ML-based event classification for DAS

- Image classification with YOLO11 -> a set of images will be classified into one of a set of **predefined** classes.
- Why?
  - Important to ultimately have a tool that allows us to quickly know whether an “**special event”** is registered with DAS
  - From a ML beginner perspective: YOLO is easy to use <- Dídac experience (thank you!)
- Method (first steps):
  - Classify spectrograms from different channels
    - DAS data is saved continuously (20 channels in 5-min hdf5 files) (thank you, Abdel!)
    - Optimize inputs: image resolution, shape, pixel number, etc
    - Define classes + **manual classification**

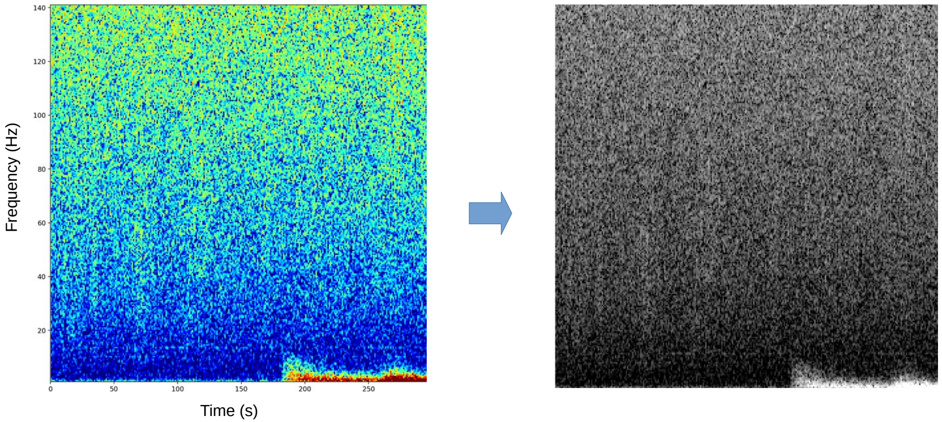
ML-based event classification for DAS

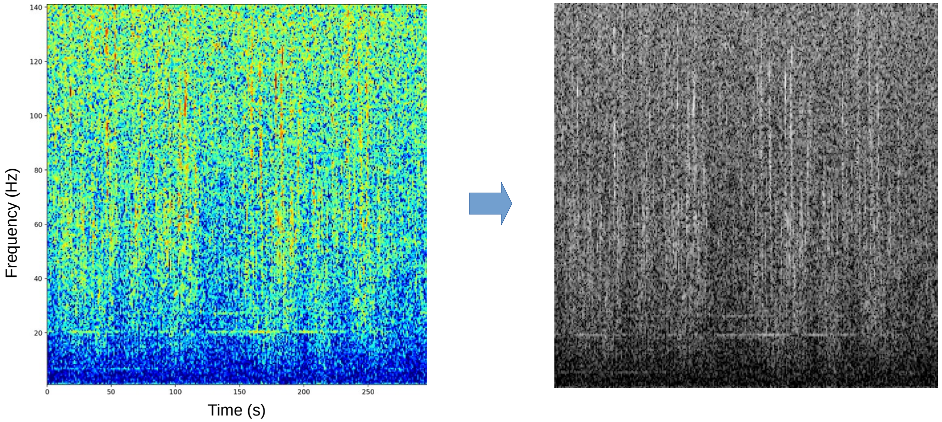
Spectrograms as inputs

- Squared images
- Black and white colour scale
- Keep the same colour scale
- Maximize image to the whole frame -> no axes
- Number of pixels -> multiple of 32

Dídac’s script:

```
Square spectrogram:
freqs: [1.0,141.1] Hz -> 288 pxs
Tanalysis: 295.9360s
NFFT: 1024 ; Tbin_spect: 1.0240s ; Fbin_spect: 0.49Hz ; Fmin_spect: 0.98Hz
```





