ALBUS : Anomaly detector for Long duration BUrst Searches

Vincent Boudart
PhD student, University of Liege

Advisor : Dr Maxime Fays

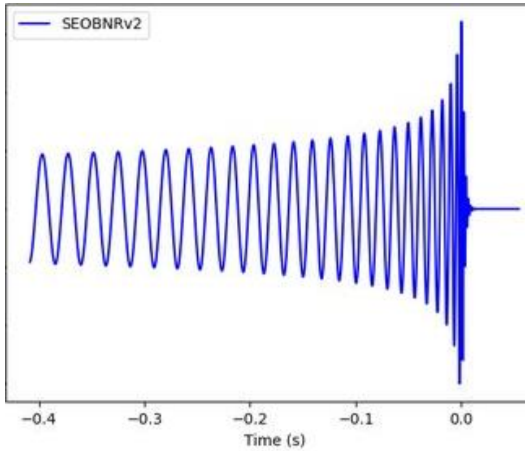
BeGWaMe 2 (ULB/VUB), Nov 3, 2021

Table of Contents

1) How do we detect burst signals ?

- CBC detection : general relativity => model of collision = waveform

=> then try to match those models to the data (matched filtering)

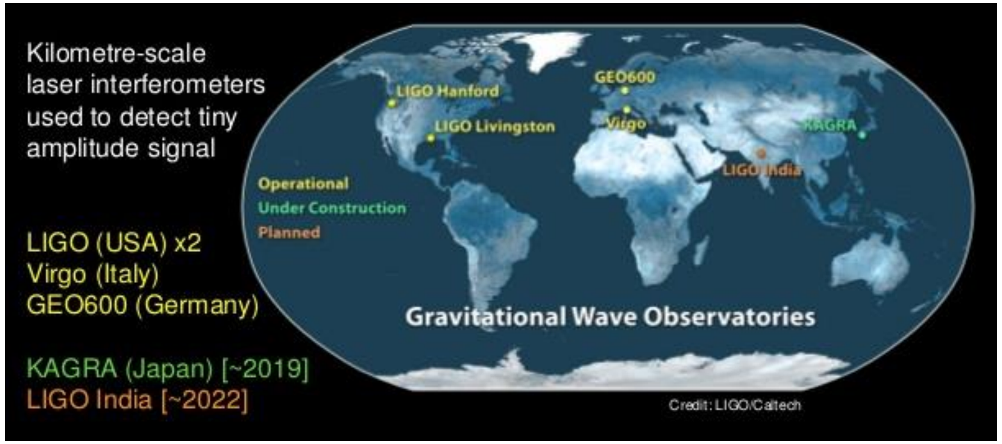


- Many other phenomena can generate GWs !

=> But physics is poorly known...

=> Models not accurate enough to apply match filtering.

Solution : use multiple detectors to find correlation in the data



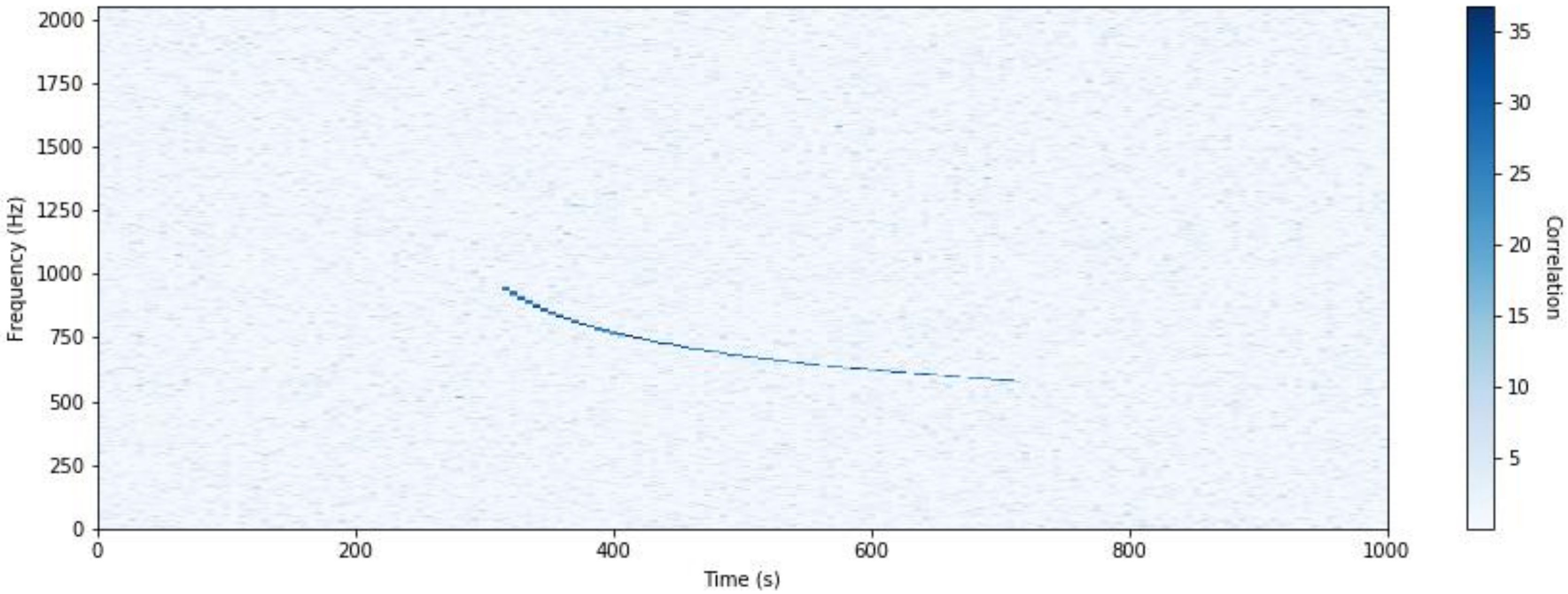
1) How do we detect burst signals ?

