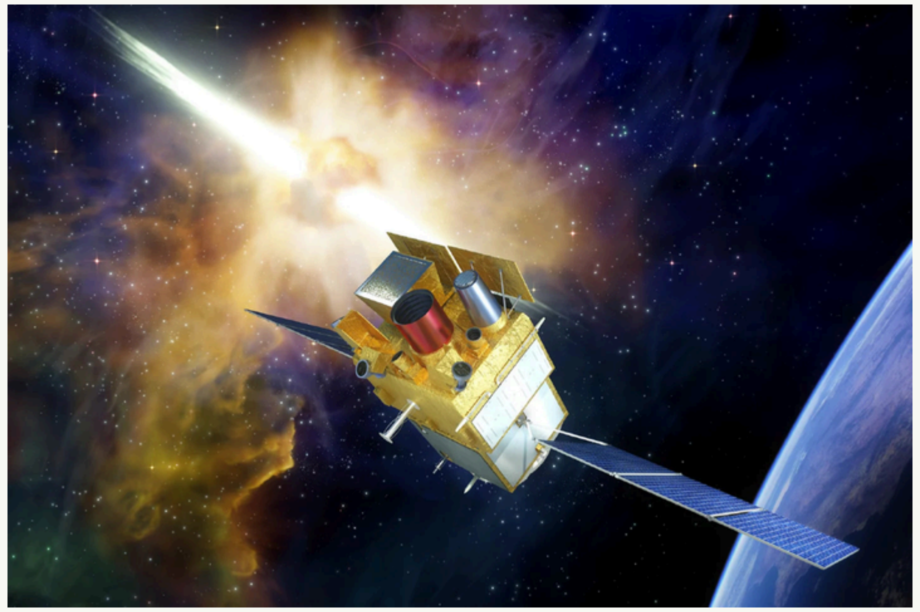
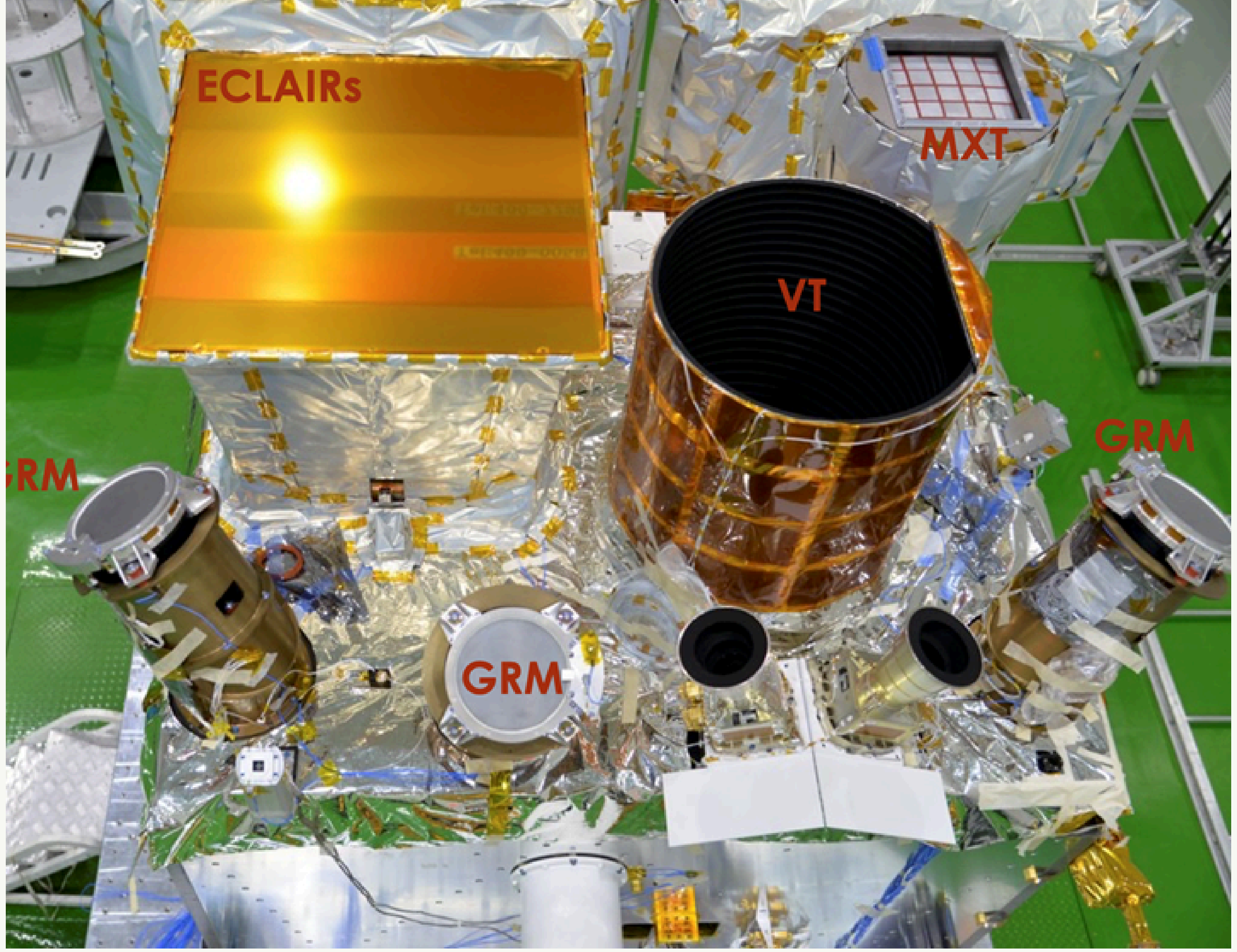
Deep Learning for gamma-ray burst images classification



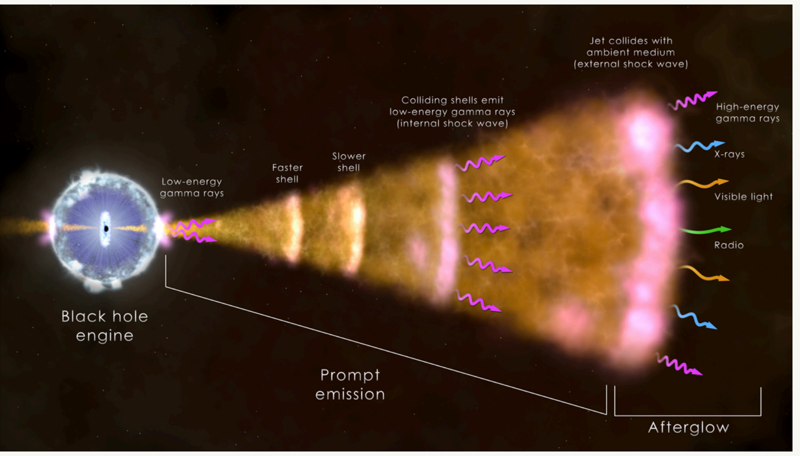
B. Hubert, CEA (Irfu/DAp)
S. Schanne, CEA (Irfu/DAp)

