coherent (as suggested in the provided `starter-kit`).

6 FLOPs evaluation

The only FLOPs that are computed in the presented pipeline are the differences used in the preprocessing step and the features normalization. The total number of FLOP needed for the inference of one sample is then: $4 \times 512 \times 512 + 600 = 1049176$. This result is coherent with the SVC and RFC FLOPs quantity proposed for the classification task in the `QuakeSet` presentation article.

7 Results

We show in figures 3a and 3b the confusion matrices for the classification of the validation and the test datasets, and the expected/predicted magnitudes for the same sets in figures 4a and 4b. We see that the classification is entirely correct.

8 Robustness study of the pipeline

In order to assess the robustness of our pipeline, we compute the different metrics when the model is trained and validated on different portions of the dataset. We show that our model is pretty robust to the different combinations. We show in figure 5 the `f1-scores` obtained for different combinations of training set (X) and testing set (y) and the `MAE` for the regression task on the same combinations in figure 6.

9 Conclusion

In this work, we have detailed our solution for the SMAC. We have shown that a great proportion of unaffected areas presented no change in the images, and that this information could be used to improve the predictions. We have shown that using quantile-based features alongside `Gradient Boosting` methods could produce high-quality results while remaining highly scalable. Other strategies have been attempted, such as **CNN**, **Transformers** and **RNN**, and several other strategies for features extraction have been implemented. However, they lead to an extreme increase of required resources, while not improving a lot the performances of the pipeline. We have consciously made the choice of simplicity for the sake of scalability.

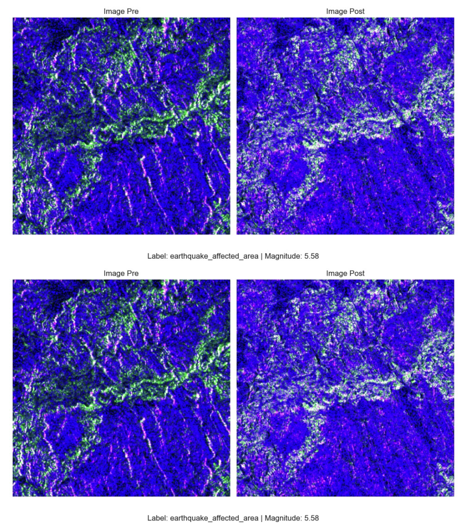
The code is available on GitHub: https://github.com/pascaltribel/smac-solution/tree/main.

Disclosure of Interests. The author declares having no competing interests.

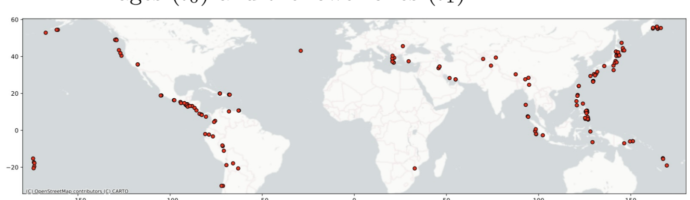
References

1. In More Depth: MFLOPS as a Performance Metric, https://course.ccs.neu.edu/cs3650/ssl/TEXT-CD/Content/COD3e/InMoreDepth/IMD4-MFLOPS-as-a-Performance-Metric.pdf, last accessed 2024/06/24

10 Figures

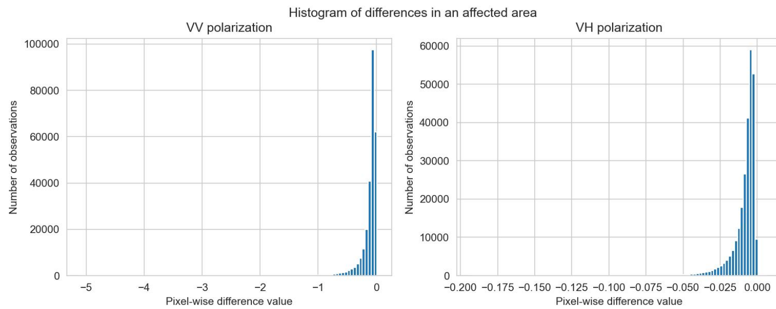


(a) An example of item. We see that there are a lot of changes between the upper images (t_0) and the lower ones (t_1).

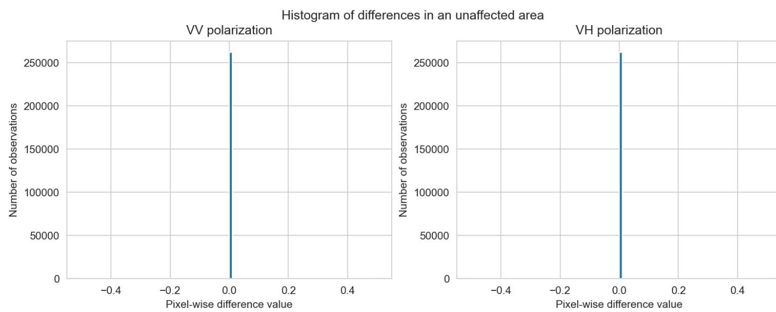


(b) Geographic distribution of the earthquakes epicenter across the world.