- Excess of power method

=> Search in Time-Frequency space : bursts should be clusters of high-correlation pixels

=> Many sources of noise (seismic, laser noise, suspensions, etc.)

=> Focus on long duration events (>10 seconds)

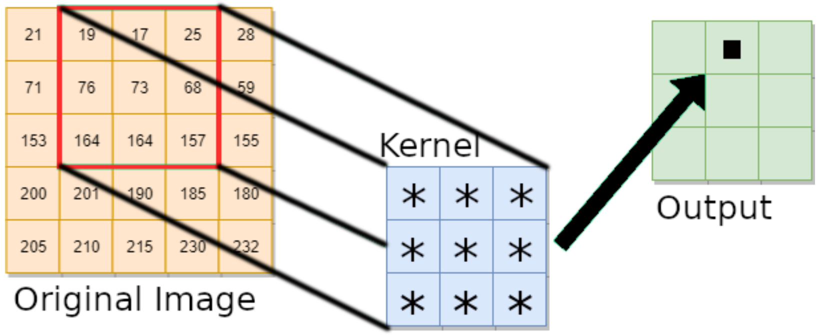


2) Convolutional neural networks

- Class of artificial neural networks employing convolution

=> easy to use and understand

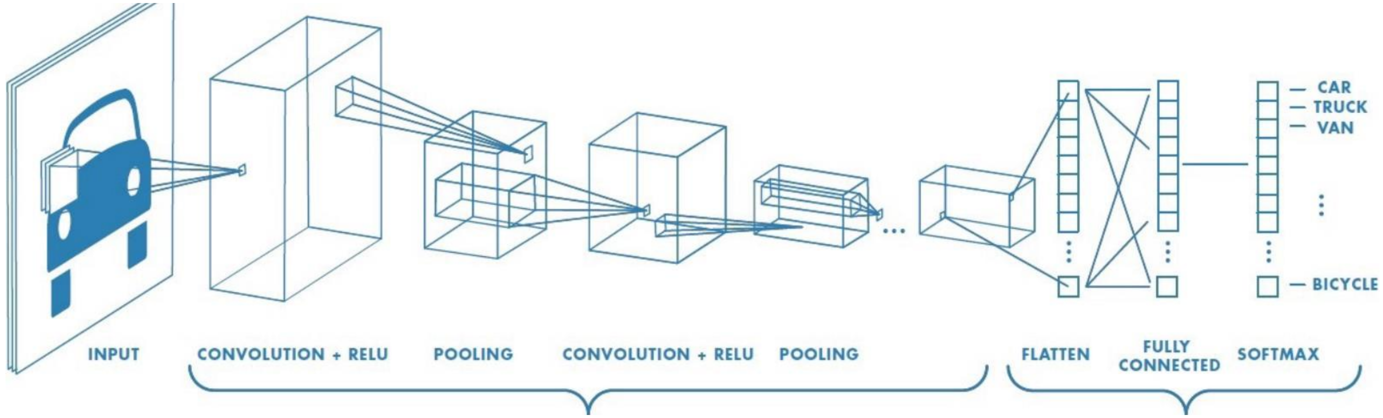
=> allows to downscale the information

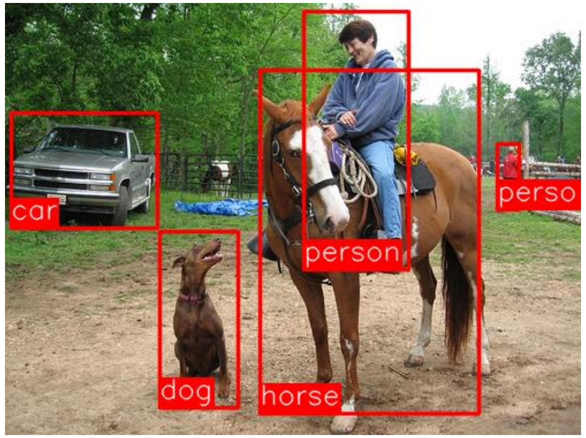


- Image processing applications often require :

=> classification tasks (medical images, galaxy catalogs, ...)