2025/11/27, Caen, Irfu/In2p3 Workshop on IA

SVOM Mission / ECLAIRs

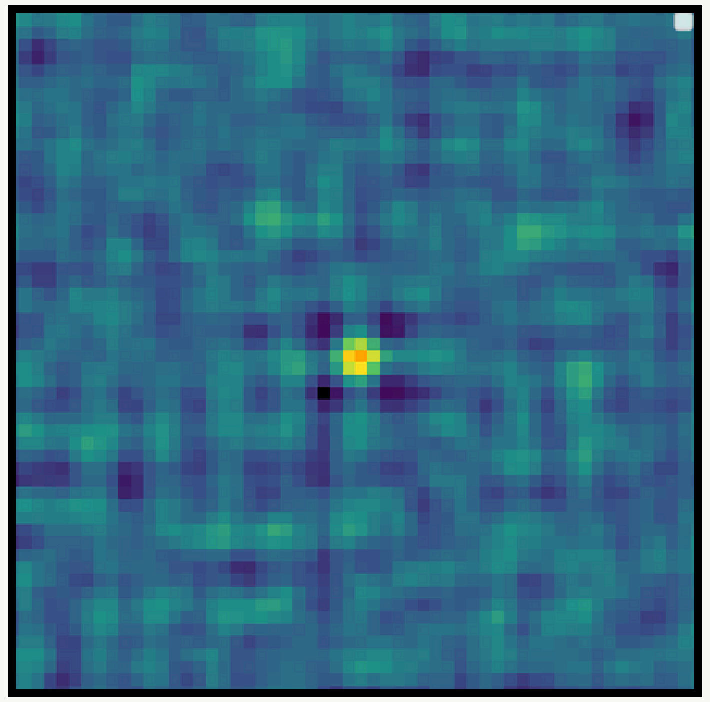


- Satellite launched in June 2024, French-Chinese collaboration between CNSA+CNES (+CEA+CNRS)
- **Dedicated to Gamma-Ray Burst (GRB) studies** (formation of black holes in distant Universe)
- **ECLAIRs instrument + Onboard Trigger (CEA):** detection of new transient gamma-ray source, localization in the field of view (FoV 2 sr), repointing spacecraft in 2 min for follow-up (Visible & X-rays)
- **Alert sequence transmitted to ground in real-time** for each detected GRB (over dedicated VHF net.).
- **Subimage received at the end of an alert sequence**

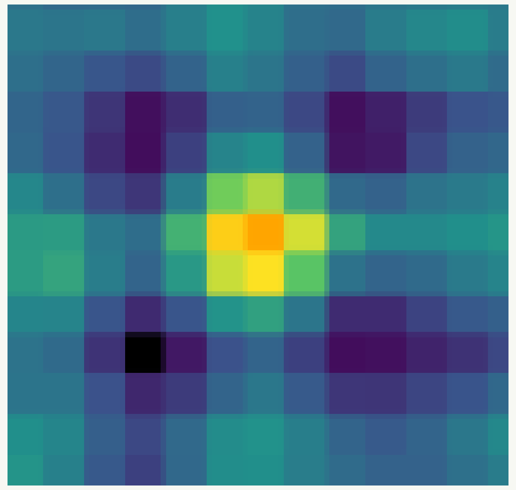


Example

- *Subimage from a true source*
- *56x56 subimage from a 200x200 pixels FoV*
- *each pixel has a SNR value*



Centered point-like source



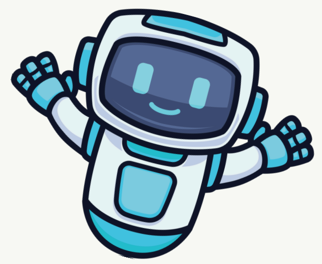
Zoom:
1 maximum ,
4 minima around

SVOM Mission / ECLAIRs

Onboard Trigger:

- Sky image reconstruction (coded mask deconvolution, very frequent ~ 1 per second)
- Alerts produced when a new source is found in sky image when (basically):
  - highest peak (outside known sources) with SNR (signal to noise ratio) above threshold
  - image quality criteria ok (standard deviation ok, highest peak well above 2nd peak)
- Sending of Alerts to ground + Automatic request of Satellite slew (above higher threshold)

On ground a person, the "Burst Advocate" on shift, has to decide:

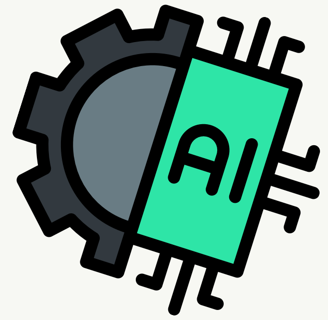


- was it a True GRB ? => organize ground follow-up with visible telescopes, space follow-up with other satellites (Swift and Einstein Probe)
- was it a False Alert ? => cancel the SVOM repointing, invalidate follow-up started

After >1 year of operations: about 75 True GRBs + hundreds known sources + hundreds False Alerts