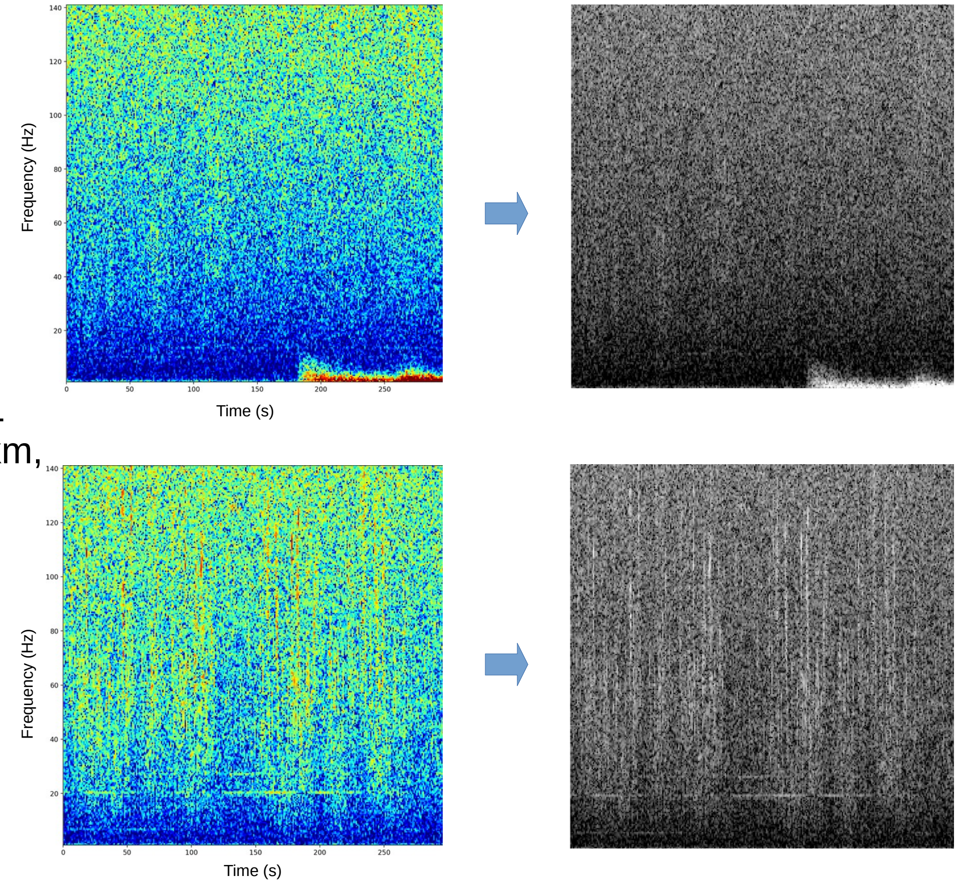
ML-based event classification for DAS

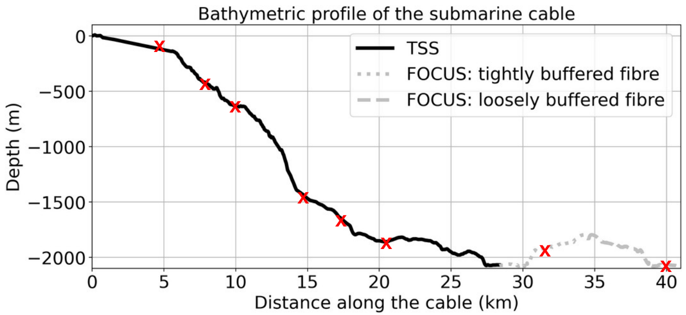
Spectrograms as inputs

- Squared images
- Black and white colour scale
- Keep the same colour scale
- Maximize image to the whole frame -> no axes
- Number of pixels -> multiple of 32
- Used data from 3 different days (7 June, 10 June and 21 June) and channels (distances from shore: 4.9 km, 7.8 km, 9.8 km, 14.8 km, 17.7 km, 21.6 km, 32.4 km, 40.2 km)

Dídac's script:

```
Square spectrogram:
freqs: [1.0,141.1] Hz -> 288 pxs
Tanalysis: 295.9360s
NFFT: 1024 ; Tbin_spect: 1.0240s ; Fbin_spect: 0.49Hz ; Fmin_spect: 0.98Hz
```





ML-based event classification for DAS

How many classes?

- Manual labelling -> ~ 1500 spectrograms
- I tried to include noisy data for a better performance, however, I left out many confusing signals -> lower number of total images
  - 2 classes: vessels (VS) & noise (NO)
  - 3 classes: vessels (VS), explosions (EX), noise (NO)
  - 4 classes: vessels (VS), explosions (EX), noise (NO), airguns (AG)

ML-based event classification for DAS

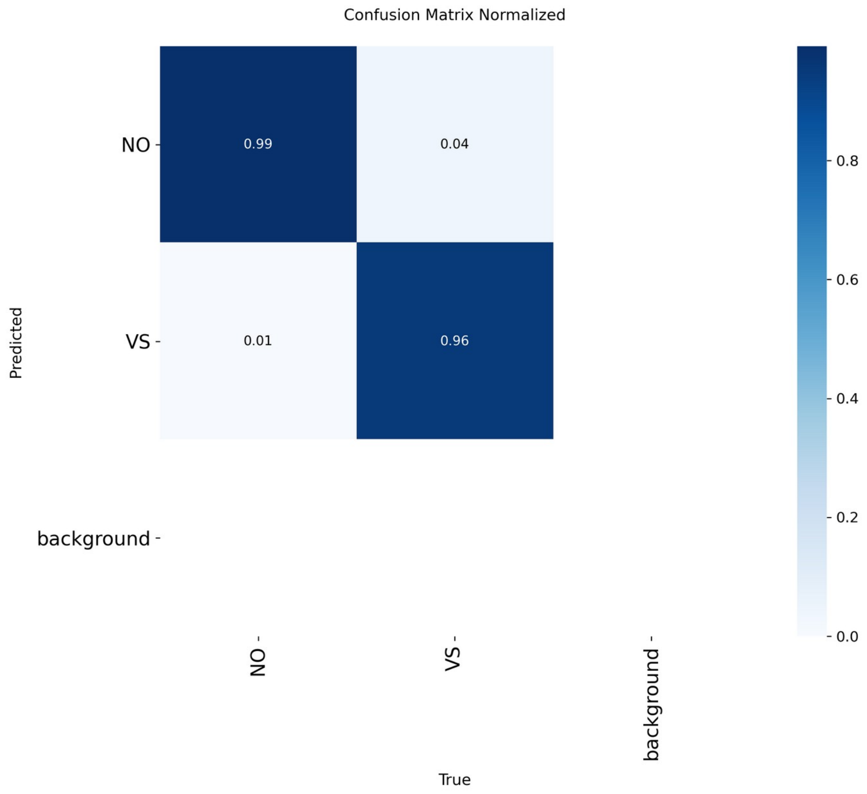
2 classes: vessels (VS) & noise (NO)