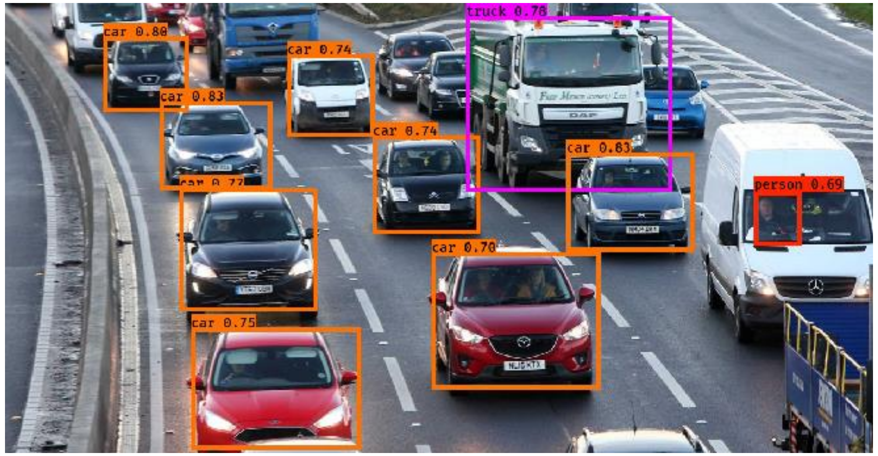
=> bounding box determination (self-driven cars, face recognition, ...)

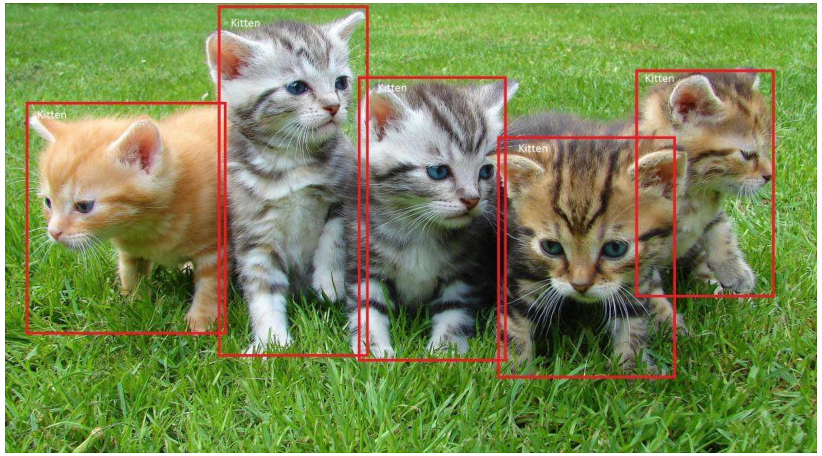




2) Convolutional neural networks

- Efficient at recognizing patterns and shapes :



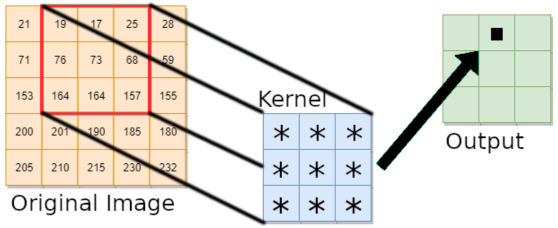


- Note : a neural network is nothing without a well-designed loss function !

=> loss function = what you want to minimize to achieve your goal (classification, prediction, …)

=> loss function gives feedbacks to update the weights (in kernels, …)

=> bad weight updates = badly conditioned training = bad results



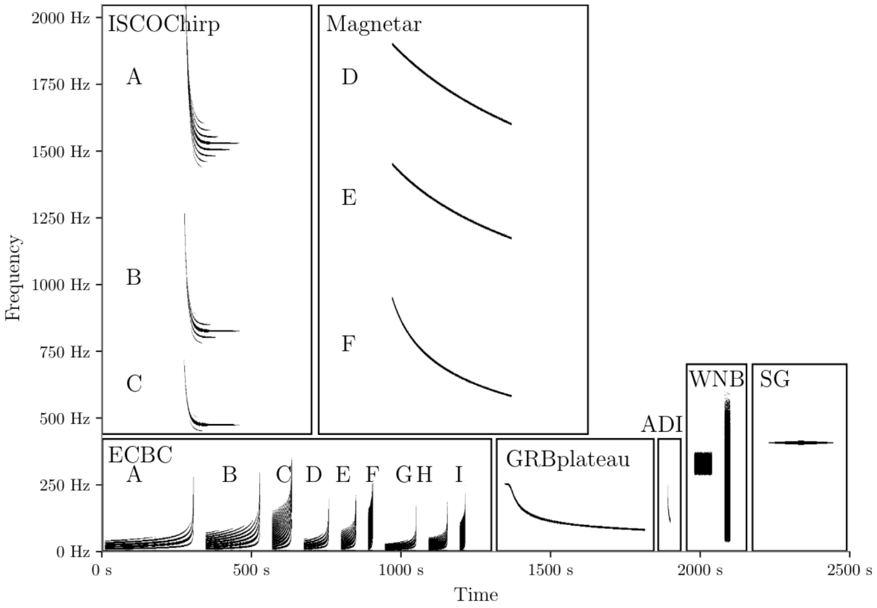
3) New approach : mimic long-duration burst signals