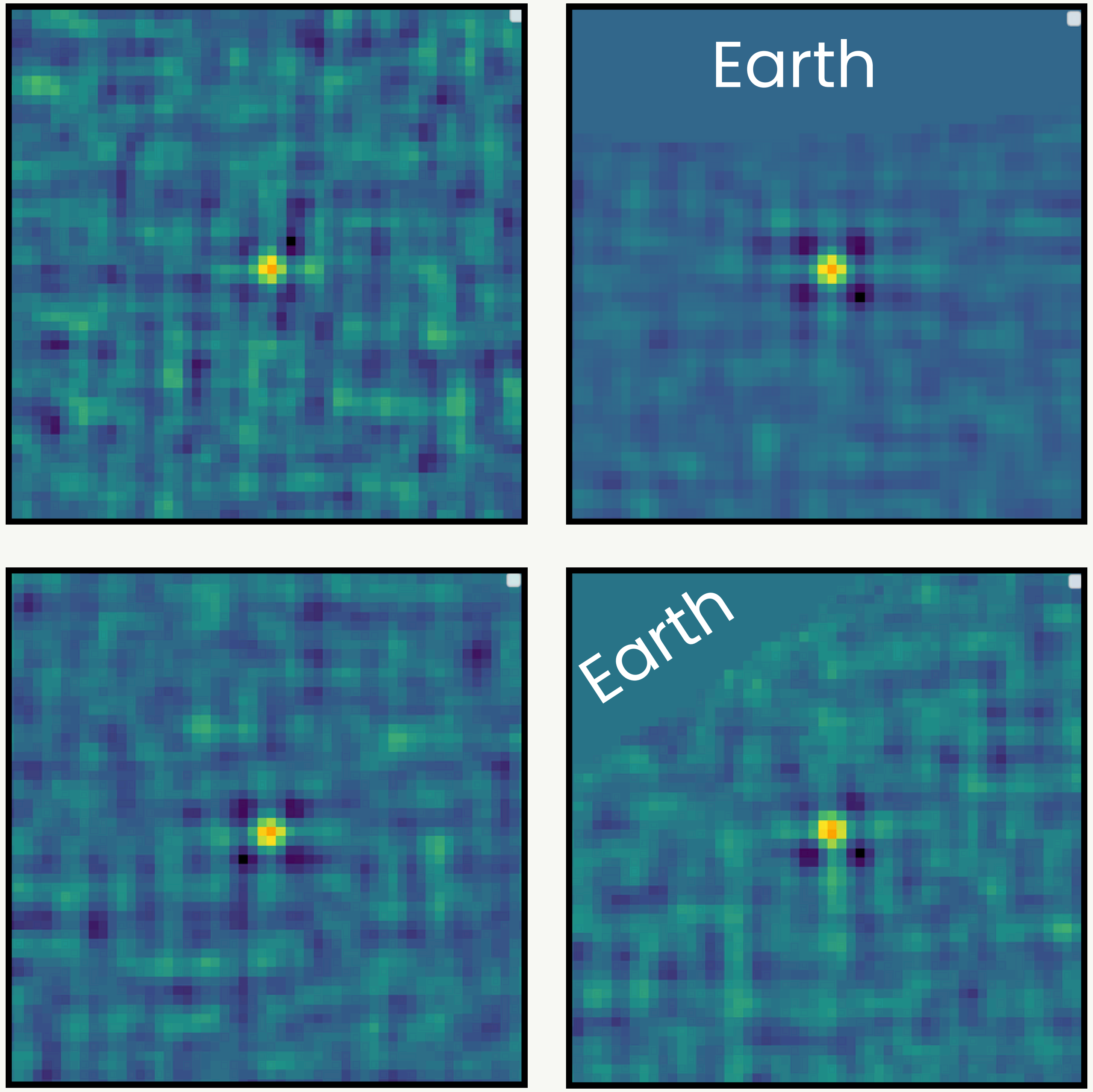
➡ **Automatic classification ?** Machine Learning algorithms applied to transmitted Sub-Images

Solution : Convolutional Neural Network (CNN)
To be used on ground first, and then on board maybe…



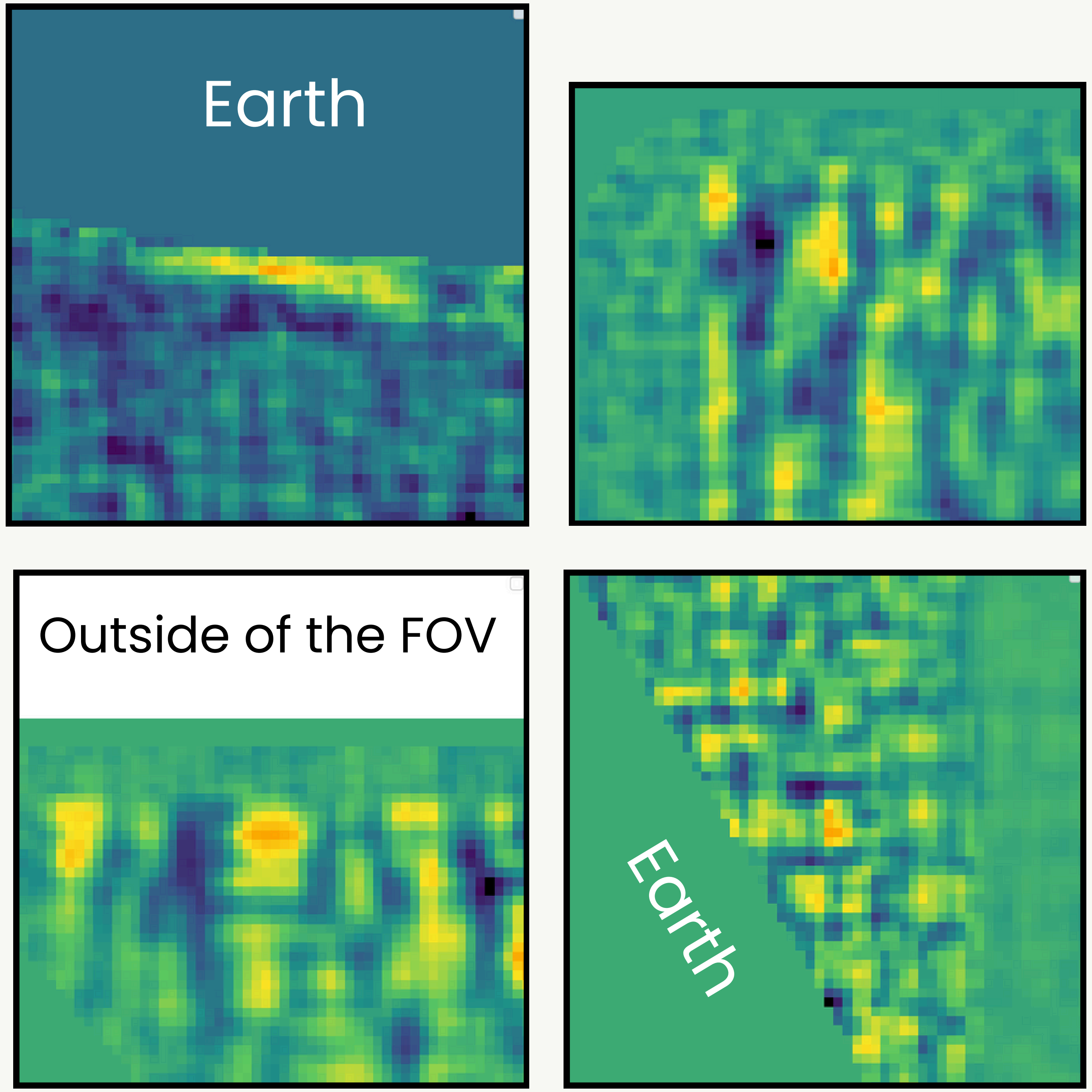
What types of images do we work with ?

