Gravitational waves & images

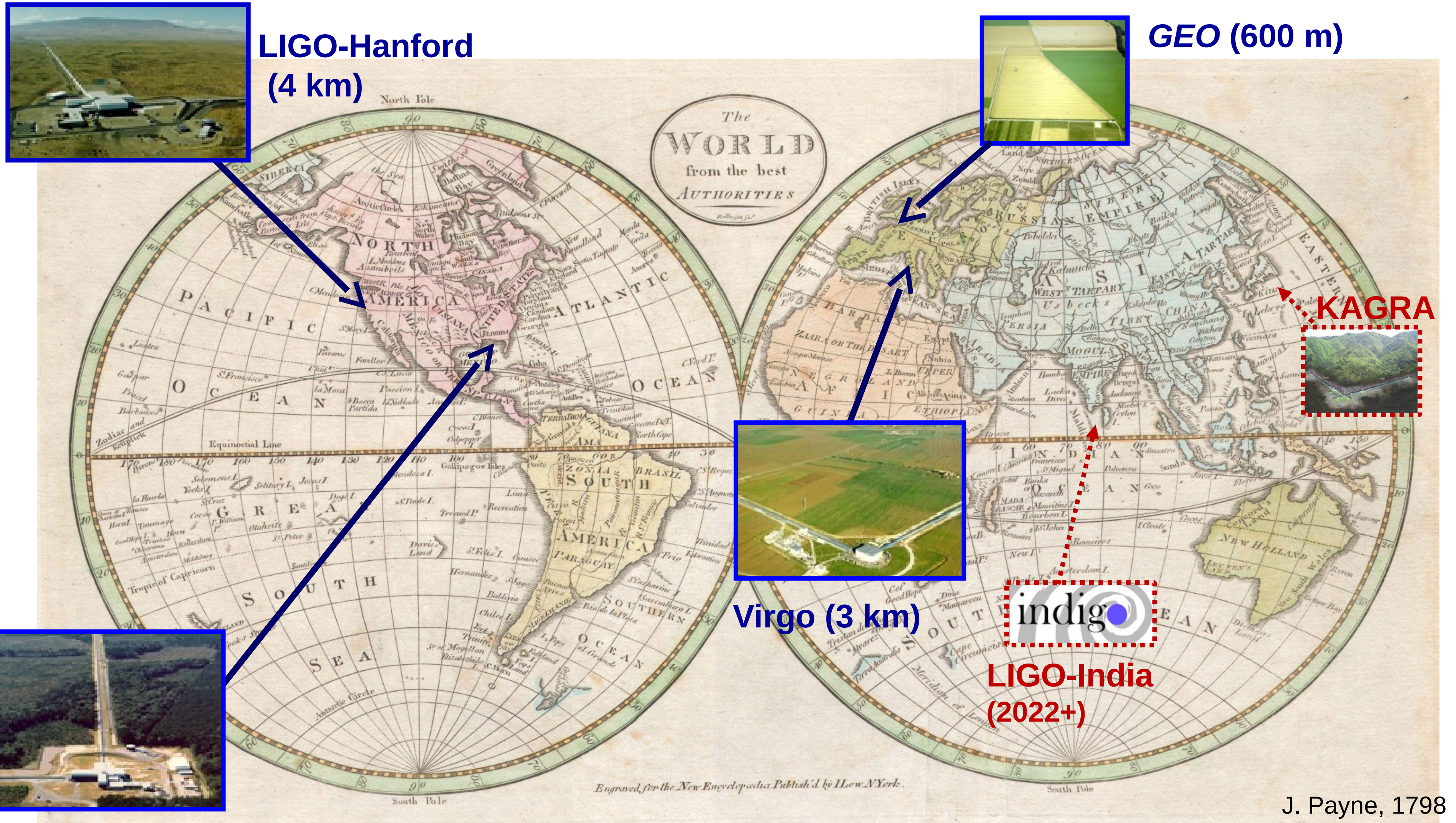
Image-based transient signal classification with deep learning

Massimiliano Razzano[1,2]

[1]University of Pisa [2]INFN-Pisa [3]EGO

1st Conference on Machine Learning for Gravitational Waves, Geophysics, Robotics, Control System
EGO, 14-16 Jan, 2019

The era of Advanced GW detectors

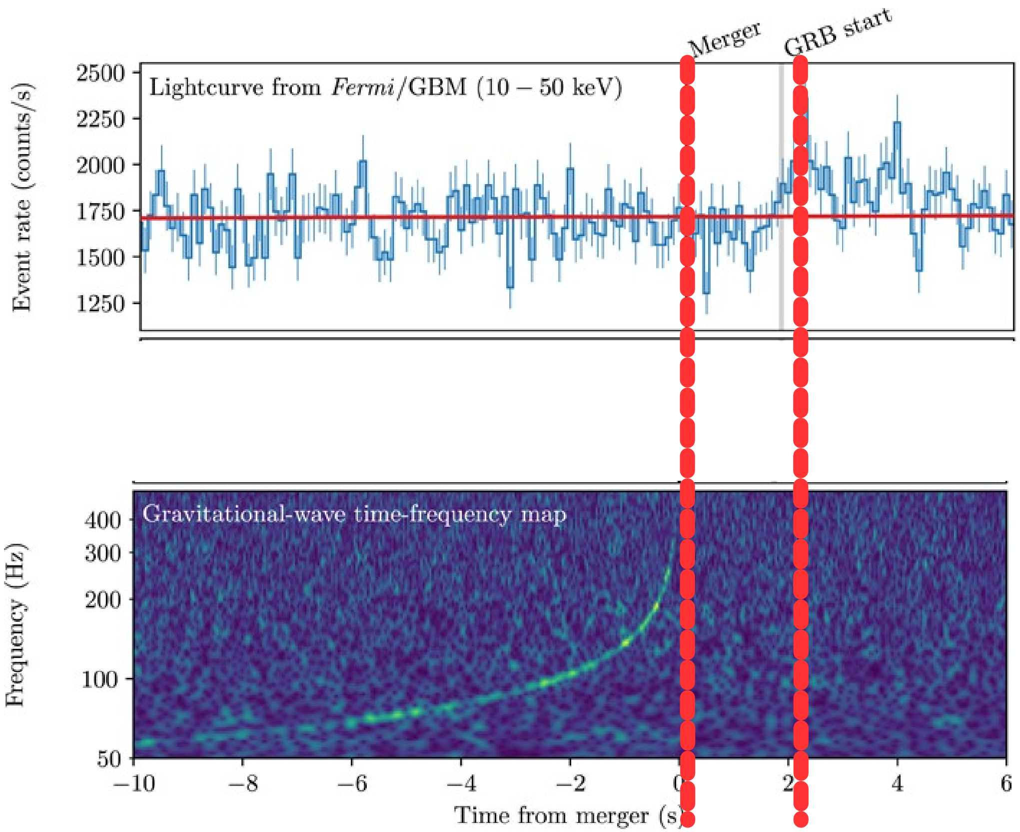


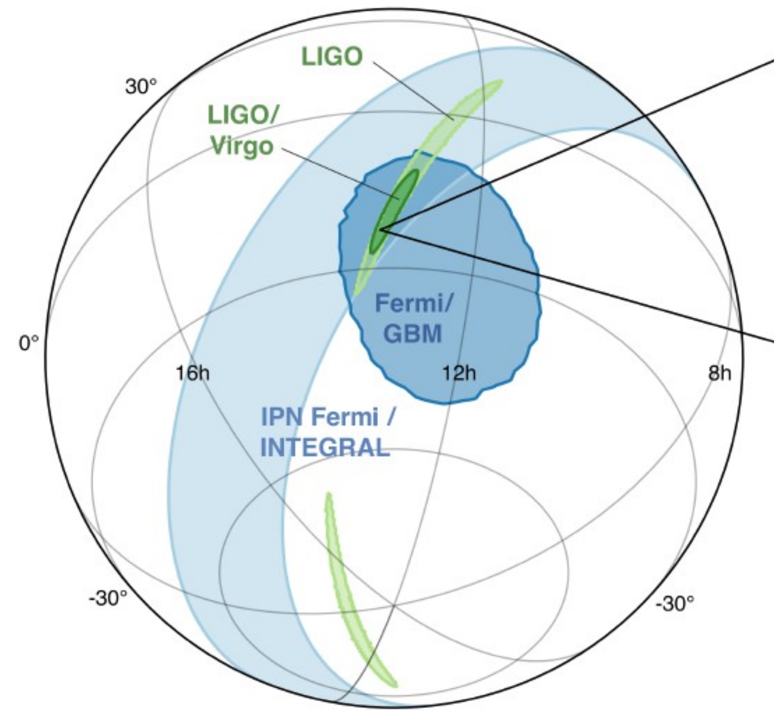
LIGO-Livingston
(4 km)

O1 and O2 ended. Looking forward to O3!