- Problem : can't rely on the long-duration models

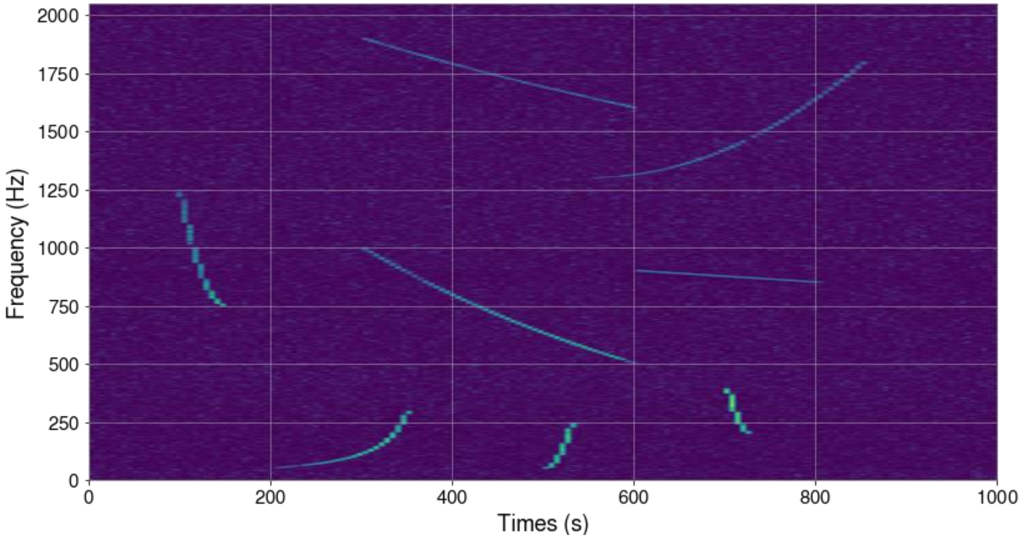
- too many uncertainties in the physical phenomena
- cannot be used as patterns to recognize

- They all show a "chirp up" or "chirp down" behavior

==> easily mimicked thanks to the *Python Scipy* library !
==> Allow to generate chirps as time series

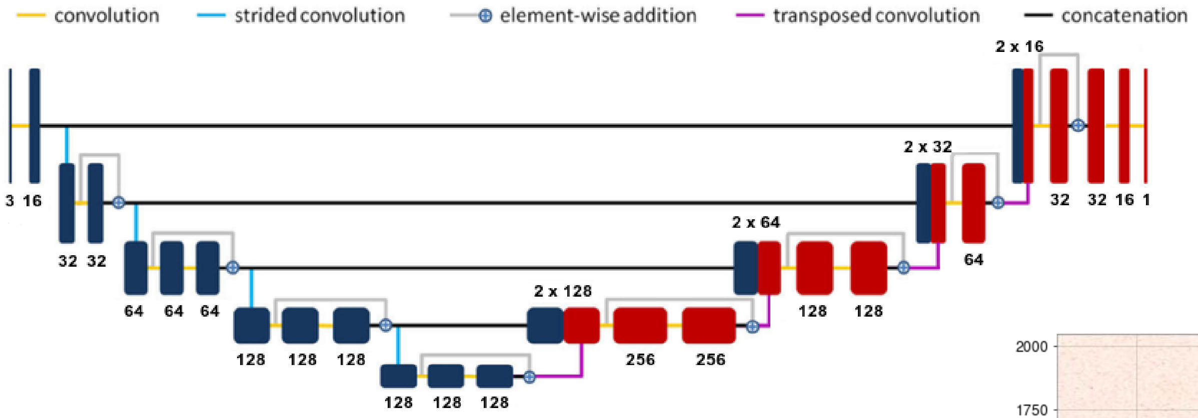


Taken from O3 long-duration paper : https://dcc.ligo.org/public/0174/P2100078/011/o3_long_duration.pdf



3) New approach : mimic long-duration burst signals

- Inspired by ***Xing et al., 2019***. (https://doi.org/10.1186/s12859-019-3037-5), coded with PyTorch
- Downscaling and upscaling network + skipped connections + ELU activation