Clear “true” cases…



Known gamma-ray sources,
GRBs, unknown gamma-ray sources,…

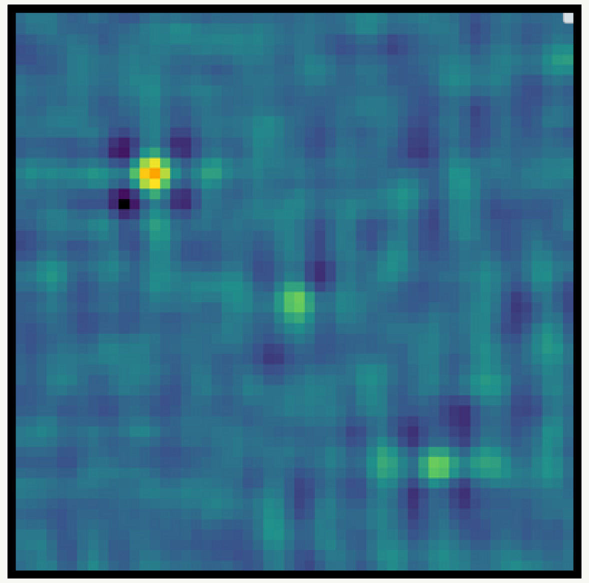
…clear “false” cases…



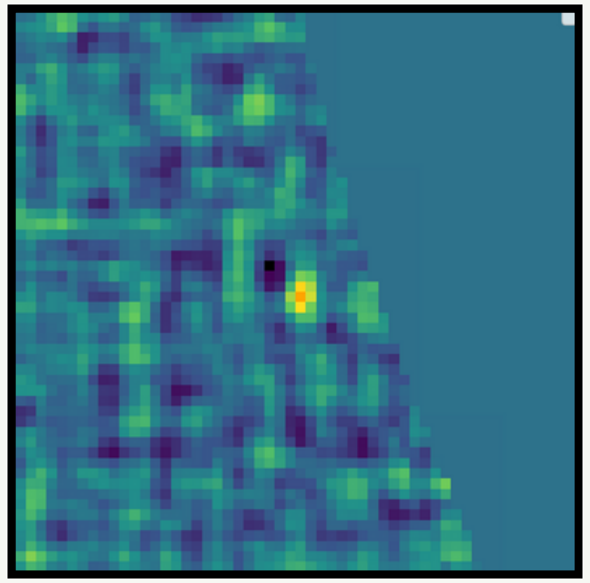
SAA, bad substraction of strong sources, artefacts of bright sources, “column effect”, “Xrdpix inhomogeneity”

What type of images do we work with ?

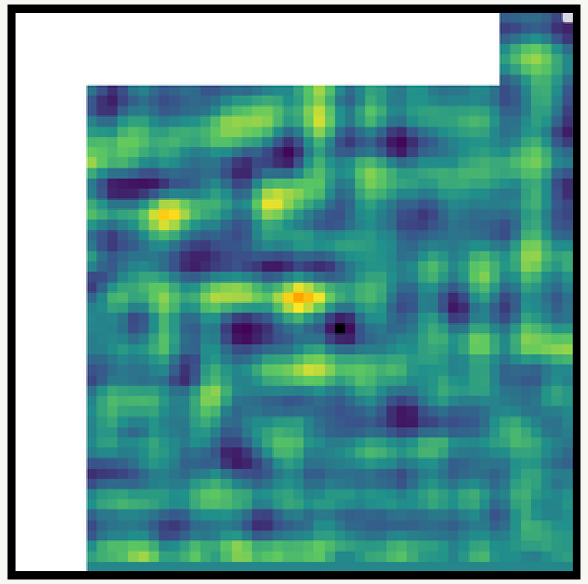
…and sometimes unclear cases:



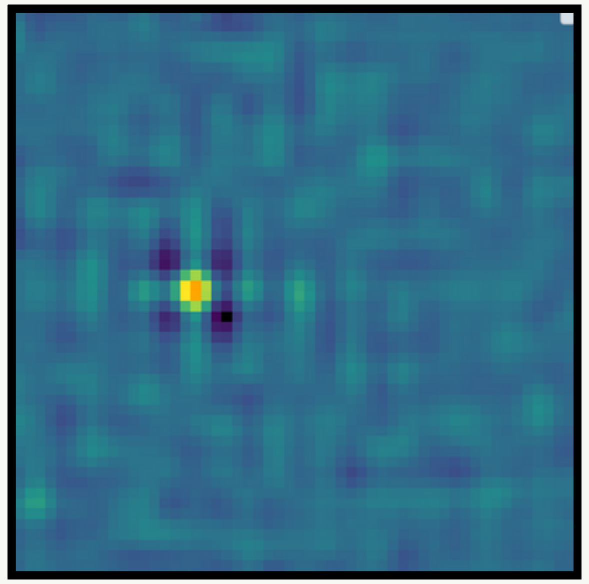
- 2 other sources in the subimage (possibly known-sources)
- The central source (the trigger) looks point-like
- conclusion : unclear (new unknown source or result of coding-noise from the known sources around)