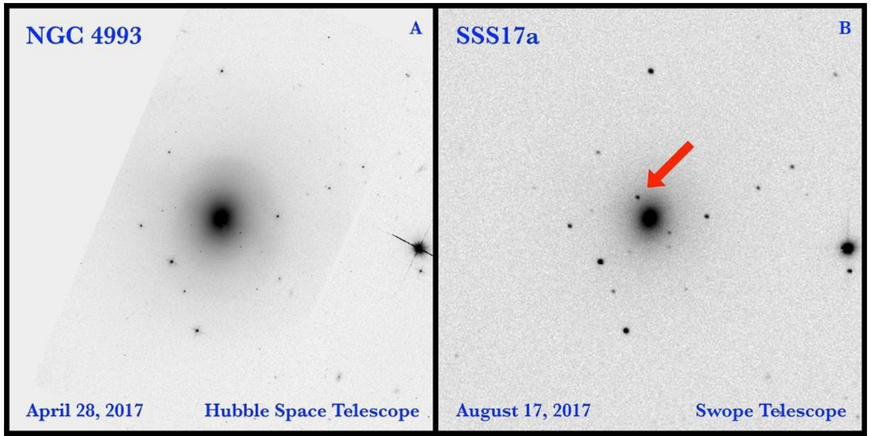
M. Razzano

GW170817 and multimessenger astronomy





Sending out fast alerts is key to EM follow-up

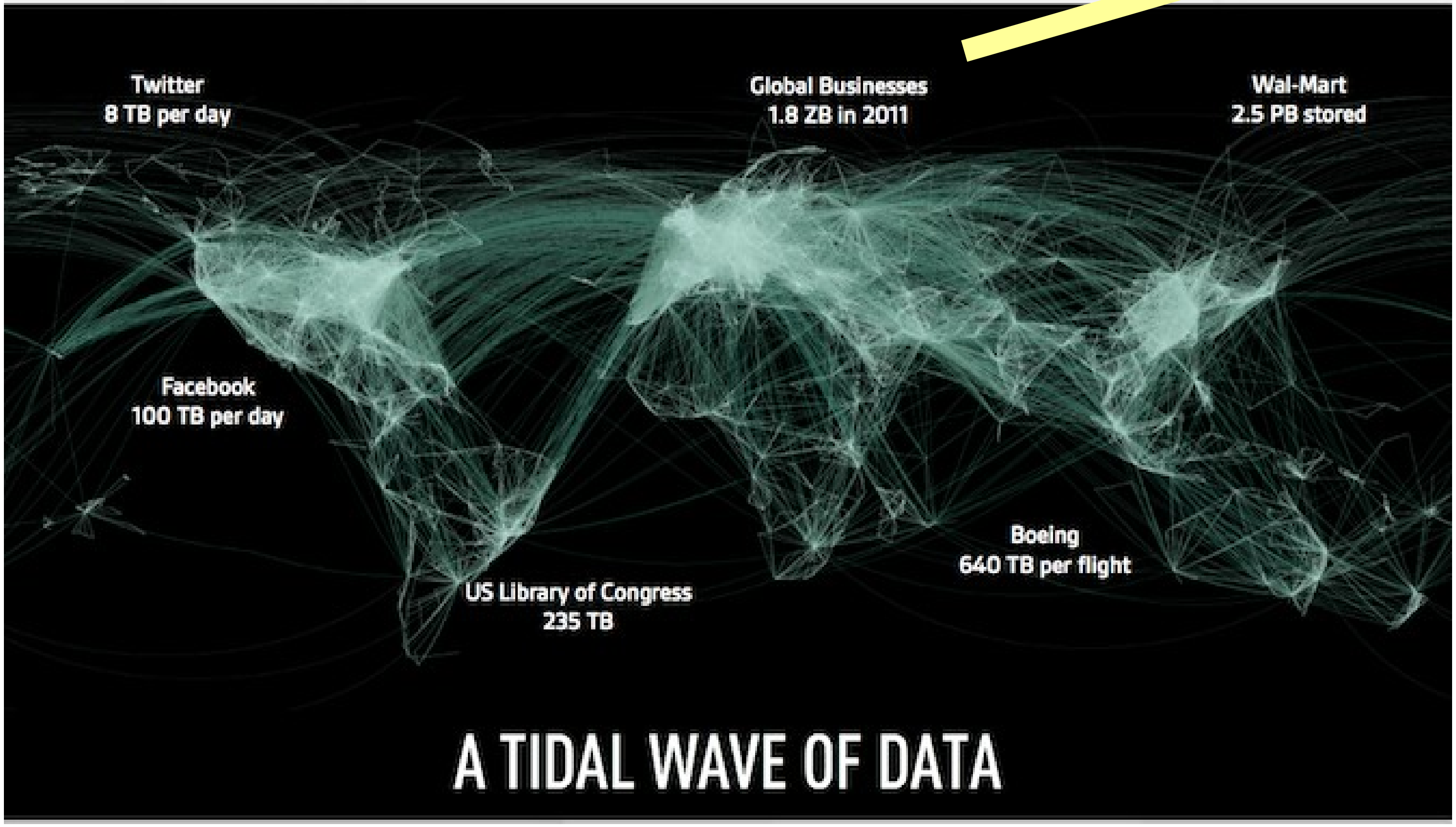


Abbott+17, PRL 119,161110
Abbott+17, ApjL,848,12
Coulter+17,Science,358,1556

The challenge of Big Data

- Not just HE physics anymore
- Sloan Digital Sky Survey : 125 TB
- Large Synoptic Survey Telescope: > 15 TB/year
- Gaia: 73 TB (+ additional data → 1 PB)

10^6 PB



This is Big Data !

Big Data in gravitational waves

- Interferometers are producing lots of data everyday
- Virgo 50 MB/s → about 0.5 TB/day from ~1000 channels
- Signals are buried in a high noise

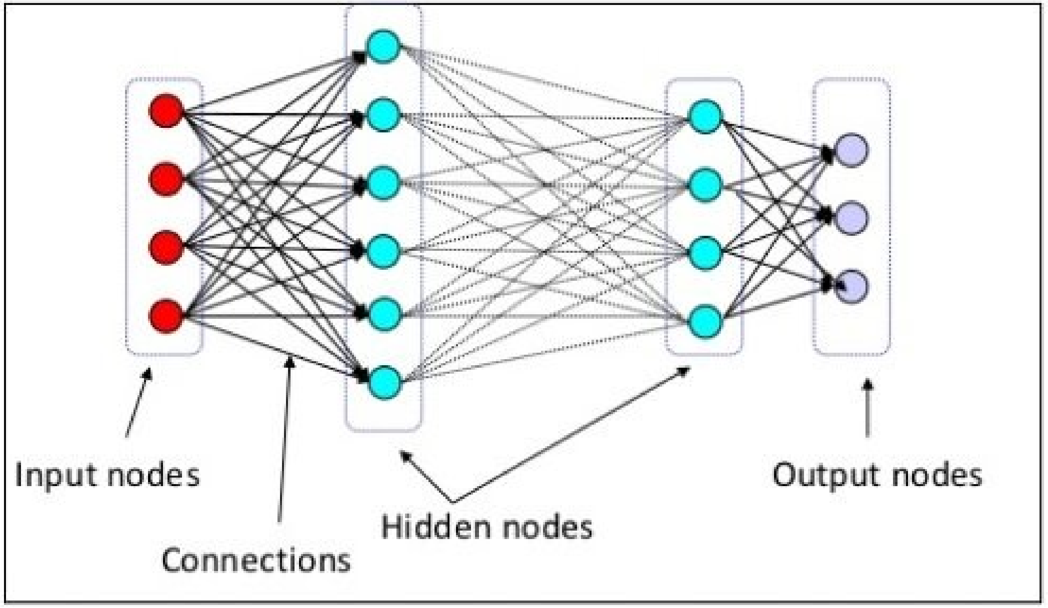
- Big data methods are required at least for 2 reasons

- On shorter timescales
  - Low-latency analysis for quick EM alert
    - Detector characterization
    - Detection and quick localization
- On longer timescales
  - Search for new sources (not just CBC but also CW etc)

Why Deep Learning?

- Deep Learning (DL) is at the frontiers of ML studies
  - Born from works on neural networks and artificial intelligence
- Combines the architecture of Neural Networks (NNs) with the power of ML
- Building block is an artificial neuron (perceptron), acting as a nonlinear processing unit
- From a single perceptron to a multilayer network of perceptrons

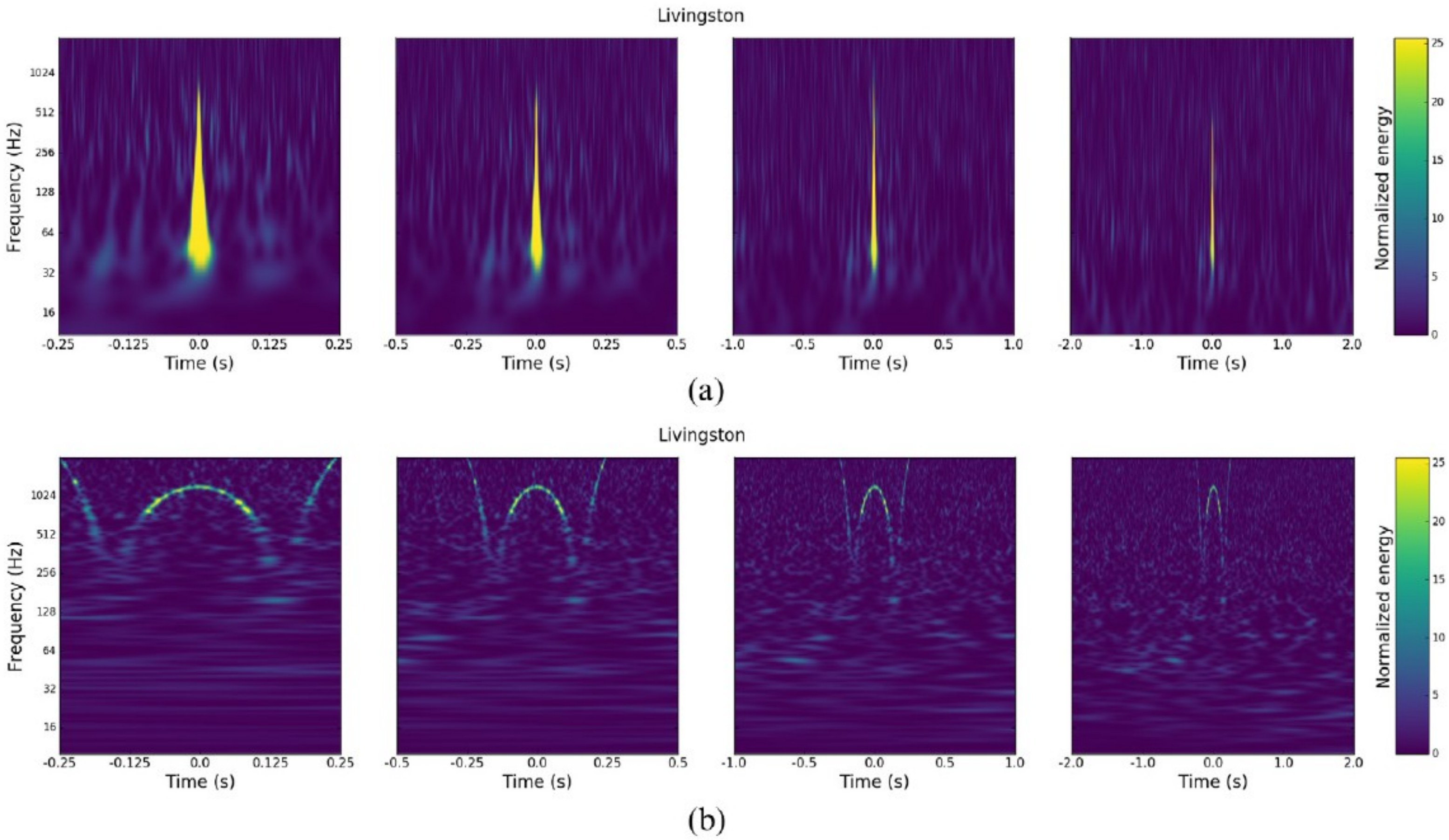
- Various projects in progress in LVC to apply ML and DL to GW studies
- In principle, a deep network can approximate any continuous function (universal approximation)



Deep learning for glitch characterization & classification

- Interferometers are limited by stationary and nonstationary noise
- Transient noise events (glitches) can impact data quality and mimic real astrophysical signals
- Detect and classify glitches is one of the most important taks for detector characterization and data analysis
- Low-latency data quality important for multimessenger follow-up
- Glitches can have complex time-frequency signatures → difficult to classify manually
- Automatic methods have been tested (e.g. Powell+15, CQG,32,215021, Mukund+17,PRD,95,104059)
- Many groups working on this in the LVC