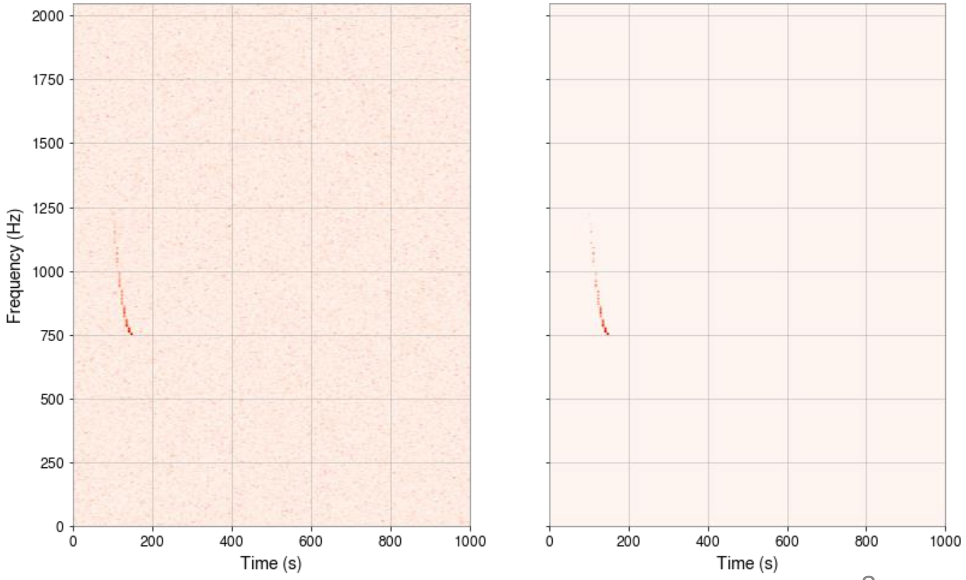


- Method :

- train the network so that : output (O) ≃ target (T)

==> our target will be injection in empty TF map

==> Empty map for noise-only images



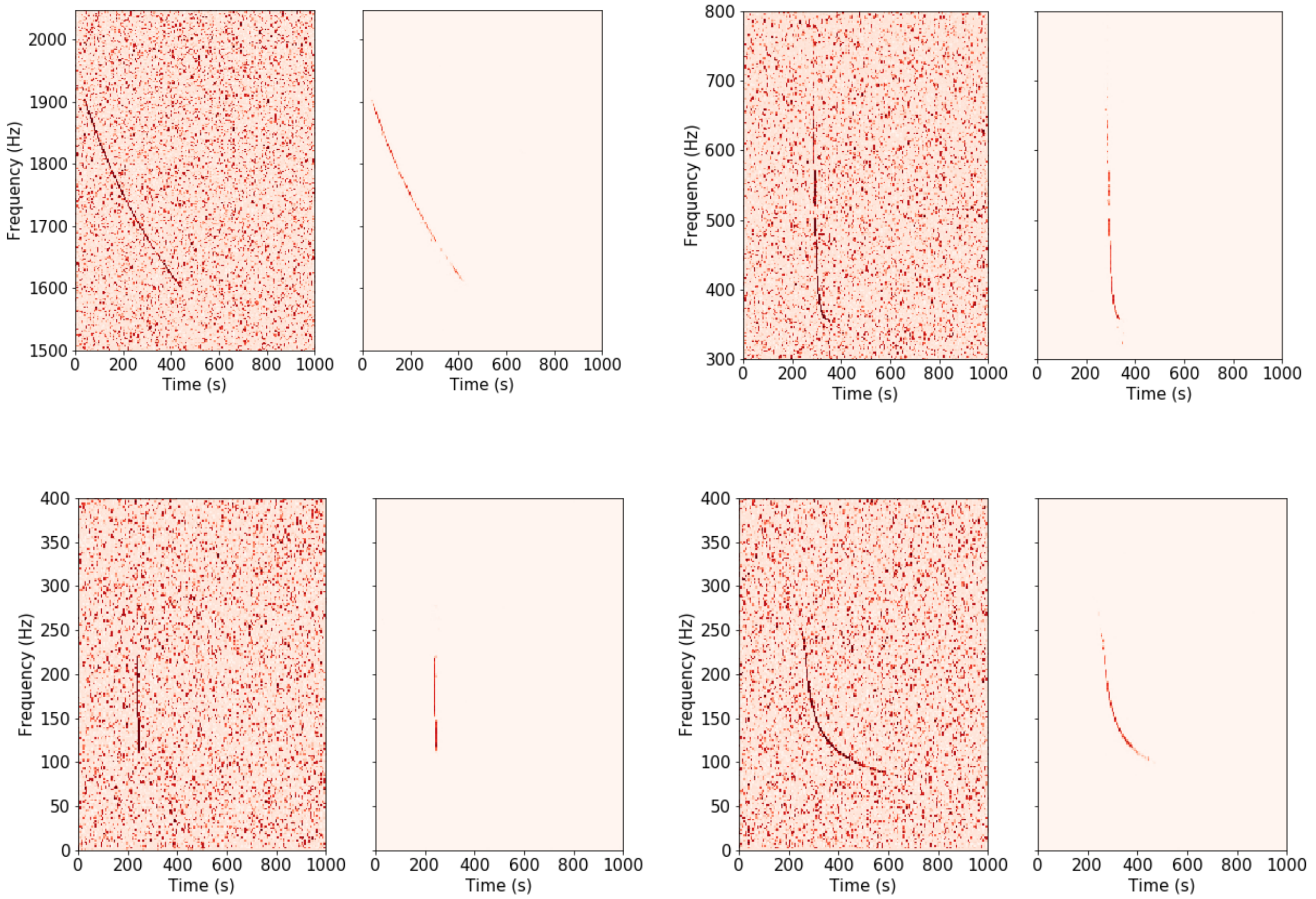
- Loss that is being minimized : $MSE = \frac{1}{2} \sum_{i,j} (T_{ij} - O_{ij})^2$

4) Early Results

- Localization : TF maps with injection

- Values > 0.5 for the detected signals

- Pixel-wise localization reached !