Fig. 1: Dataset overview

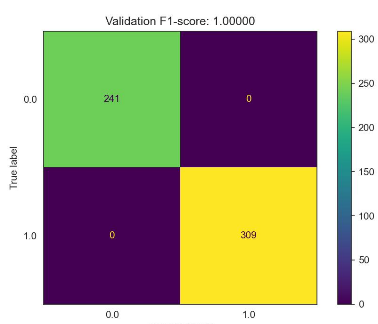


(a) Example of histogram of the difference between images in an affected region

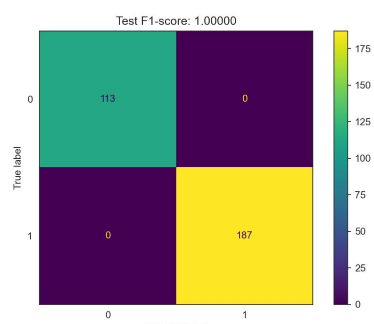


(b) Test set

Fig. 2: Example of histogram of the difference between images in an unaffected region

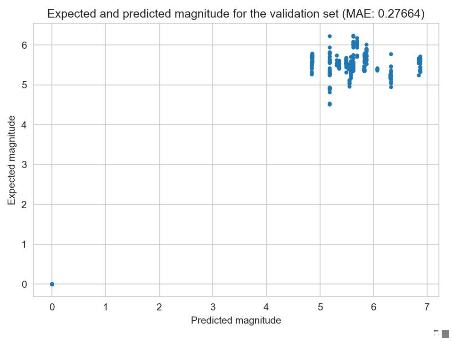


(a) Validation set

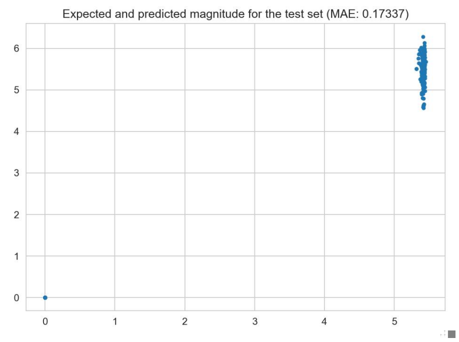


(b) Test set

Fig. 3: Classification confusion matrices



(a) Validation set



(b) Test set

Fig. 4: Regression plot of the expected/predicted values

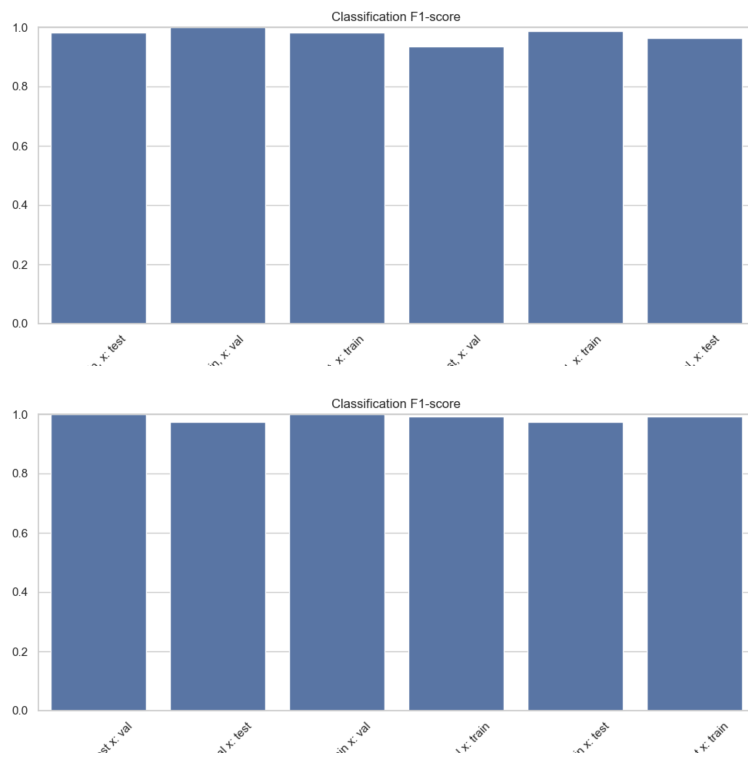


Fig. 5: Classification f1-scores for different combinations of dataset. The result is quite similar for all the combinations, showing the robustness of the classification pipeline.

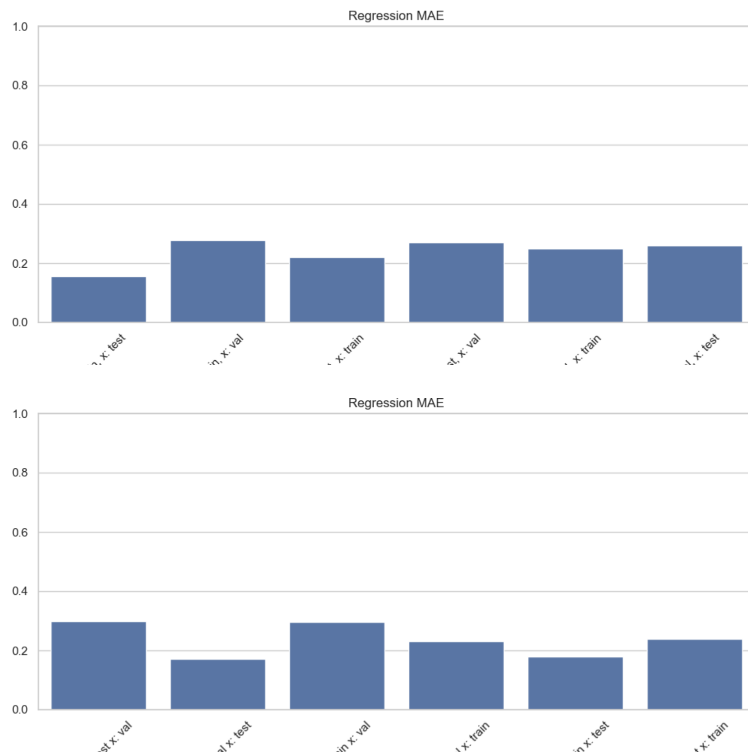


Fig. 6: MAE of the regression task for different combinations of dataset. Again, the results are consistent for all the combinations.