Sample glitch gallery

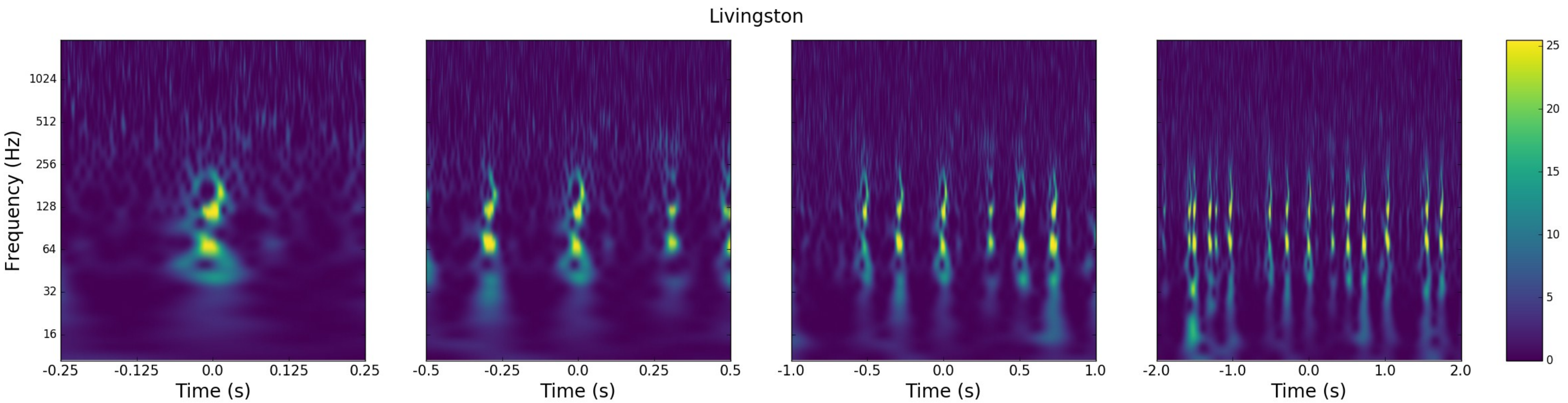


“blip” glitch

“whistle” glitch

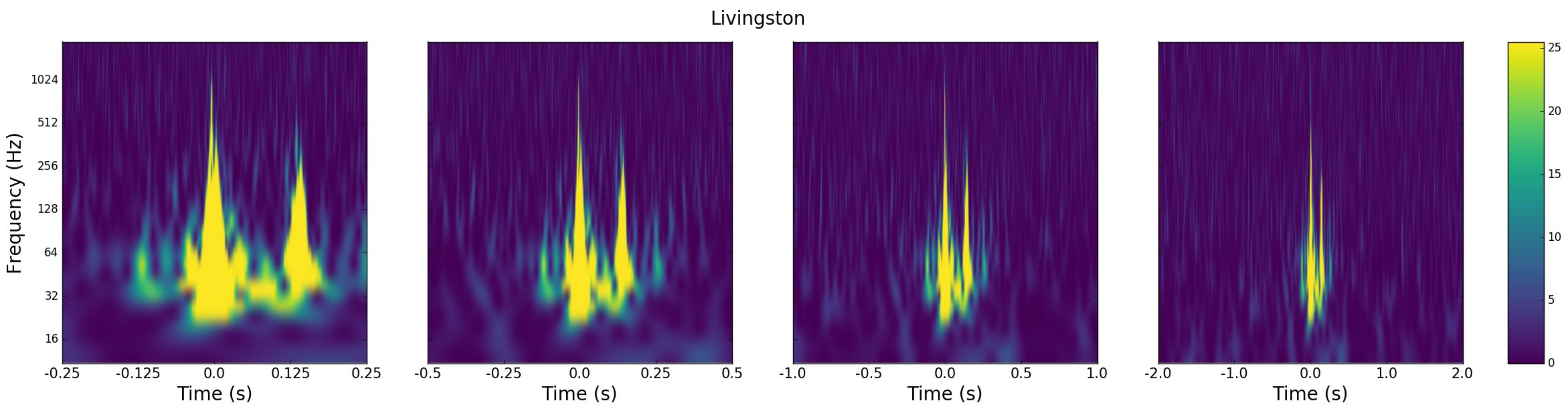
Examples of time-frequency glitch morphology (Zevin+17)

Sample glitch gallery



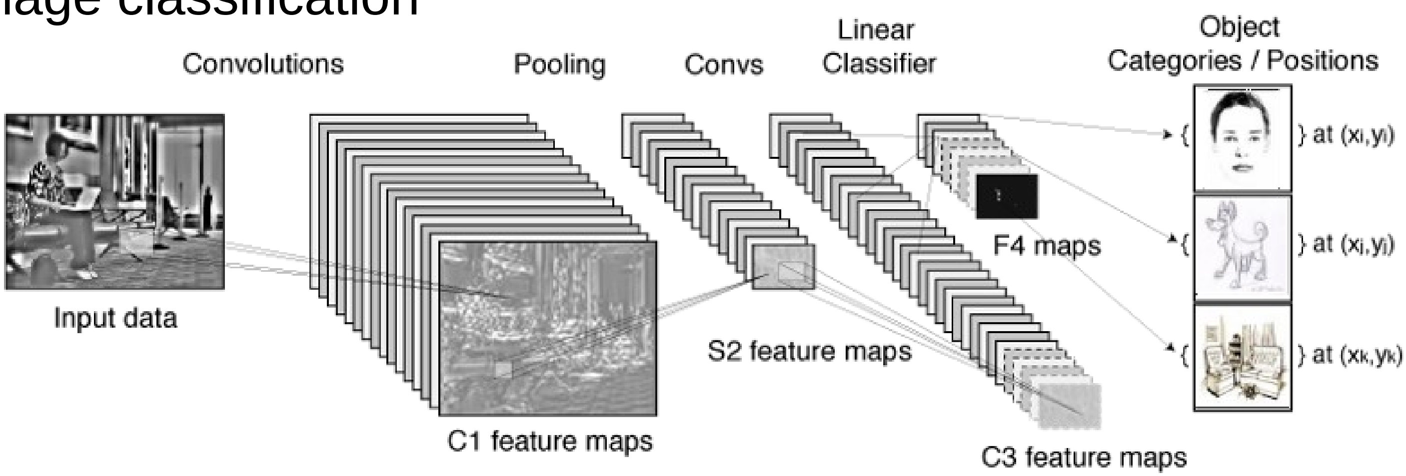
Helix glitch

Koi fish glitch



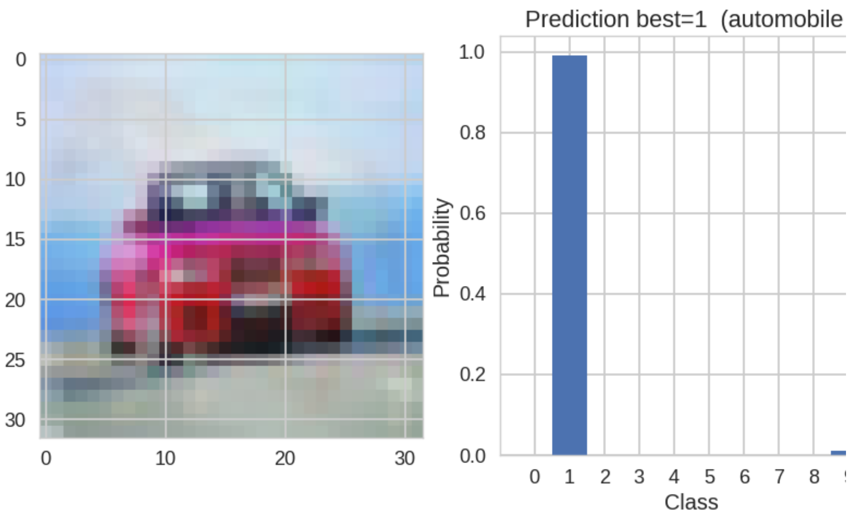
Deep Learning & glitches

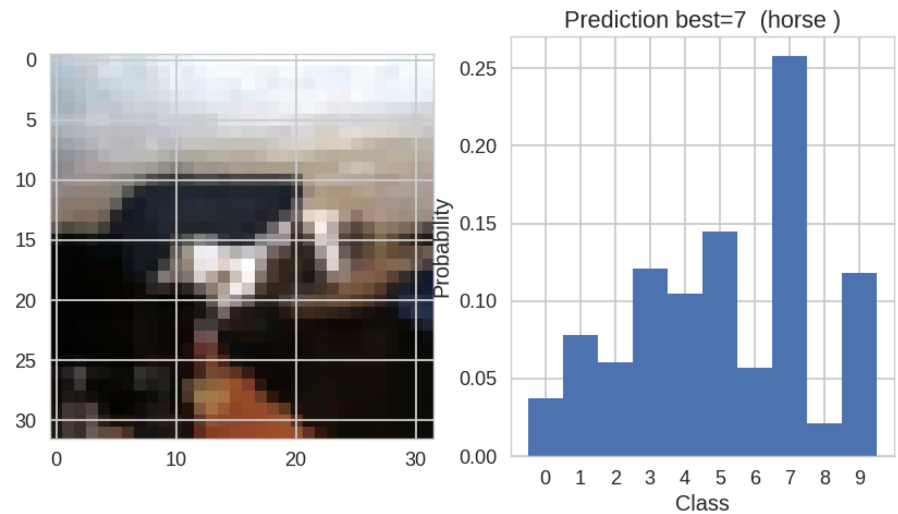
- Promising tool to classify complex patterns
- Deep network to approximate a classification function
- In our case, the function F is:
  - F: glitch GW data → glitch class
- We focus on images
  - Easy to spot signal "types" (training)
  - Compress long data stream (time-frequency)
  - Image recognition techniques
- Simple deep neural networks are not optimal (too CPU expensive)
- We use Convolutional deep Neural Networks (CNNs)
  - More complex than NNs
  - Optimized for image classification



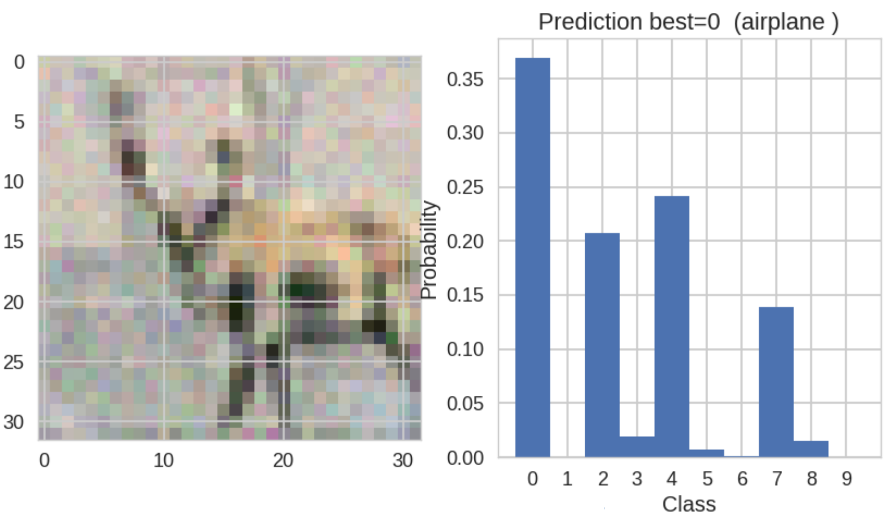
Work in collaboration with E. Cuoco (EGO, SNS)

Some first tests on general images





Not easy to spot!