- Total number of spectrograms -> Vessels (VS) = 308; Noise (NO) = 872
- **Training** step -> 70 % of the dataset
- Rest of the dataset split into **validation** (15 %) and **test** (15 %)

<u>**Results from training step**</u>

Confusion matrix:

- To know how well the ML model predicts each class
- For 2 classes (VS & NO) the model is performing pretty well!



ML-based event classification for DAS

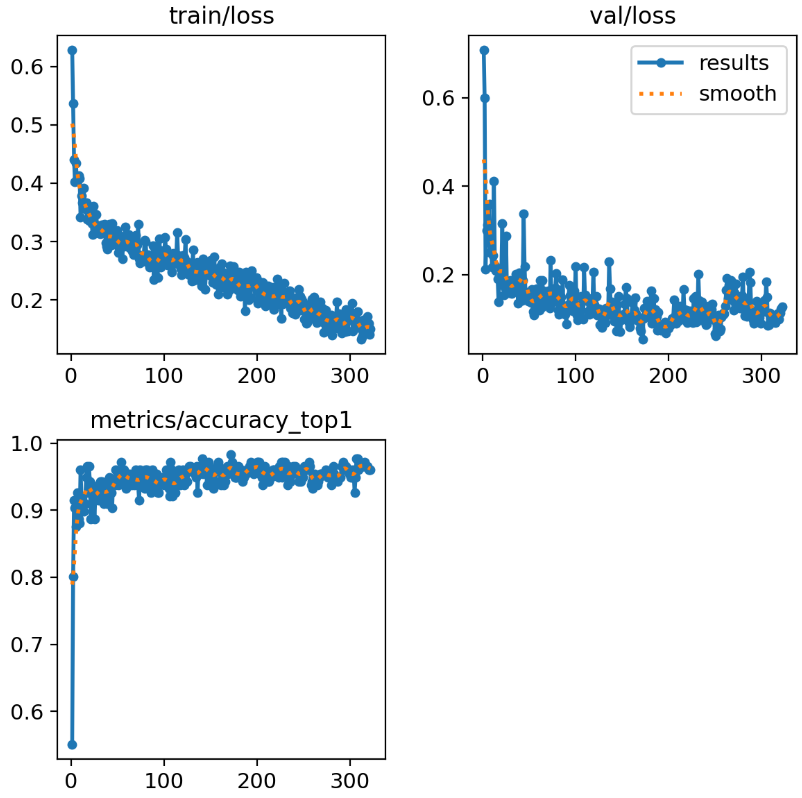
2 classes: vessels (VS) & noise (NO)

- Total number of spectrograms -> Vessels (VS) = 308; Noise (NO) = 872
- **Training** step -> 70 % of the dataset
- Rest of the dataset split into **validation** (15 %) and **test** (15 %)

<u>Results from training step</u>

Performance metrics:

- **Train/Loss**: how wrong the model is on the **training data.** Drops from ~0.62 to 0.15 -> Model is learning and improving.
- **Val/Loss**: how wrong the model is on the **validation data.** Drops from ~0.8 to ~0.1, then levels -> Model generalizes well.
- **Accuracy Top 1** (accuracy on **validation set**): % times the **most likely predicted class** matches the true label. Reaches ~0.97 -> excellent performance



ML-based event classification for DAS

2 classes: vessels (VS) & noise (NO)