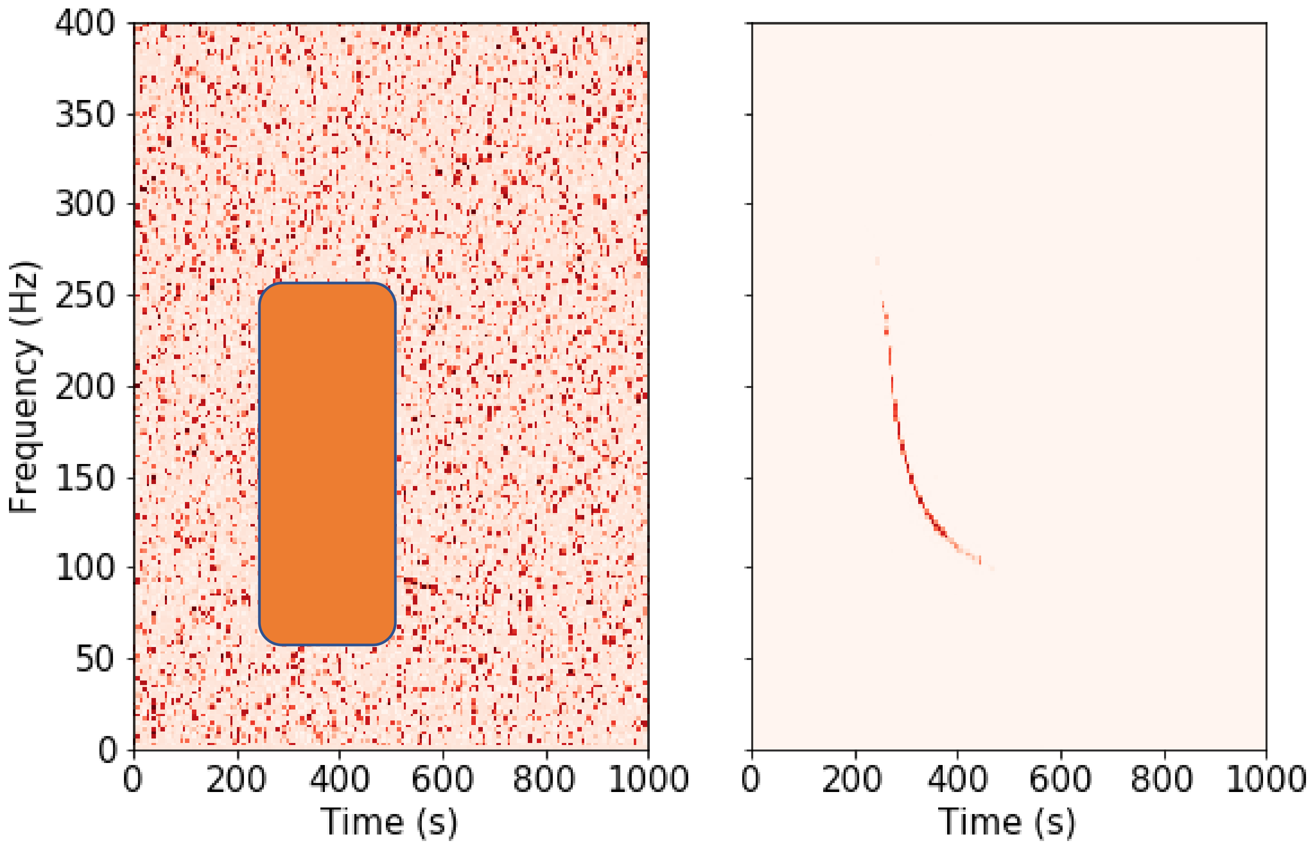


4) Early Results

- Localization : TF maps with injection

- next step : learn the connectivity
 between pixels

==> What about the time-frequency maps with only pure noise ?