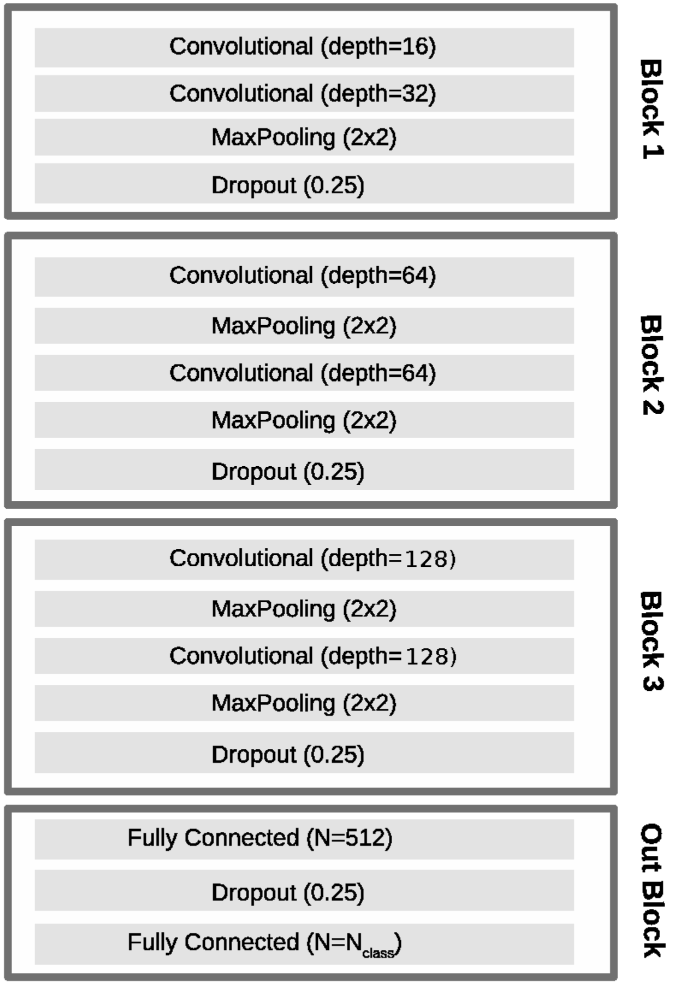


Deer (4)

Python libraries
(Keras+TensorFlow)
Run on GPU

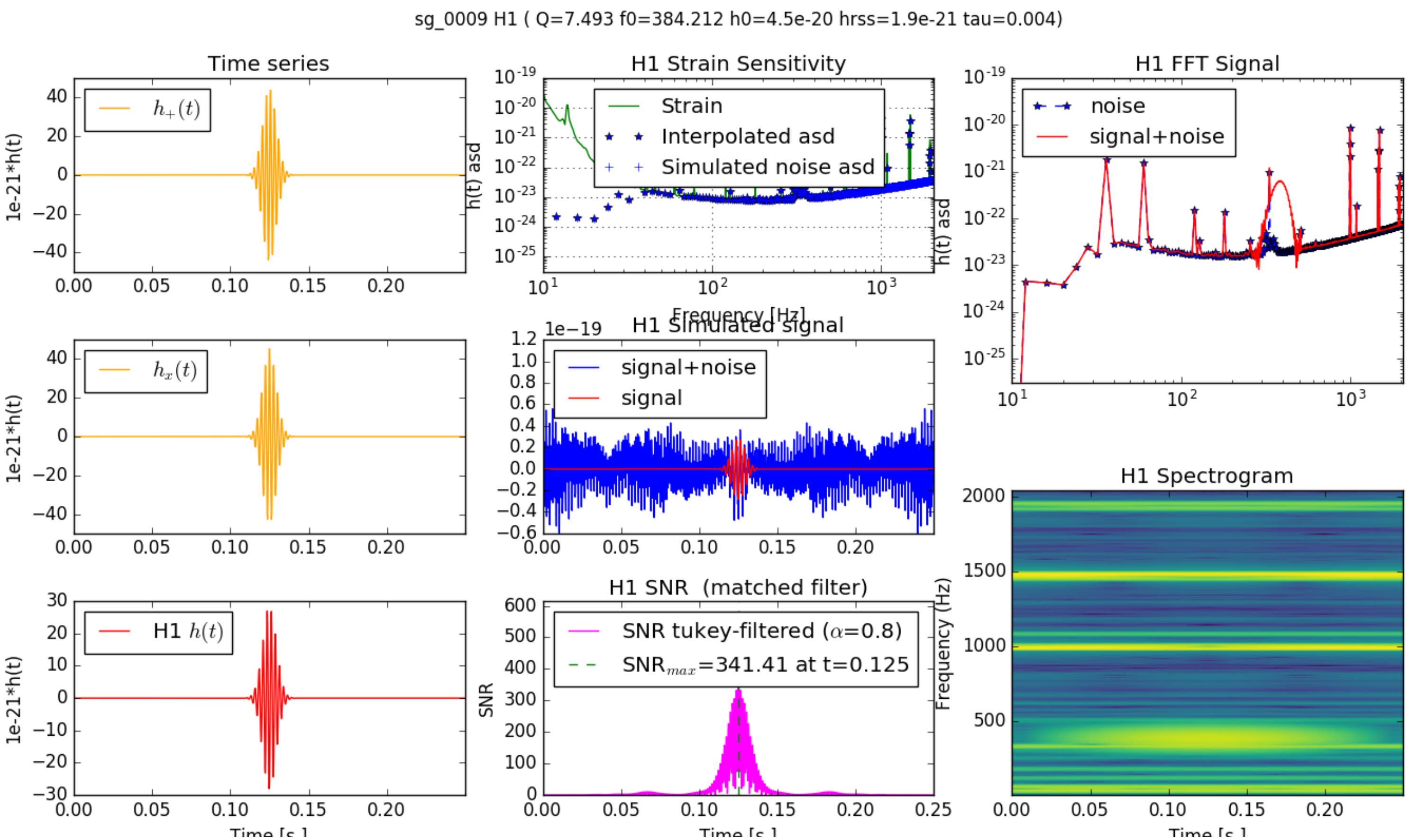
Our running configuration

- Input GW data
  - Image processing
  - Time series whitening
  - Image creation from time series (FFT spectrograms)
  - Image equalization & contrast enhancement
- Classification
  - A probability for each class, take the max
  - Add a NOISE class to crosscheck glitch detection
- Network layout
  - Tested various networks, including a 4-block layers
- Run on GPU Nvidia GeForce GTX 780
  - 2.8k cores, 3 Gb RAM)
  - Developed in Python + CUDA-optimized libraries



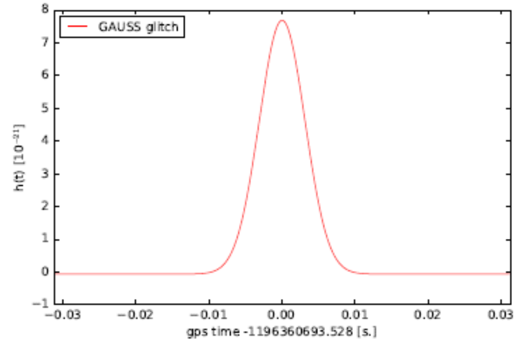
Tests on simulations (I)

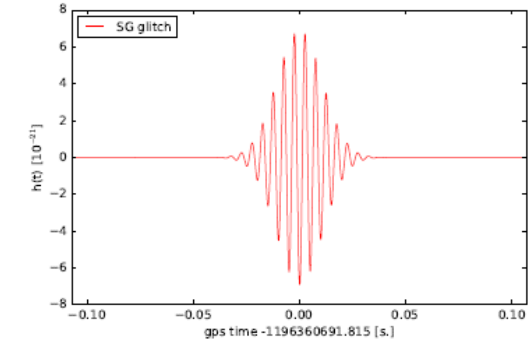
- To test the pipeline, we prepared ad-hoc simulations
- Simulate colored noise using public H1 sensitivity curve
- Add 6 different classes of glitch shapes

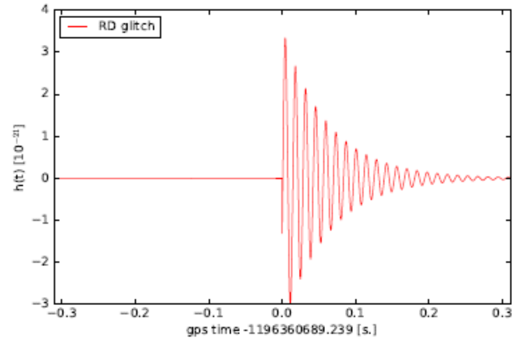


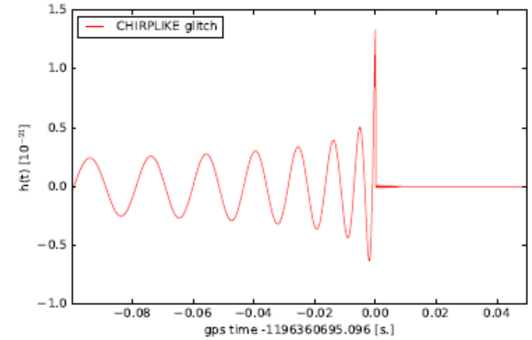
Tests on simulations (II)