<u>Results from validation set</u>

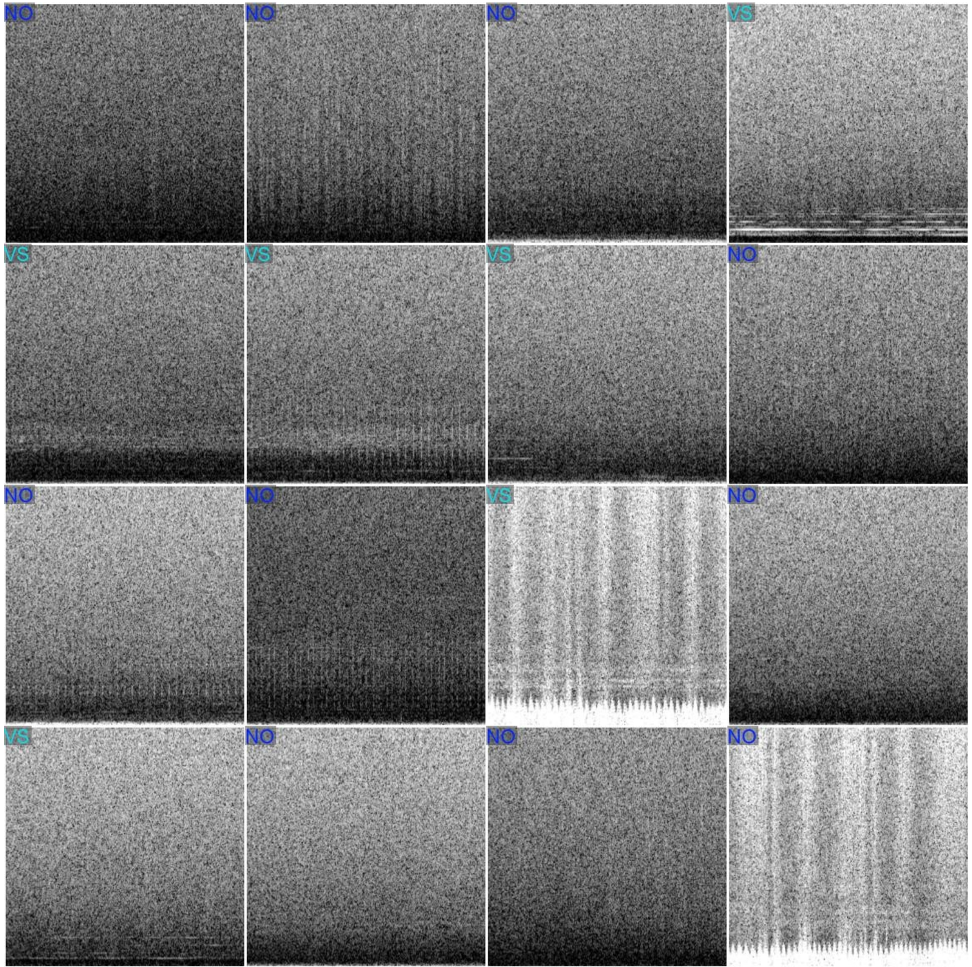
- Accuracy top 1: ~ 0.98

<u>Results from test set</u>

(unbiased evaluation for trained model)

- Accuracy top 1: ~ 0.89

Example (validation set): **label**



ML-based event classification for DAS

2 classes: vessels (VS) & noise (NO)