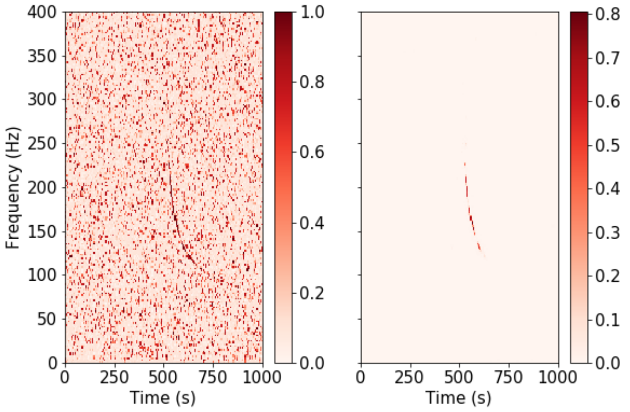


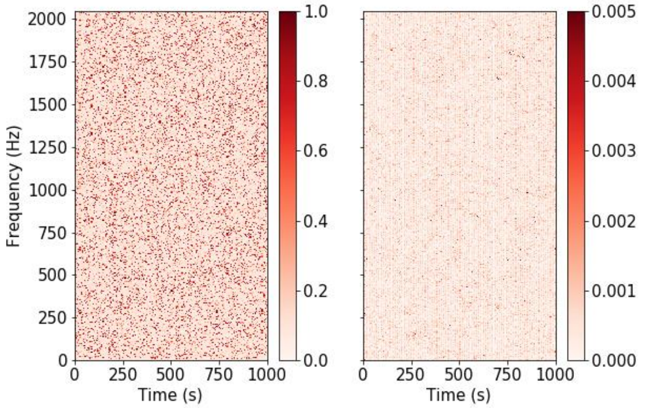
4) Early Results

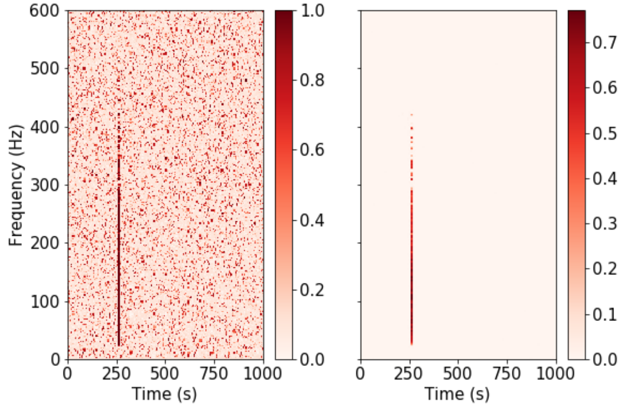
- Localization : TF maps with pure noise

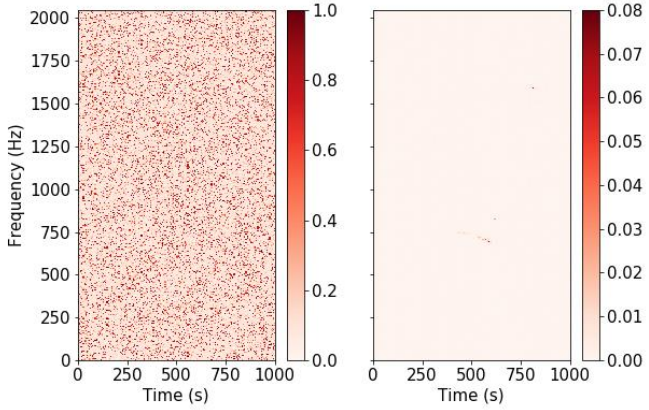
- Empty map when nothing is seen

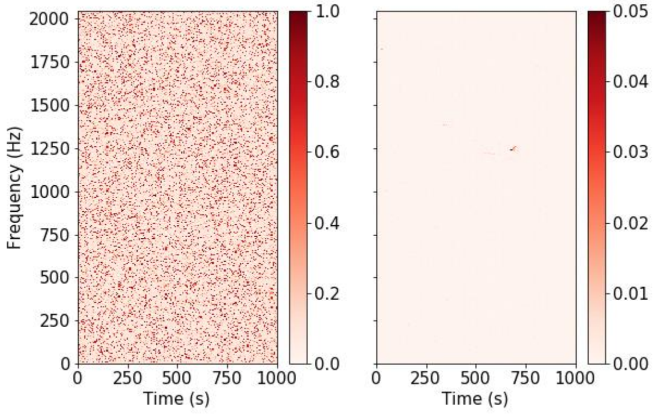
- Instrumental/environmental noise transients (glitches) are detected !











5) Improvements and future plans

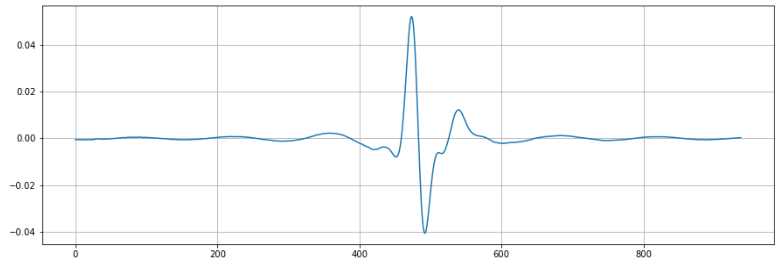
- State of the work : internal LVK review start by the end of November

- Combine the training procedure with Curriculum Learning (train with the easiest samples at first)