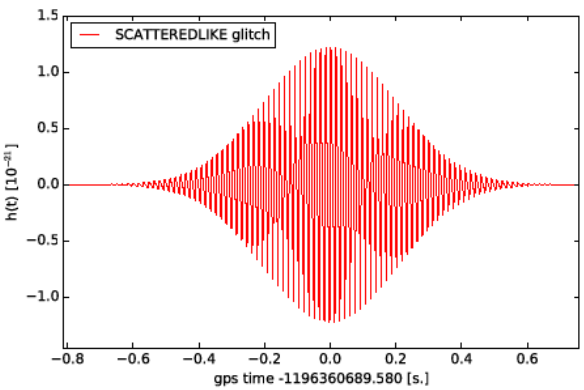
Simulated families

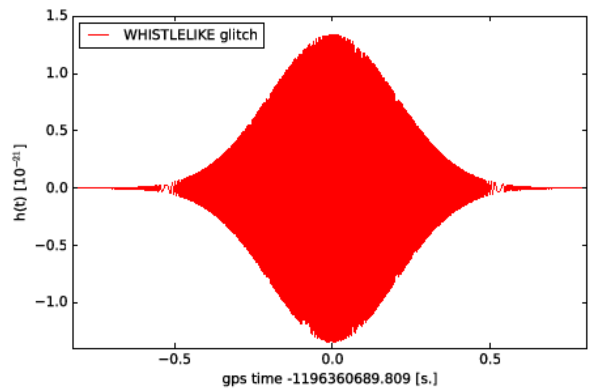








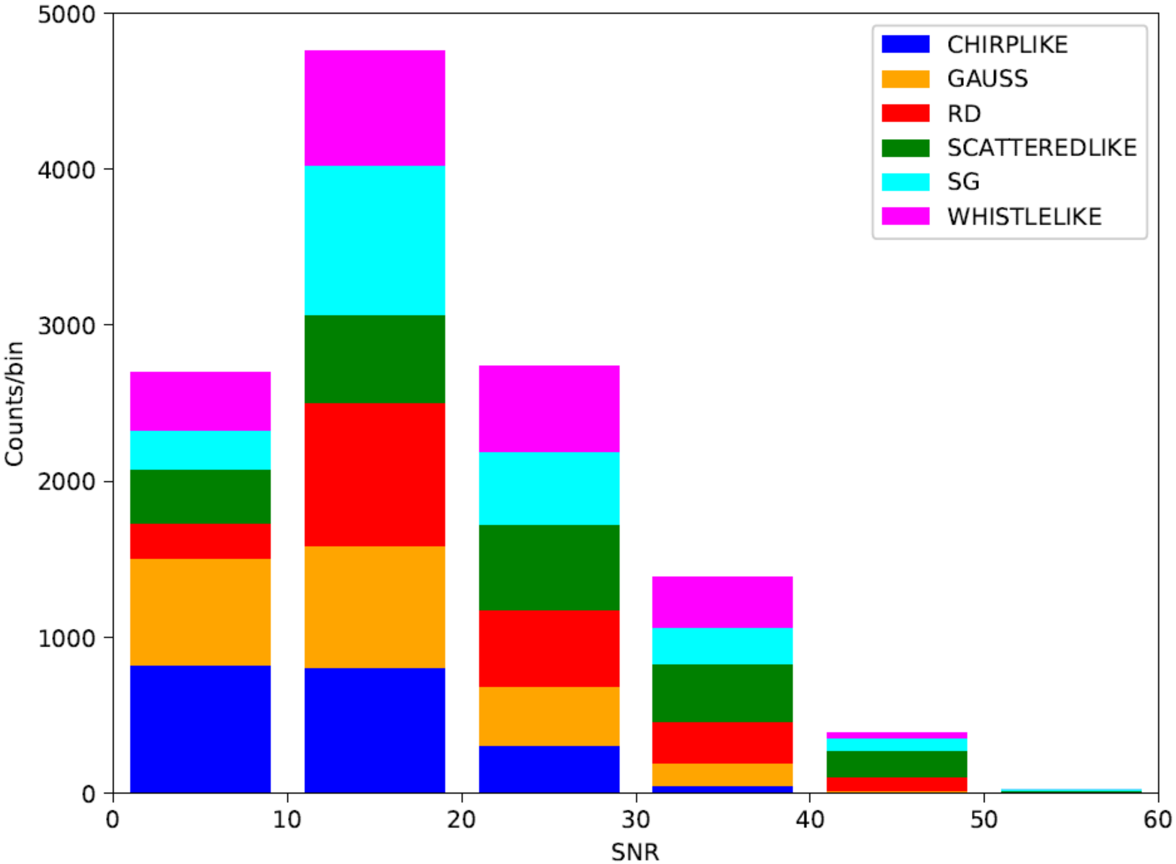




To show the glitch timeseries here we don't show the noise contribution

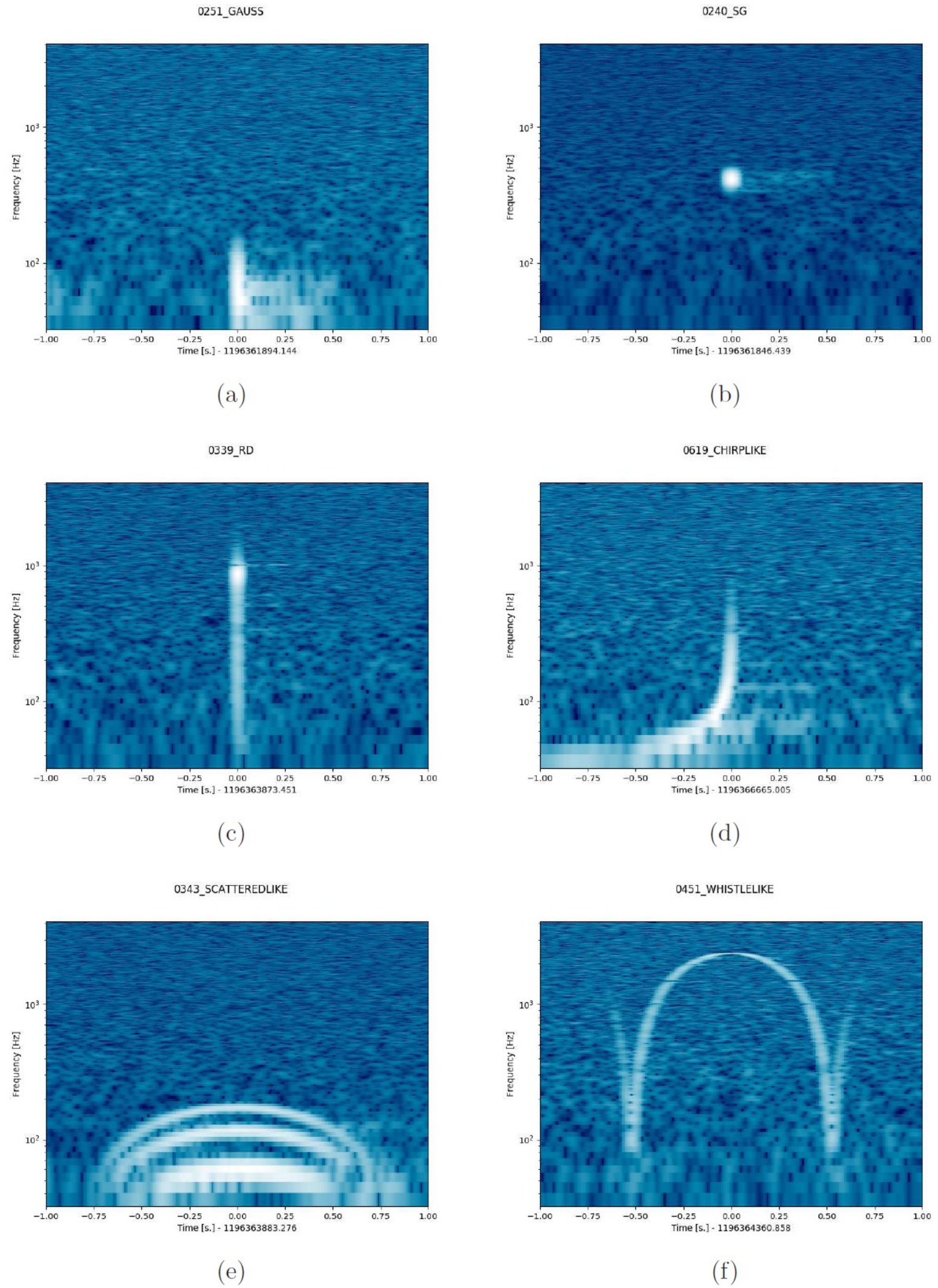
Tests on simulations (III)

- Simulated time series with 8kHz sampling rate
- Glitches distributed with Poisson statistics μ=0.5 Hz
- 2000 glitches per each family
- Glitch parameters are varied randomly to achieve various shapes and Signal-To-Noise ratio



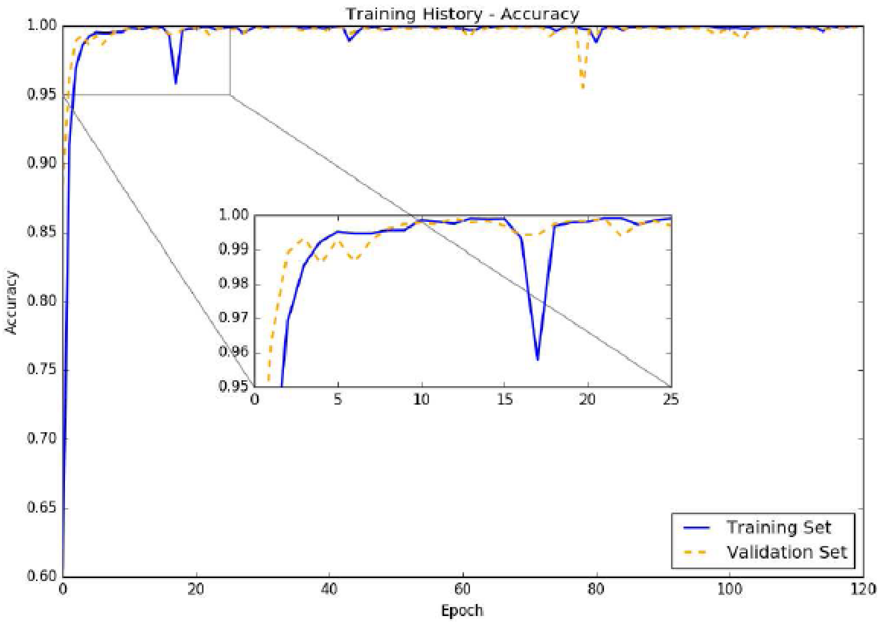
Building the images

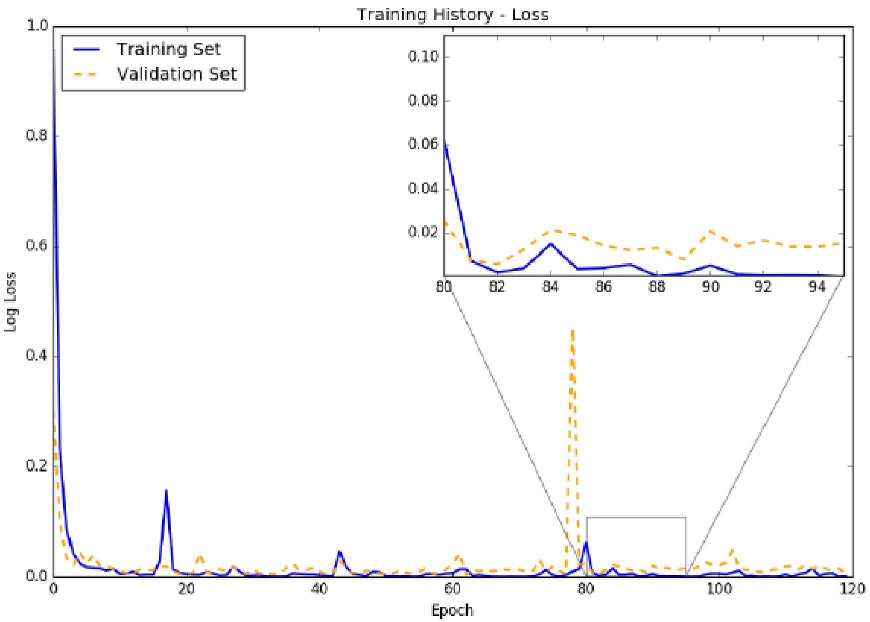
- Spectrogram for each image
- 2-seconds time window
to highlight fatures in long glitches
- Data is whitened
- Optional contrast stretch