Results from validation set

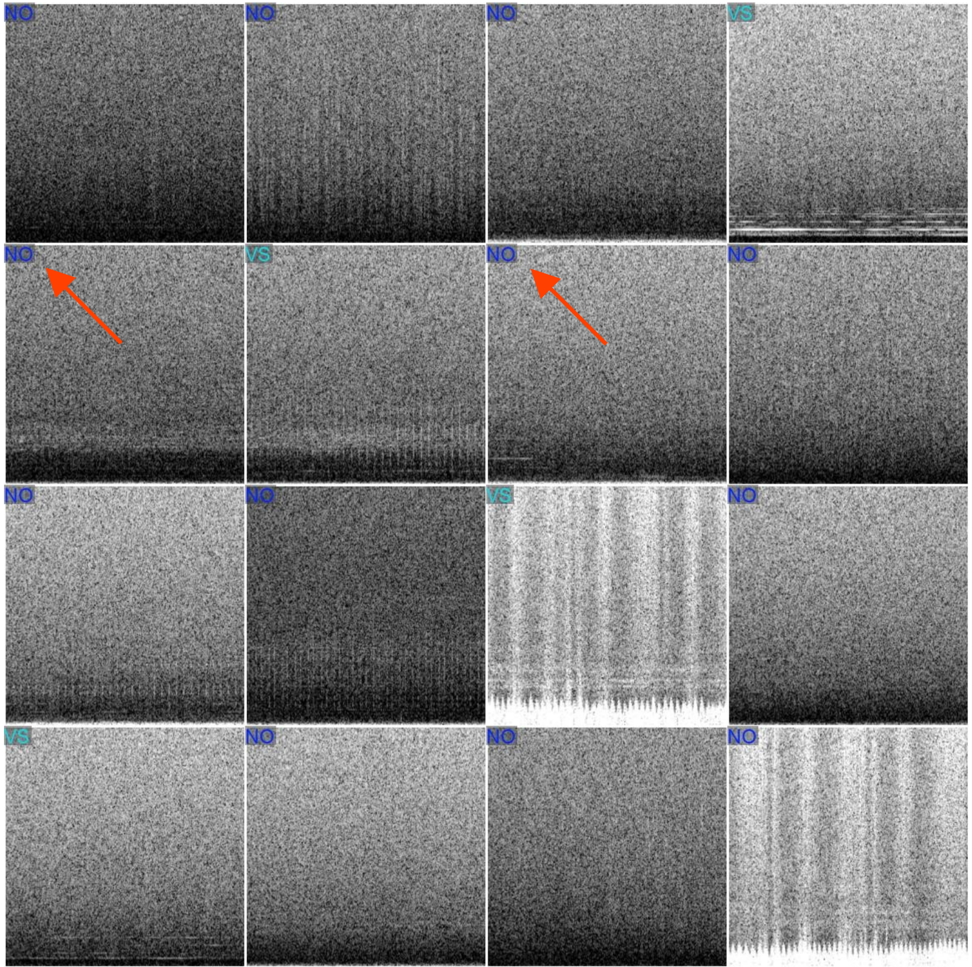
- Accuracy top 1: ~ 0.98

Results from test set

(unbiased evaluation for trained model)

- Accuracy top 1: ~ 0.89

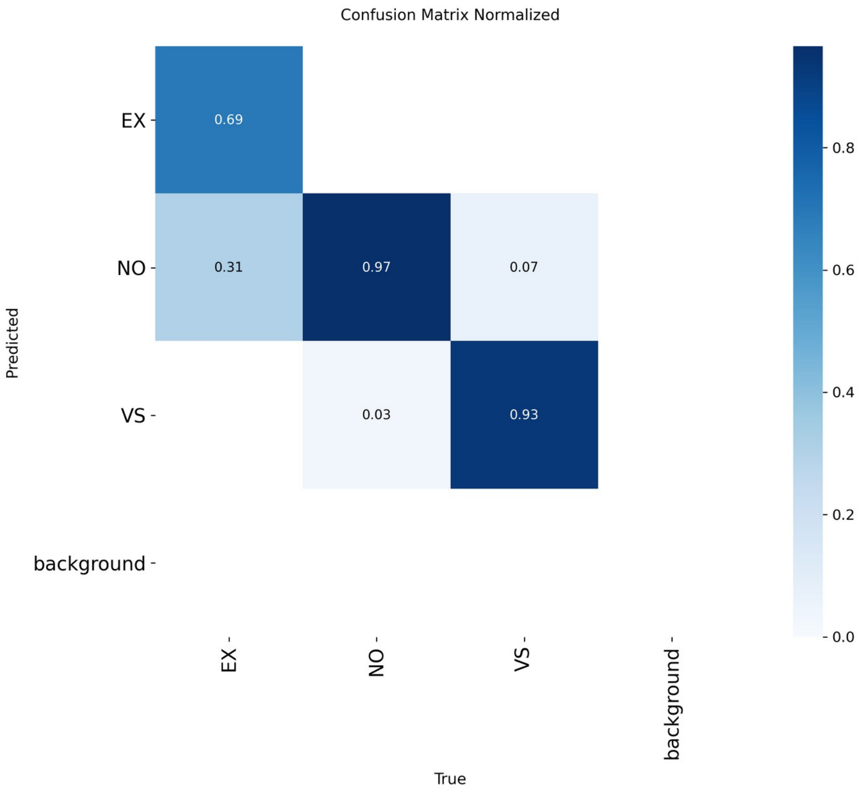
Example (validation set): **prediction**



ML-based event classification for DAS

3 classes: vessels (VS), explosions (EX), noise (NO)

- Total number of spectrograms -> Vessels (VS) = 308; Explosions (EX) = 90; Noise (NO) = 791
- **Training** step -> 70 % of the dataset
- Rest of the dataset split into **validation** (15 %) and **test** (15 %)



<u>Results from training step</u>

Confusion matrix:

- To know how well the ML model predicts each class
- ~ 31% of explosions/earthquakes (EX) are misclassified as noise (NO)

ML-based event classification for DAS

3 classes: vessels (VS), explosions (EX), noise (NO)