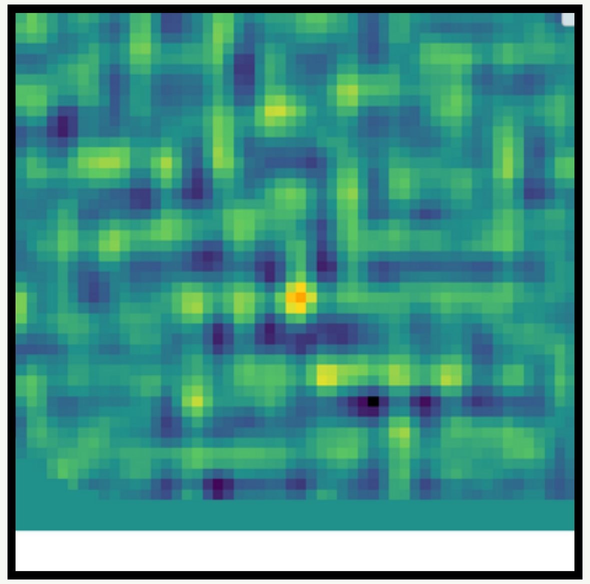
- Source that was close to Earth limb, then disappeared behind Earth
- conclusion : true alert



- Incomplete images
- in this case: false alert



- Looks like a point-like source
- Bad subtraction of a known source
- conclusion : false alert



- Looks like a point-like source
- Background looks perturbed
- conclusion : unclear

Major difficulties for a CNN

Limited number of data → approximately 600 usable images (after putting aside empty images and incorrecly centered images due to software issues…)
Among those 600 images :

- ≈250 false alerts (labelled 0)
- ≈350 true alerts (labelled 1)

A pretty equal distribution → 42/58

Problems to train a model with such a small amount of data :

- Overfitting → the model will be excellent on train set but bad on test set
- Not enough diversity → bad adaptation to atypic cases

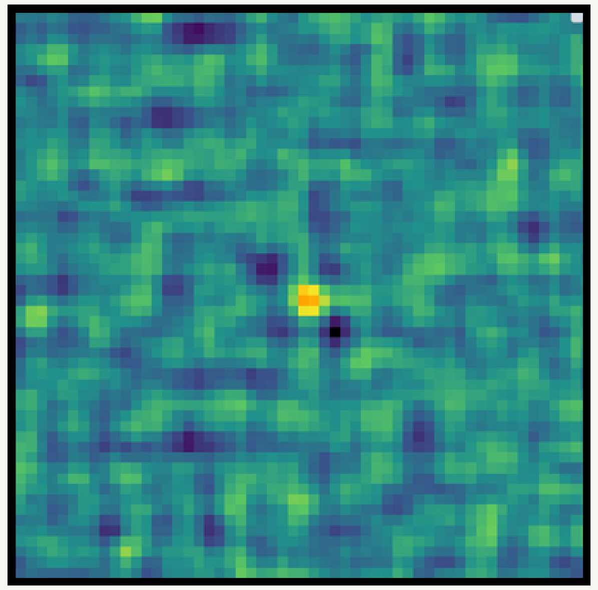
Multiple solutions :

- Transfer learning → use a pre-trained model and freeze all layers except few lasts ones
- Data augmentation → increase artificially the dataset with flips, rotations, adding noise…
- Regularisation → Dropout, Weight Decay, Early Stopping
- Reduced model → Force the model to learn basic features (not too specific patterns)

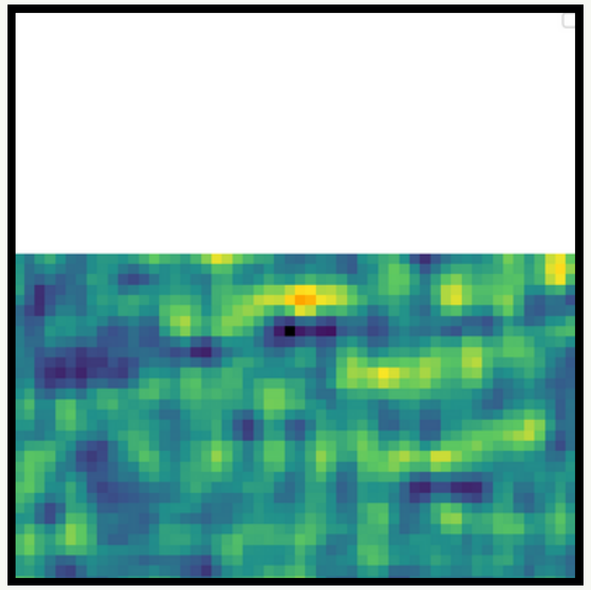
A first simplified study

A **simplified** study, ignoring unclear data to avoid giving the model contradictory labels :

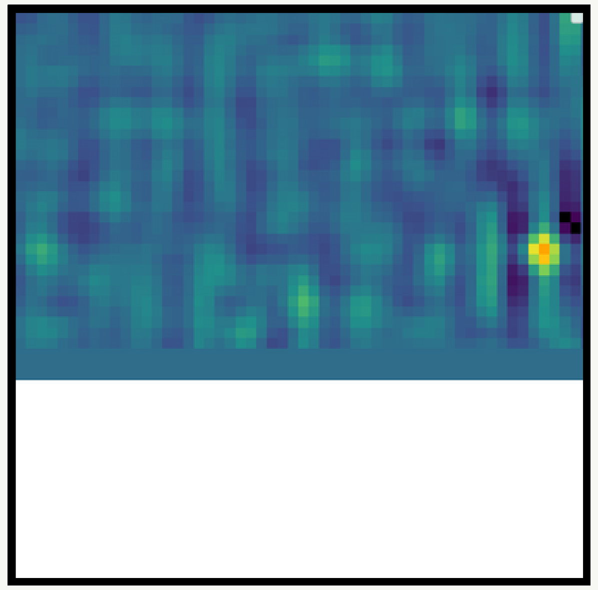
- **Doubt** on a verdict ? → **ignored**
- Use only **full images**



verdict unclear



Partial image + verdict unclear



Partial image + verdict unclear

- **Consequences ?** An even tinier dataset

337 images (20/80 repartition between 0 and 1 → new issue)

The small dataset justifies the use of :