Training the CNN

- Datasets of 14000 images
- Training:validation:test → 75:15:15
- Image size 241 x 513
- Reduced the images by a factor 0.55 due to memory constraints
- Use validation set to tune hyperparameters
- On our hardware, training time ~8 hrs for ~100 epochs
- When training is done, classification requires ~1 ms/image (on our configuration)





Classification results – metrics

- We compared classification performances with simpler architectures

Metric	Accuracy	Precision	Recall	F1 score	Log loss
SVM	0.971	0.972	0.971	0.971	0.08
Shallow CNN	0.986	0.986	0.986	0.986	0.04
1 CNN block	0.991	0.991	0.991	0.991	0.02
3 CNN blocks	0.998	0.998	0.998	0.998	0.008

Linear Support Vector Machine

CNN with 1 hidden layer

CNN with one block
(2 CNNs+Pooling&Dropout)

Deep 4-blocks CNNs

Classification results

Normalized Confusion Matrix

SVM

True class \ Predicted class	CHIRPLIKE	GAUSS	NOISE	RD	SCATTEREDLIKE	SG	WHISTLELIKE
CHIRPLIKE	1.000	0.000	0.000	0.000	0.000	0.000	0.000
GAUSS	0.000	0.976	0.000	0.024	0.000	0.000	0.000
NOISE	0.000	0.000	1.000	0.000	0.000	0.000	0.000
RD	0.000	0.034	0.000	0.917	0.000	0.046	0.003
SCATTEREDLIKE	0.000	0.000	0.000	0.000	1.000	0.000	0.000
SG	0.000	0.006	0.000	0.089	0.000	0.905	0.000
WHISTLELIKE	0.000	0.000	0.000	0.000	0.000	0.000	1.000

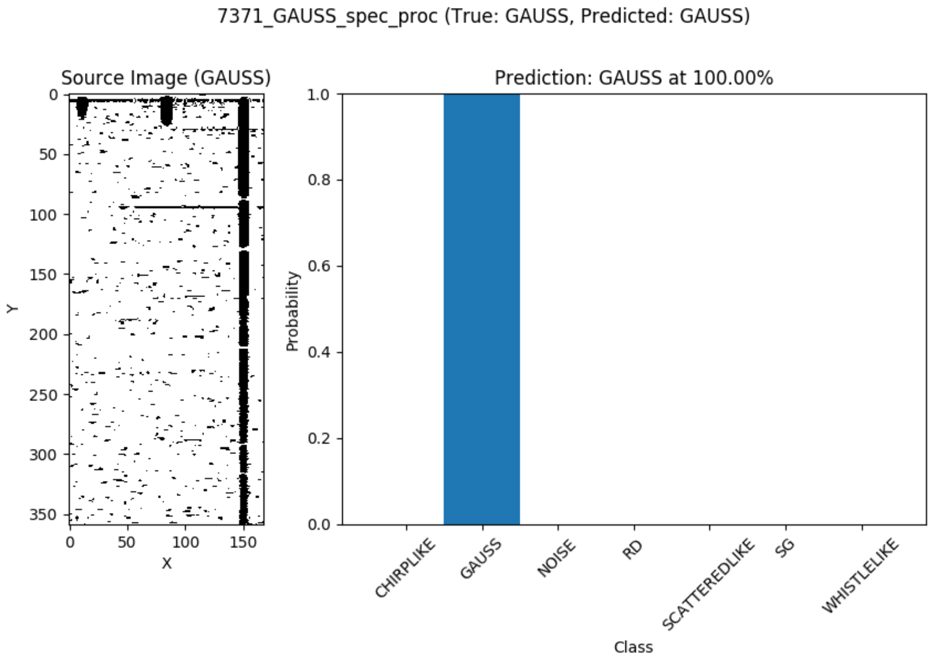
Deep CNN

True class \ Predicted class	CHIRPLIKE	GAUSS	NOISE	RD	SCATTEREDLIKE	SG	WHISTLELIKE
CHIRPLIKE	1.000	0.000	0.000	0.000	0.000	0.000	0.000
GAUSS	0.000	0.997	0.003	0.000	0.000	0.000	0.000
NOISE	0.000	0.000	1.000	0.000	0.000	0.000	0.000
RD	0.000	0.003	0.000	0.994	0.000	0.003	0.000
SCATTEREDLIKE	0.000	0.000	0.000	0.000	1.000	0.000	0.000
SG	0.000	0.000	0.000	0.003	0.000	0.997	0.000
WHISTLELIKE	0.000	0.000	0.000	0.000	0.000	0.000	1.000

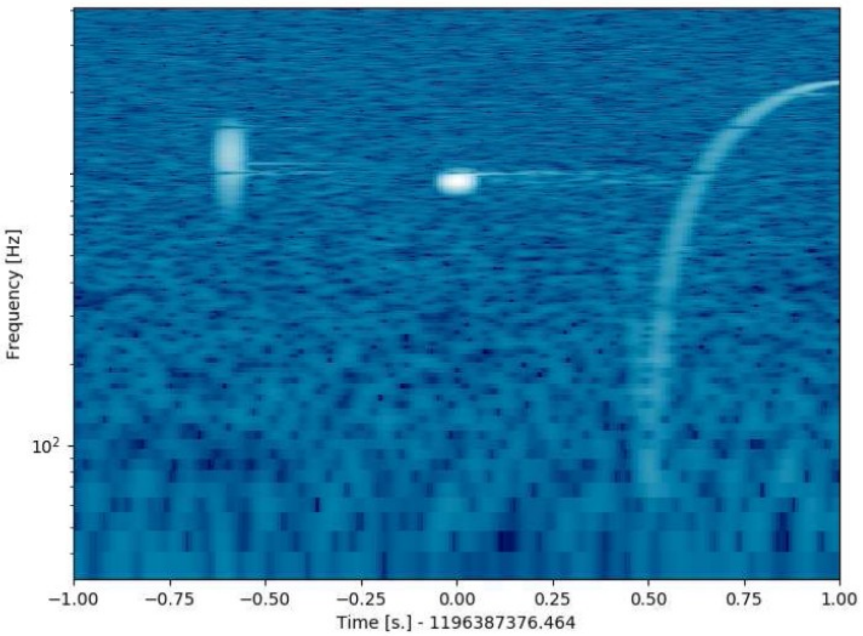
Deep CNN better at distinguishing similar morphologies

Classification results

Some cases of more glitches in the time window, always identify the right class



100% Sin-Gauss



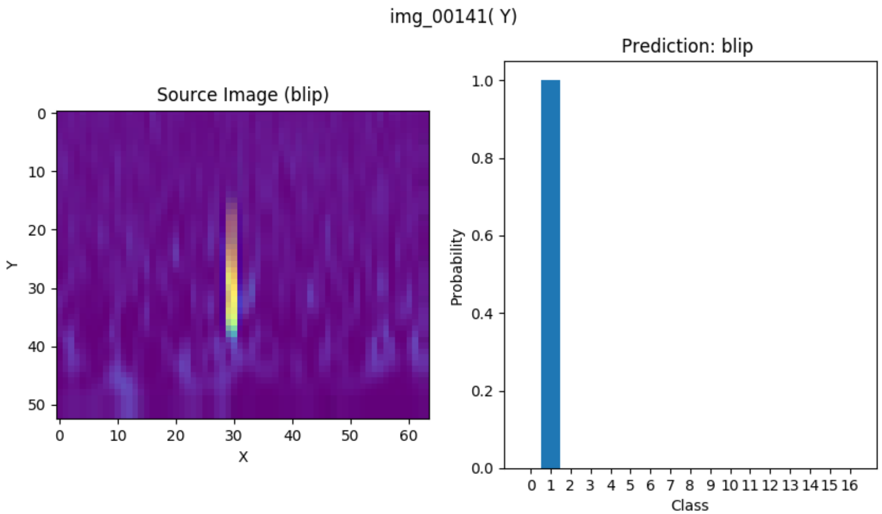
More details in
Razzano & Cuoco 2018, CQG,35,9
(arxiv:1803.09933)