Results from validation set

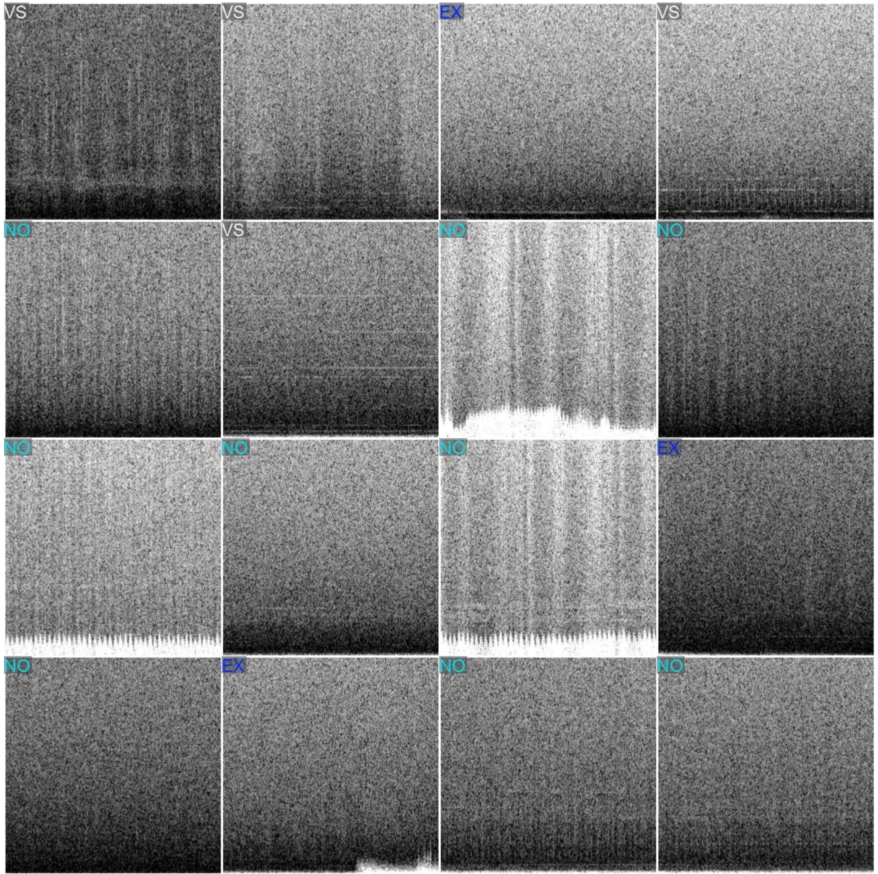
- Accuracy top 1: ~ 0.94

Results from test set

(unbiased evaluation for trained model)

- Accuracy top 1: ~ 0.86

Example (validation set): **label**



ML-based event classification for DAS

3 classes: vessels (VS), explosions (EX), noise (NO)

<u>Results from validation set</u>

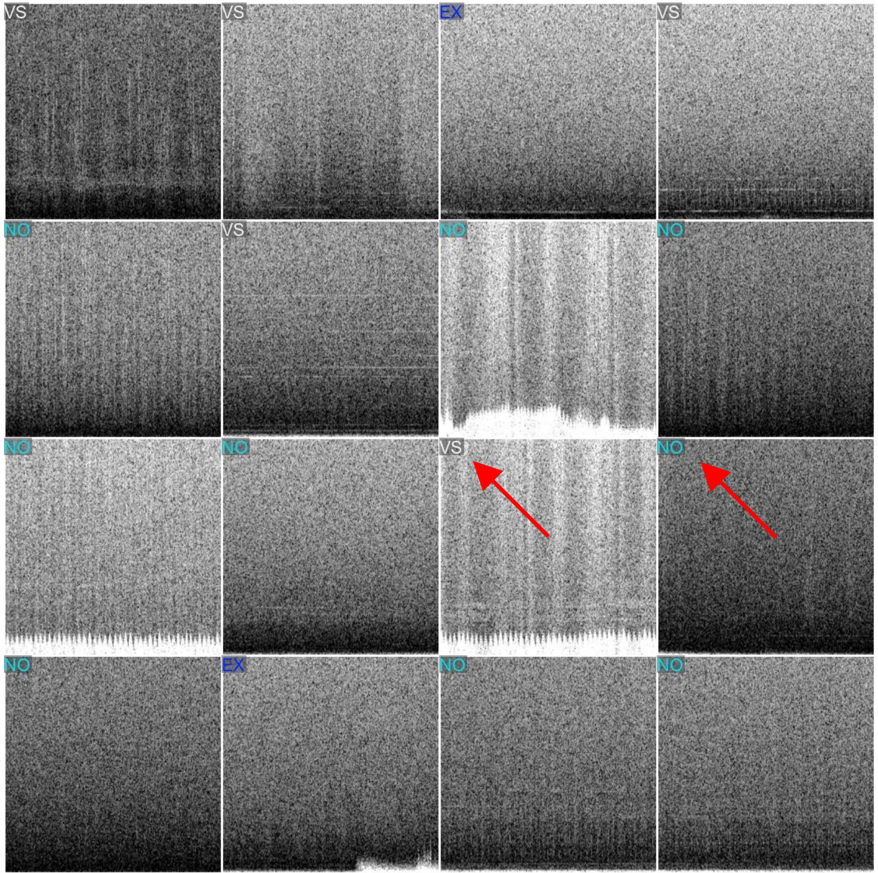
- Accuracy top 1: ~ 0.94

<u>Results from test set</u>

(unbiased evaluation for trained model)

- Accuracy top 1: ~ 0.86

Example (validation set): **prediction**



ML-based event classification for DAS

3 classes: vessels (VS), explosions (EX), noise (NO)

- Total number of spectrograms -> Vessels (VS) = 308; Explosions (EX) = 90; Noise (NO) = 791
- **Training** step -> 70 % of the dataset
- Rest of the dataset split into **validation** (15 %) and **test** (15 %)