- pre-trained model
- data augmentation

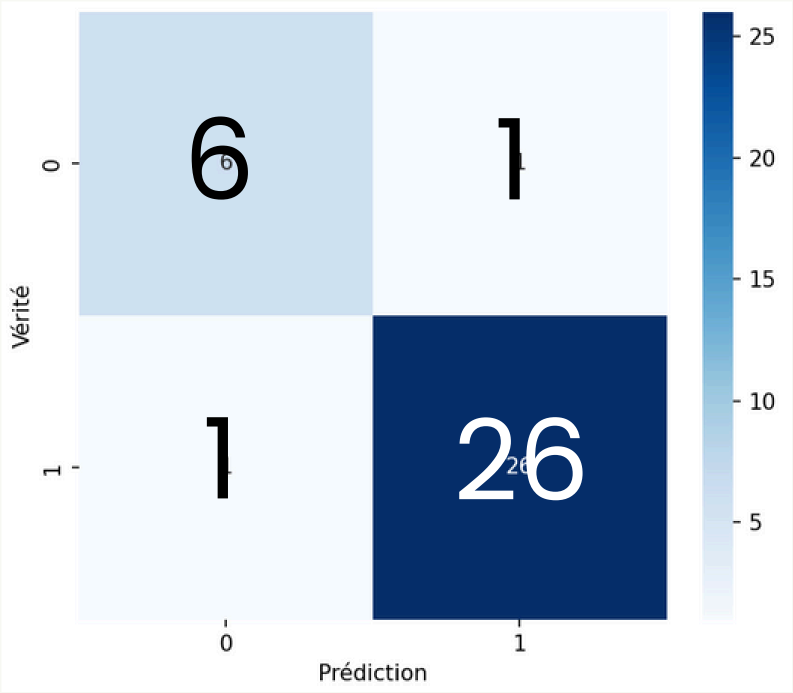
A first simplified study

ResNet50 pre-trained model

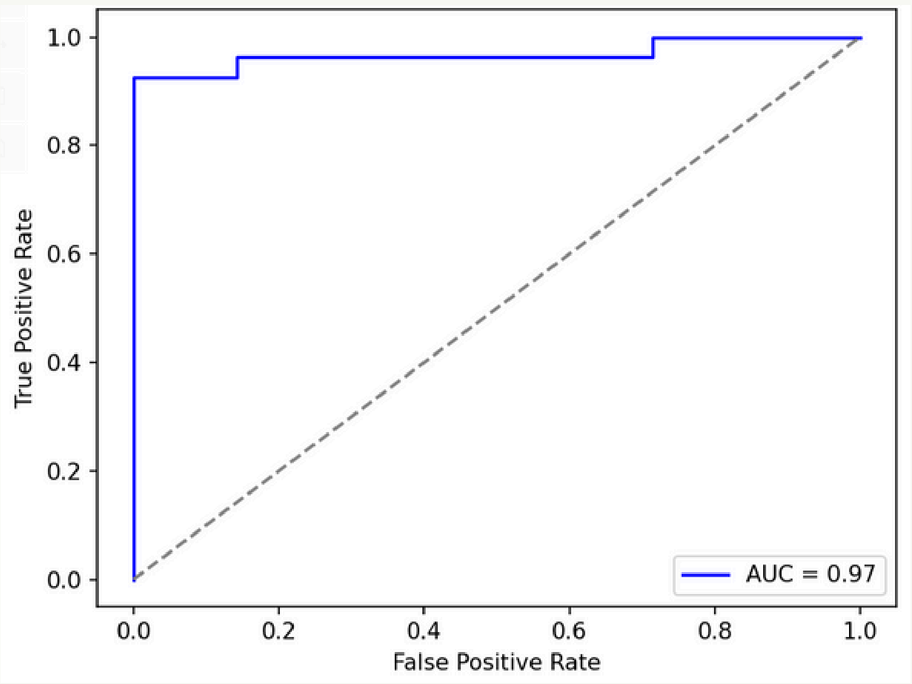
- CNN
- 50 layers (convolutional + BatchNormalisation + ReLu)
- 25 millions of parameters → heavy
- Very efficient (based on residual connections to avoid gradient dilution)
- Works with 224x224 images → reshaping of our data

Can be used as a first approach to have a first conclusive model

Results (training with 20 epochs and a batch size of 32)



Confusion matrix



ROC curve

94% accuracy on test set

Good but ? Must be taken with precaution because :

- Really simplified dataset → of course the model will be efficient
- Small test dataset (34 images)

Data-specific features

Many possible **particularities** that justify a more “advanced” model :

Sometimes image shape is not complete 56x56 pixels
→ How do we complete those images to make it usable for our model ?
→ How do we tell the model which part is the actual data and which one should not be taken into account (missing data, data outside field of view, etc) ?

2 or more sources in an image
→ How do we tell the model that this is a known source ?
→ How do we tell the model which one has triggered an alert ?

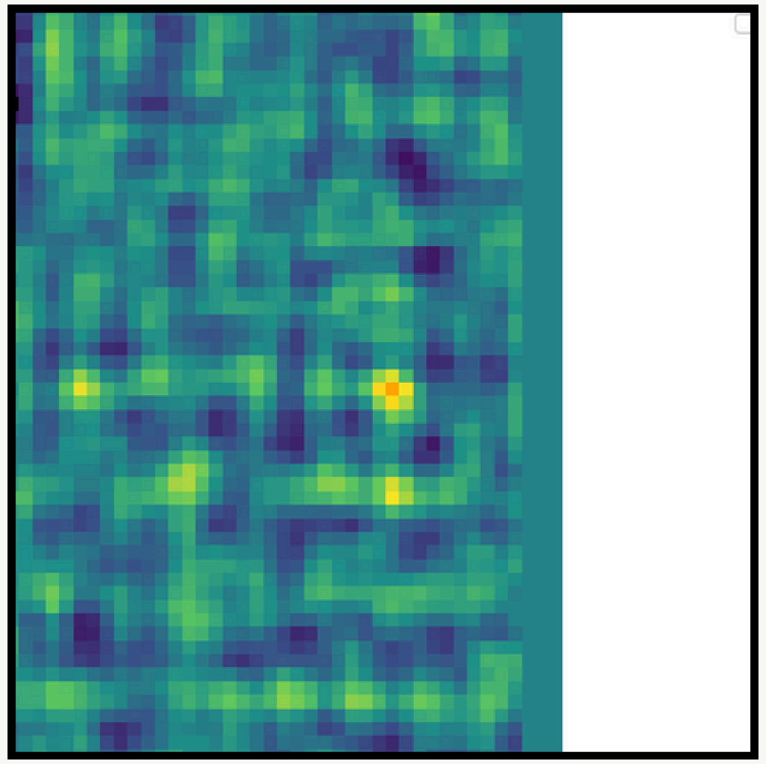
Solution considered :