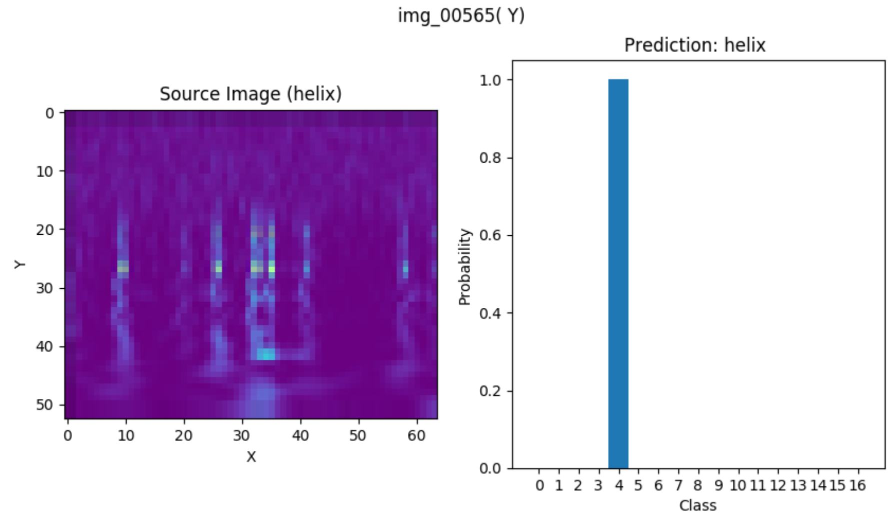
Run on O1 dataset

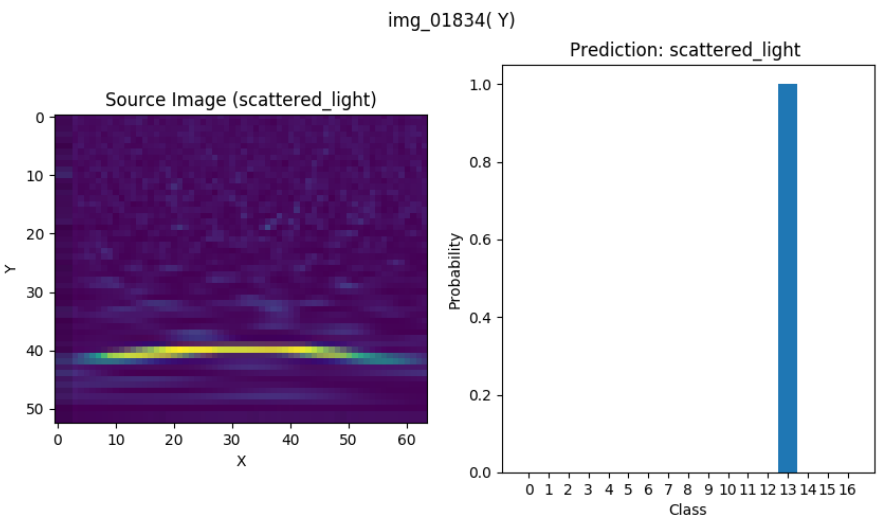
Glitch name	# in H1	# in L1
Air compressor	55	3
Blip	1495	374
Chirp	34	32
Extremely Loud	266	188
Helix	3	276
Koi fish	580	250
Light Modulation	568	5
Low_frequency_burst	184	473
Low_frequency_lines	82	371
No_Glitch	117	64
None_of_the_above	57	21

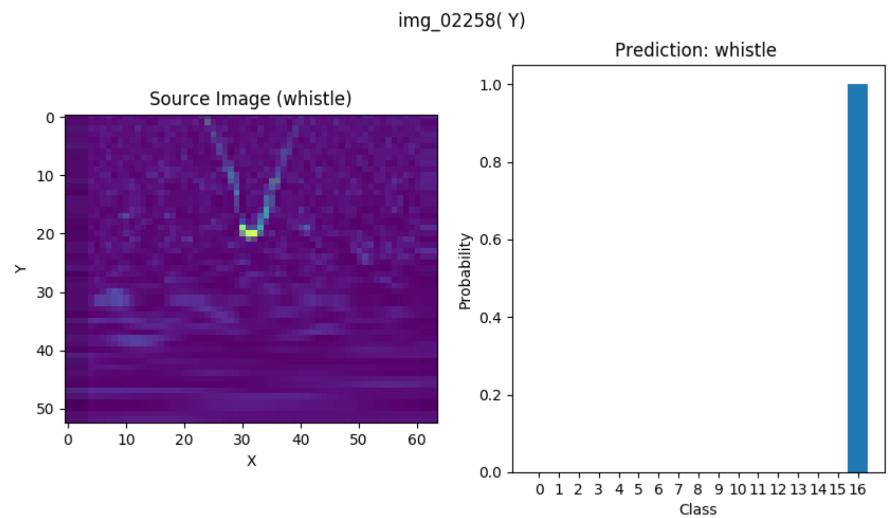
Glitch name	# in H1	# in L1
Paired doves	27	-
Power_line	274	179
Repeating blips	249	36
Scattered_light	393	66
Scratchy	95	259
Tomte	70	46
Violin_mode	179	-

Sample results

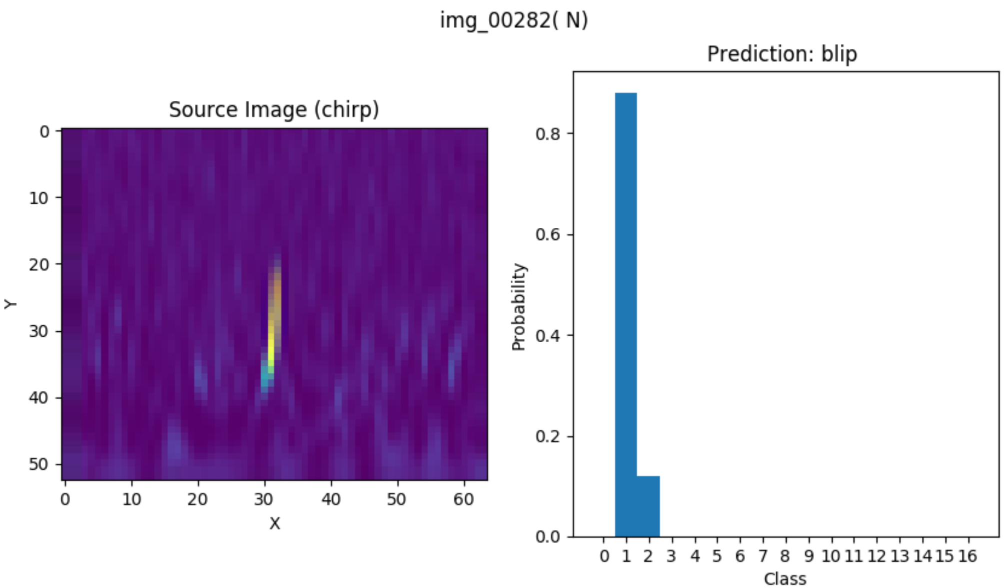




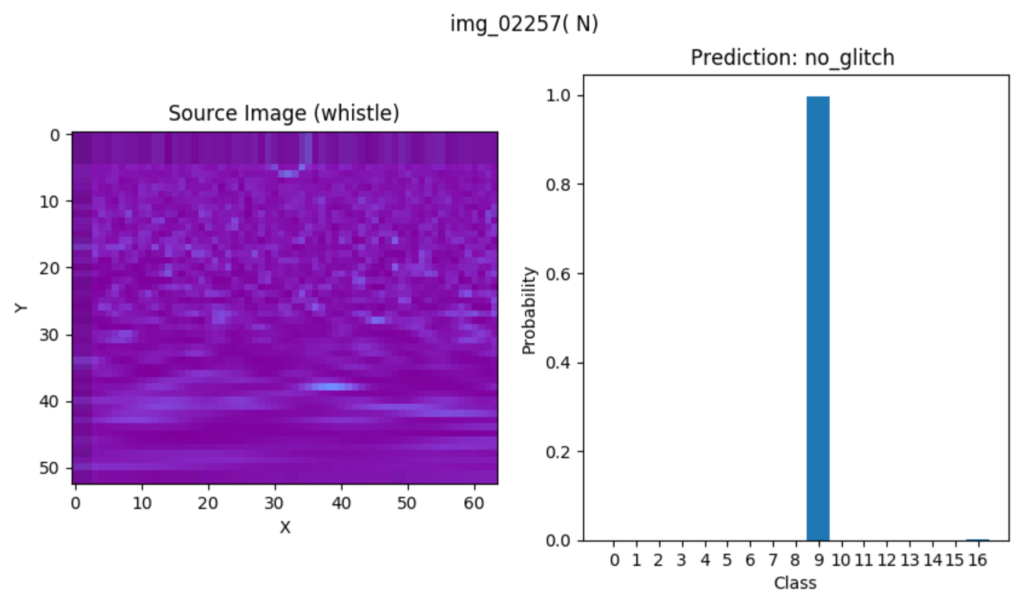




Sample misclassifications



Here the problem is the zoom on the image

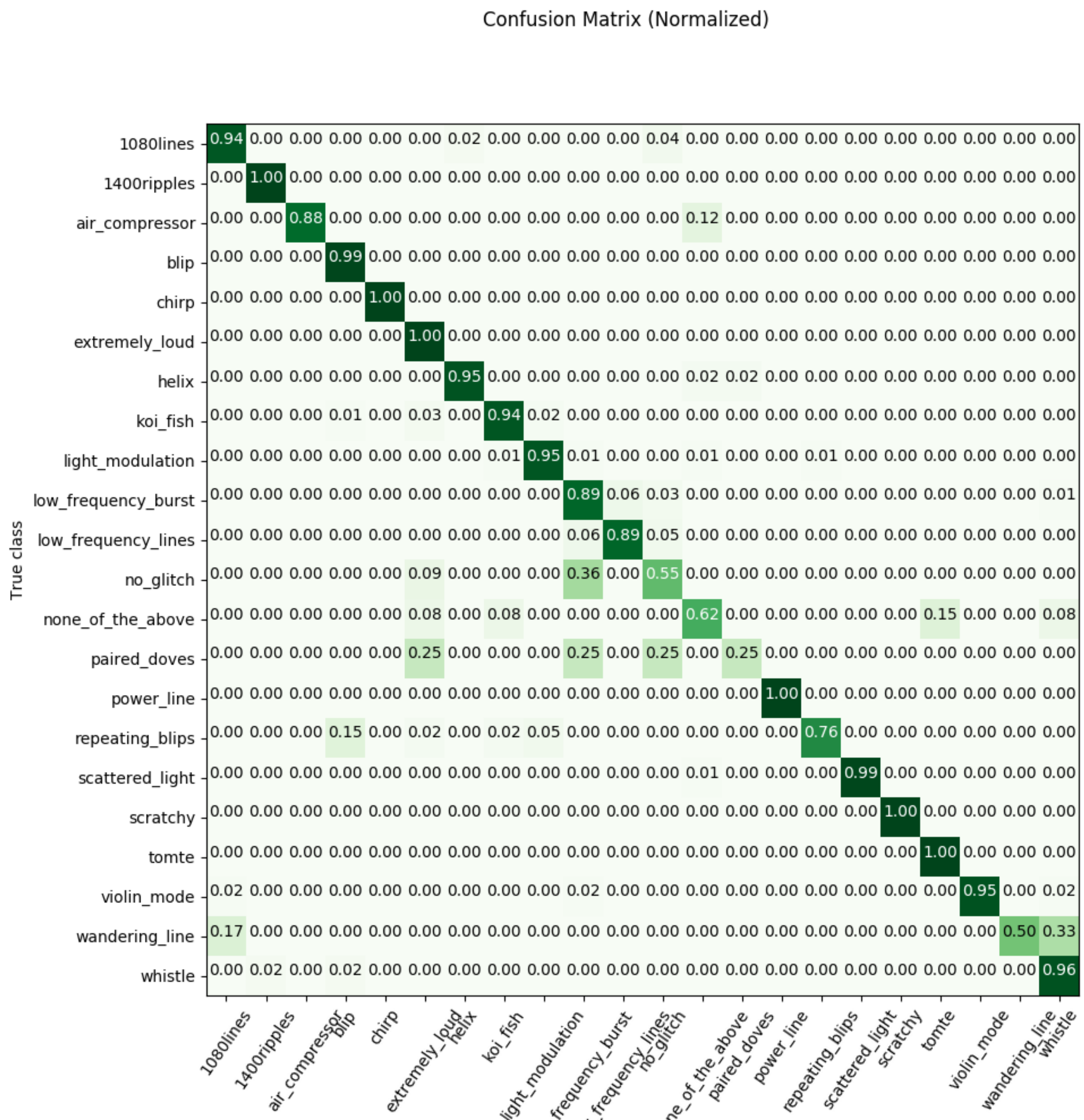


- Here the problem is the poor contrast
- We can solve with whitening

More examples available

Classification results

Normalized
Confusion Matrix



1 CNN block

Classification results

Confusion Matrix (Normalized)