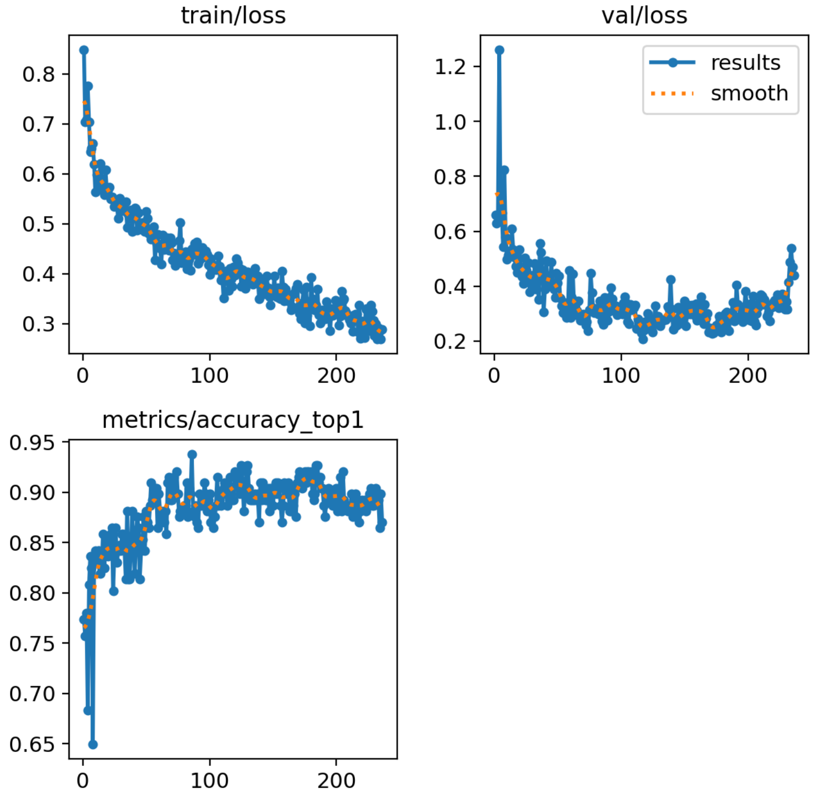
<u>**Results from training step**</u>

Performance metrics:

- **Train/Loss**: Drops from ~0.85 to ~0.3 -> Model is learning and improving.
- **Val/Loss**: Drops from ~1.2 to ~0.35 -> Model generalizes well, **low overfit.**
- **Accuracy Top 1** (accuracy on **validation set**): Oscillates ~ 0.9.

-> In general, the model is doing pretty well because:

- Results from validation set are good (high accuracy, low loss)
- Improvements: EX class -> **Problem of class imbalance?**



ML-based event classification for DAS

3 classes: vessels (VS), explosions (EX), noise (NO)

- Total number of spectrograms -> Vessels (VS) = 308; Explosions (EX) = 90; Noise (NO) = 791
- Checking class imbalance: using **scikit-learn** to get per-class metrics.

Per-class metrics (scores between 0 and 1)

- **Precision**: how many **predicted items** are actually correct? Related to false positives.
- **Recall**: how many of the **real items** are actually found? Diagonal of confusion matrix.
- **F1-score**: combination of precision + recall.