- A **mask** linked to each subimage as an additional input for the model

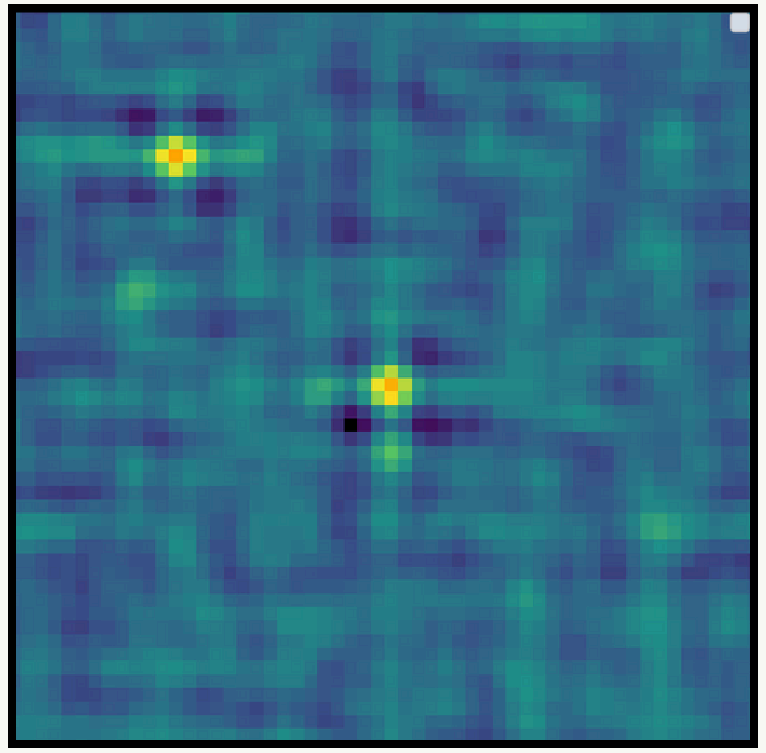
This mask solves all 4 questions mentioned above!

Data-specific features

Mask concept



1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	3	1	1	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	0	0



1	1	1	1	1	1	1	1	1
1	1	2	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	3	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

- "0" for what the model should not take into account (missing part of a sub-image, FOV edges) → red
- "1" for the background (useful data) → green
- "2" for known sources → orange
- "3" for the pixel that triggered → blue

A handmade model

Basic **handmade model**

With the mask, the model has more information to process, but it allows to keep a larger data-set → less overfitting, more diversity

=> 563 images (230 false alerts, 333 true alerts, 41/59 repartition)

Handmade model from scratch :

- Control of its size (less parameters)
- Control/Adaptation of each layer
- Increase the risk of overfitting (if not well countered)

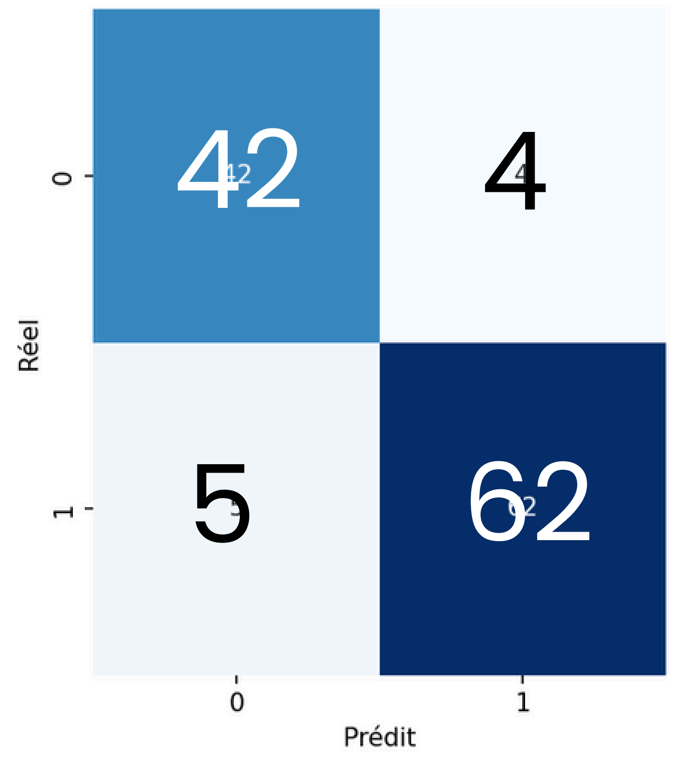
Its **architecture** :

- 1 "concatenate" layer (image + mask)
- 2 "convolutional" layers (MaxPooling, ReLu)
- 1 "dense" layer
- 1 "dropout" layer → remove a fraction of weights, help building a global structure recognition
- Total parameters : 1 624 993 (16 times smaller than ResNet50)

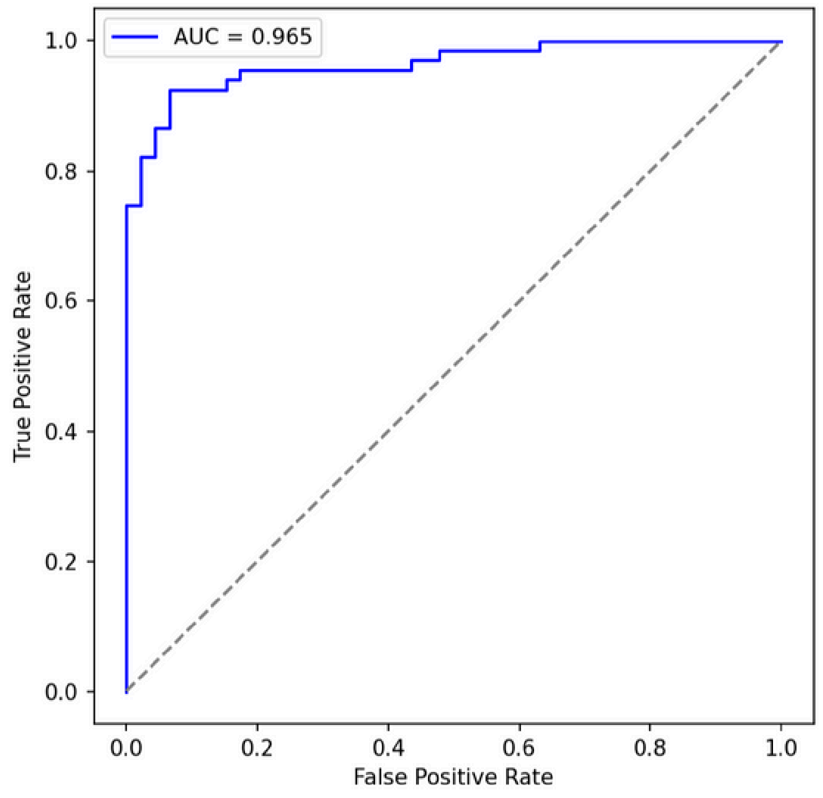
A handmade model

Results (training with 40 epochs and a batch size of 16)