True class

	1080lines	1400ripples	air_compressor	blip	chirp	extremely_loud	helix	koi_fish	light_modulation	low_frequency_burst	low_frequency_lines	no_glitch	none_of_the_above	paired_doves	power_line	repeating_blips	scattered_light	scratchy	tomte	violin_mode	wandering_line	whistle
1080lines	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1400ripples	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
air_compressor	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
blip	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
chirp	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
extremely_loud	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
helix	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
koi_fish	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
light_modulation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
low_frequency_burst	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.03	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
low_frequency_lines	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.88	0.05	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00
no_glitch	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
none_of_the_above	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69	0.00	0.00	0.00	0.00	0.08	0.15	0.00	0.00	0.08
paired_doves	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
power_line	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
repeating_blips	0.00	0.00	0.00	0.05	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00
scattered_light	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.01	0.00	0.00	0.00	0.00
scratchy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
tomte	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
violin_mode	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00
wandering_line	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
whistle	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.91

Normalized Confusion Matrix

Full CNN stack

Interactive Glitch Web Catalog

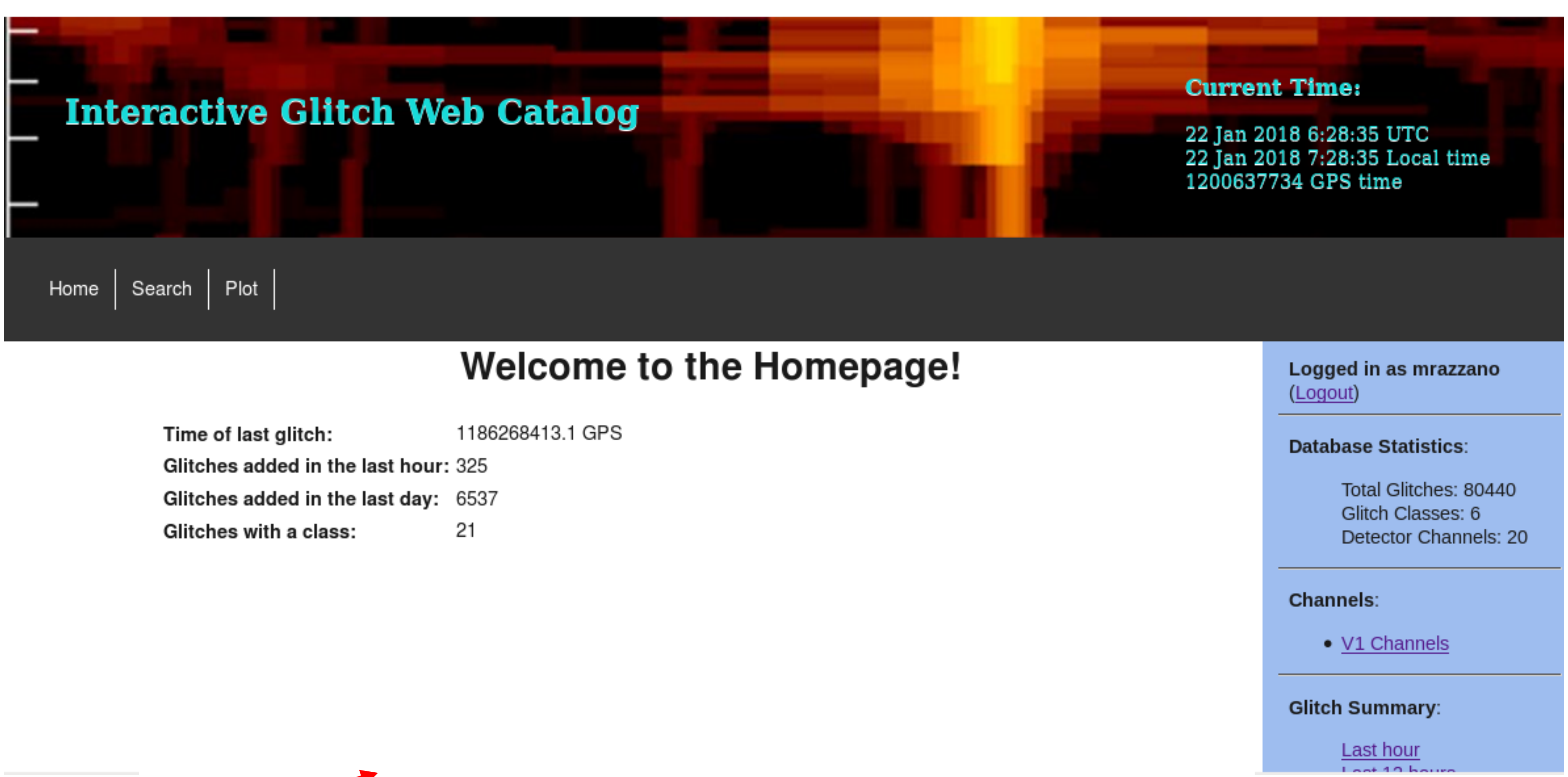
- Web tool to interactively work on glitches
- Main goals
  - Database for glitches
  - Interface with Omicron (and other pipelines if needed)
  - Quick look & quick analysis
  - Label glitches
  - Store automatic glitch classification
- Developed in Python + Django + MySQL
- Tested and working on Virgo glitches

Interactive Glitch Web Catalog

- Accessible online at a EGO machine

Real Time clock (UTC, local,GPS)

Main menu

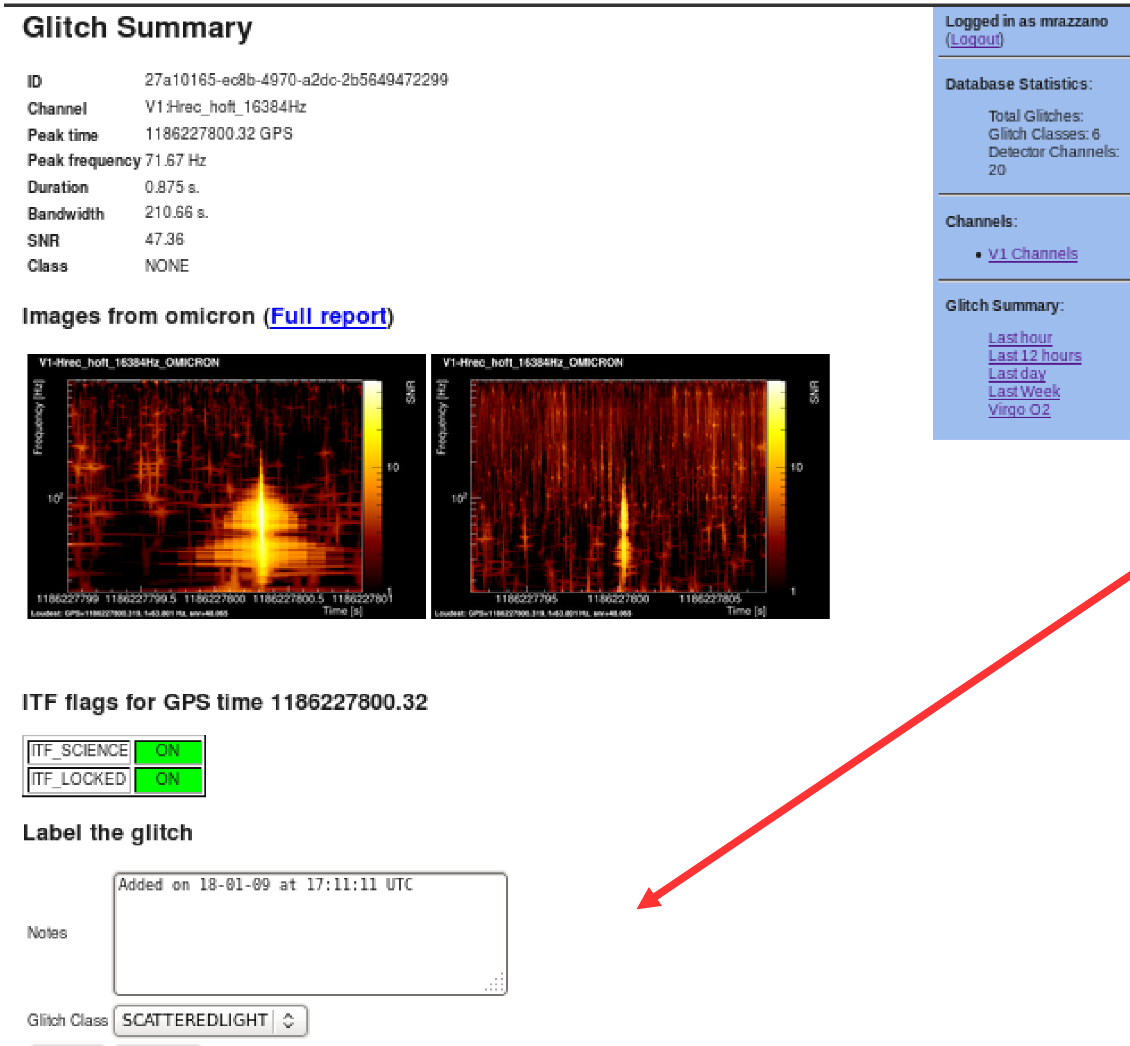


Main panel

Sidebar
With global statistics

Label glitches

By clicking in the plots or in the search results, a summary of the glitch is visible



Users can log in and then access to the labeling functionality

Check and plotting depending on labels

Conclusions and next steps

- Machine and deep learning methods are growing fast in GW community
- We have tested and developed image-based deep learning for classification of noise glitches
- Time-frequency images as input data
- Tested on simulatons & real data
  - Run on small GPU hardware
  - High accuracy
- Also interact with other ML techniques (e.g. Elena’s WDFX)
- Toward a real-time pipeline