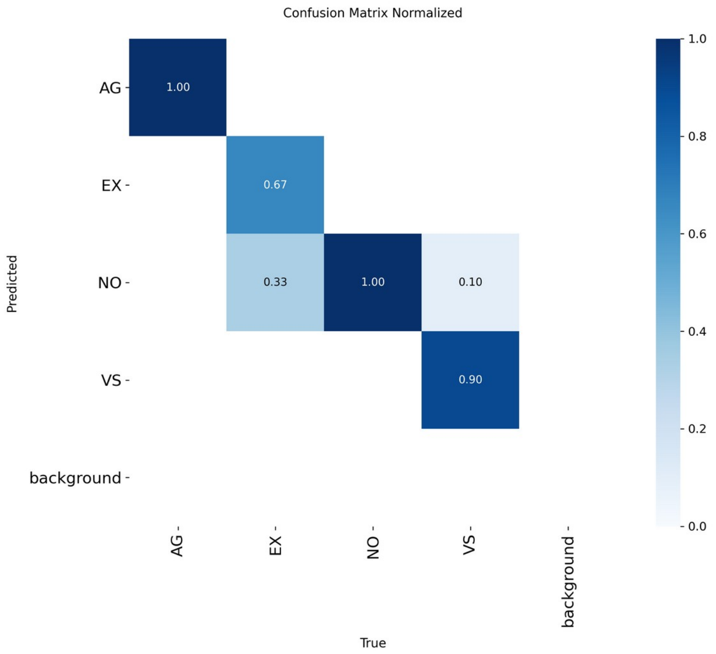
Interpretation

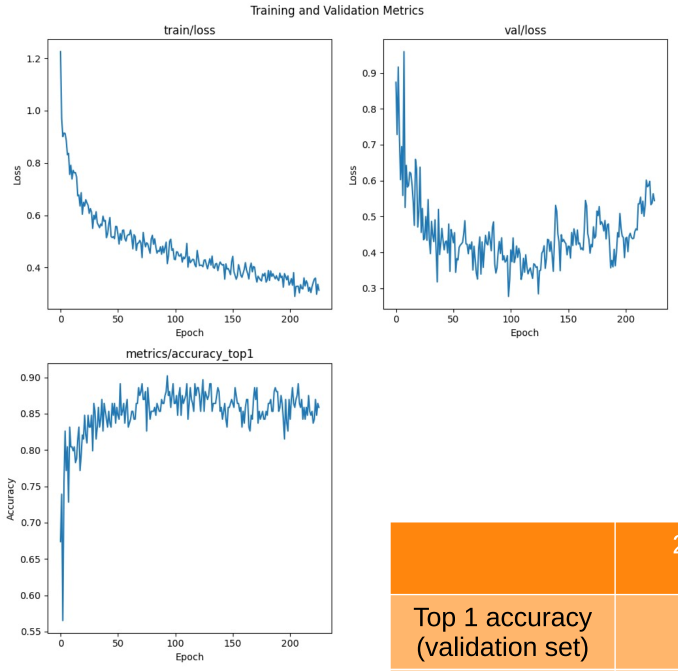
- The model is learning more about the dominant classes (NO & VS)
- Rare class (EX): fewer examples, high precision (no false positives), lower recall (misses real detections).

ML-based event classification for DAS

4 classes: vessels (VS), explosions (EX), airguns (AG) & noise (NO)

- Total number of spectrograms -> **Vessels (VS) = 308; Explosions (EX) = 90**; Airguns (AG) = 228; Noise (NO) = 607





Comparison with 3 classes

- The items in classes VS and EX don’t change.
- Similar results for EX
- Even though performance metrics are lower, the accuracy is more than 3 times better than the baseline accuracy (random chance = 0.25)

	2 classes	3 classes	4 classes
Top 1 accuracy (validation set)	0.98	0.94	0.90
Top 1 accuracy (test set)	0.89	0.86	0.94

ML-based event classification for DAS

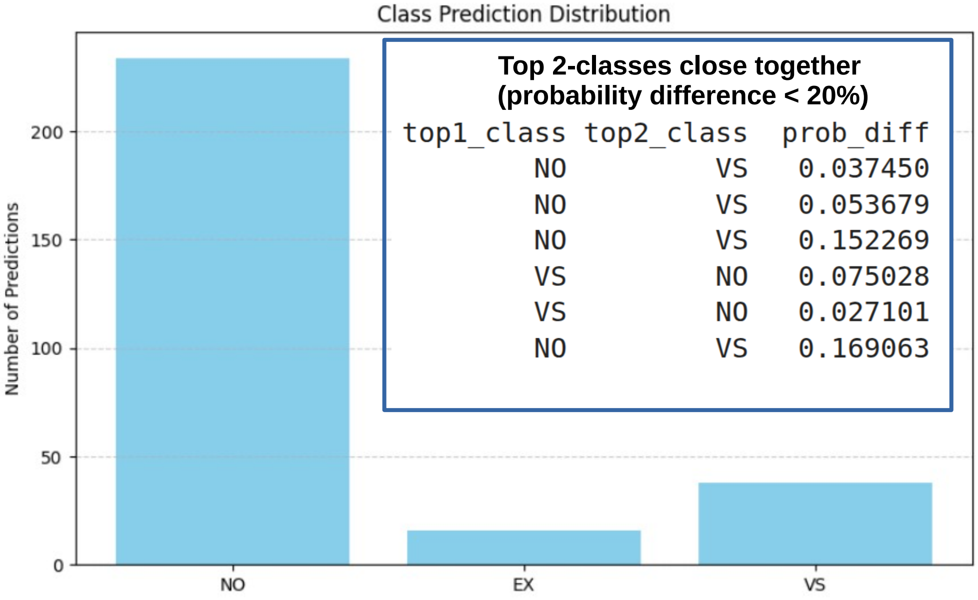
Next steps

- The class imbalance hurts earthquake/explosion detection -> Balance the class contribution:
  - Reduce the number of spectrograms from the other classes to have more similar number in all classes
  - Ideally: add more spectrograms to the class EX.
- Use models to make predictions -> classify a new set of spectrograms
- Implement an automatized classification in the workflow?

ML-based event classification for DAS

Next steps

- Use models to make predictions -> classify a new set of spectrograms
  - How well it works the model for four classes as it is (**AG, EX, VS, NO**)? -> using data from 30/06/2025, channel 1960 (distance ~ 9.8 km)



Top 2-classes close together (probability difference < 20%)

top1_class	top2_class	prob_diff
NO	VS	0.037450
NO	VS	0.053679
NO	VS	0.152269
VS	NO	0.075028
VS	NO	0.027101
NO	VS	0.169063