=> should increase the performances particularly for low amplitude injections

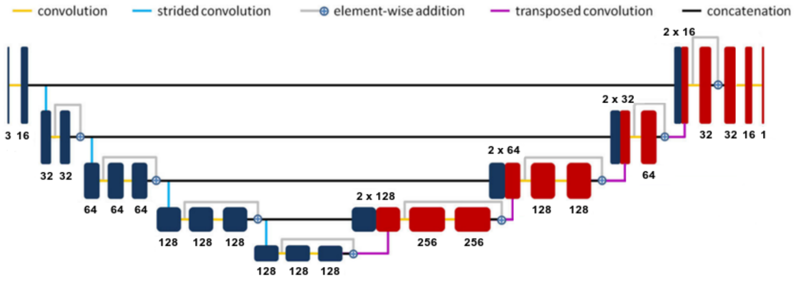
- Add a classifier to remove glitches

=> see the work of Melissa Lopez and myself (paper out soon)



- Test on new problems (can be adapted to any image shape !)

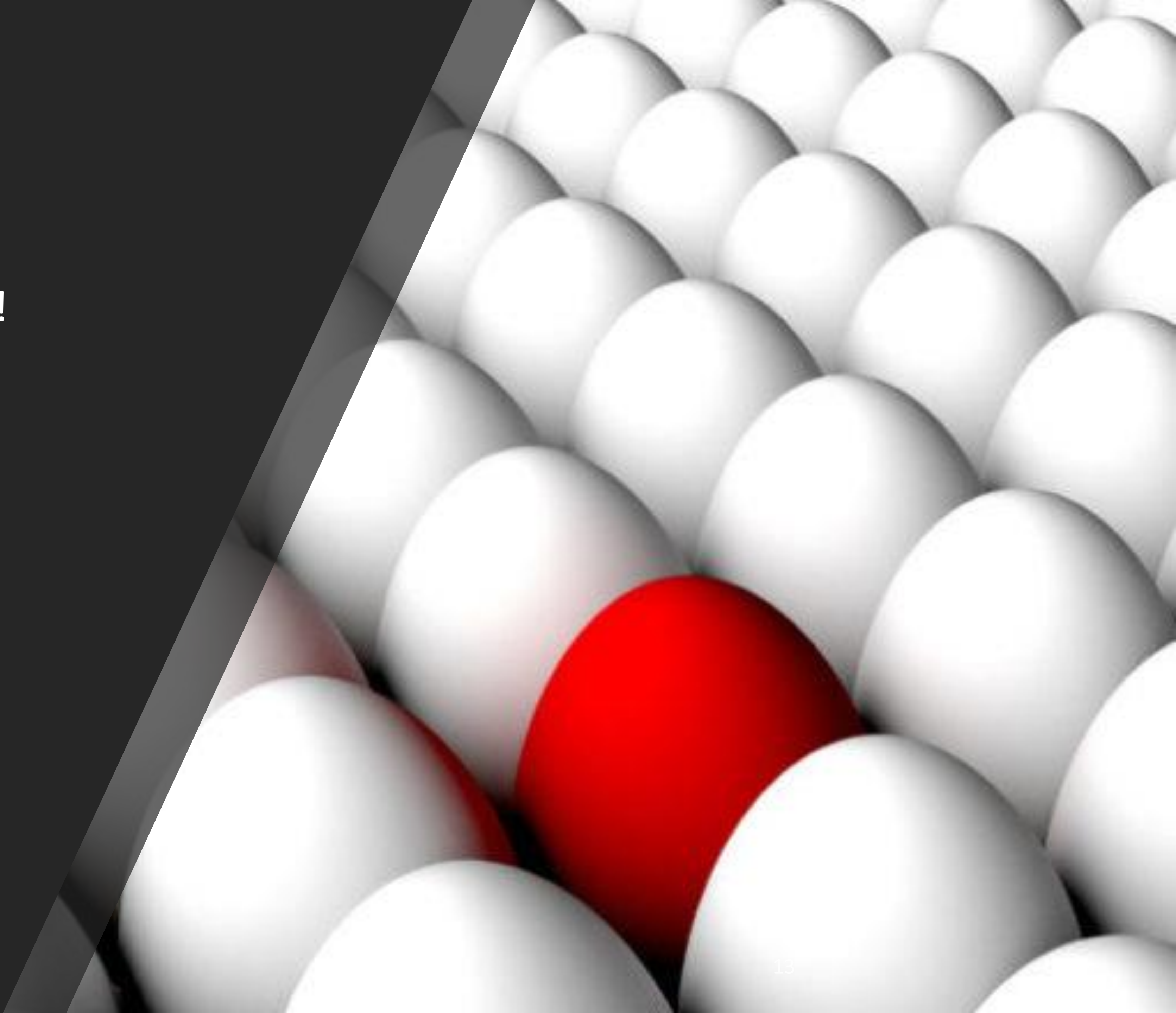
=> CBC detection, supernovae, ...



THE END

Thank you for your attention !

Questions ?



Vincent Boudart
vboudart@uliege.be