With a 113 images test set :



Confusion matrix



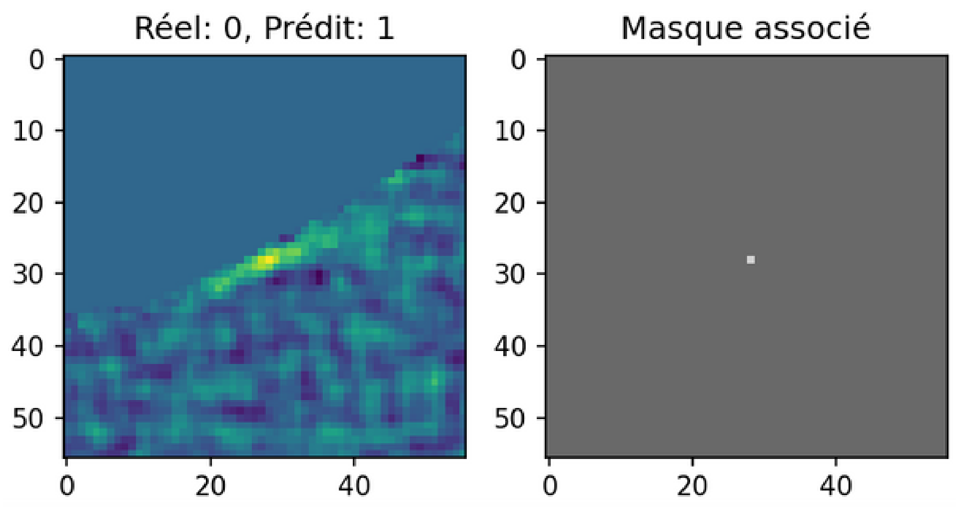
ROC curve

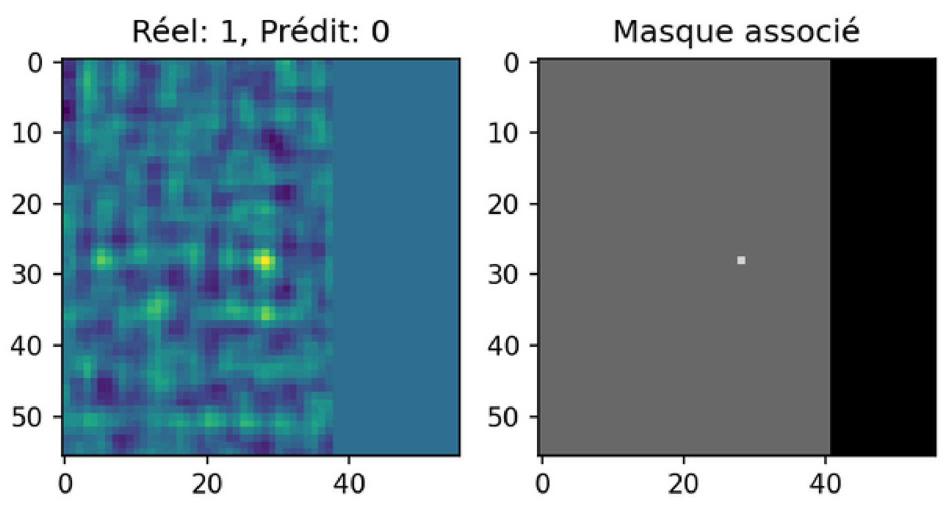
Conclusive **92% test accuracy**

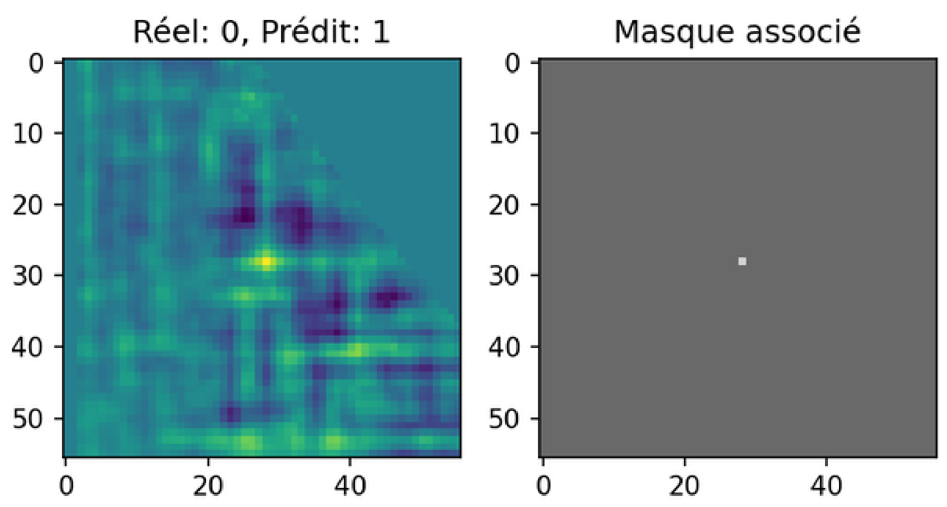
Room for progression :

- Adapt the layers to improve the accuracy
- Integrate the “catalogue” of known sources
- Optimize the number of epochs and the batch size
- Reduce the number of parameters

Exemples of wrong prediction :







What's next ?

Improve performance of the model :

- Data augmentation before training
- Optimize number of parameters
- Optimize model training (epochs, batch size, regularisation…)

Add **catalogue sources** into the mask :

- Look how the model reacts when a known-source is added by providing information in the mask

SVOM flight model

Integrate the model into the **ground segment** (French Science Center hosted at CC-IN2P3)

- Receive data from the VHF/Xband, put it as an input for the model, and get a prediction (false/true alert), as guidance for the "Burst Advocates" before taking follow-up decision

Integrate the model (if possible) into the **on-board software** :

- to avoid performing "False Slews" (repointing satellite on false alerts)

We are open to discussion to try other methods / have feedback on what we did ! :)