

# AVANTIX

## Avancement Simulateur IQ

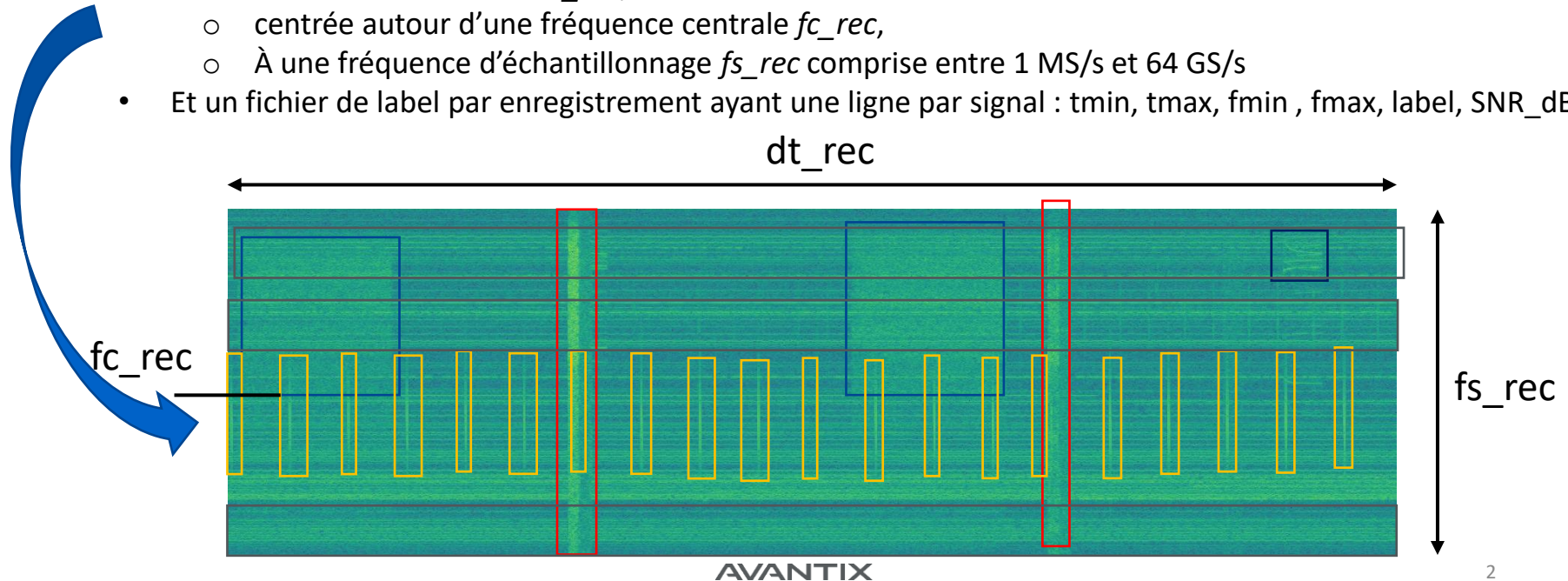
06/12/2023

© Avantix – for internal use

# Besoin

Avoir un programme qui génère en sortie :

- un enregistrement IQ
  - d'une certaine durée  $dt\_rec$ ,
  - centrée autour d'une fréquence centrale  $fc\_rec$ ,
  - À une fréquence d'échantillonnage  $fs\_rec$  comprise entre 1 MS/s et 64 GS/s
- Et un fichier de label par enregistrement ayant une ligne par signal :  $tmin$ ,  $tmax$ ,  $fmin$ ,  $fmax$ ,  $label$ ,  $SNR\_dB$



# Simulateur

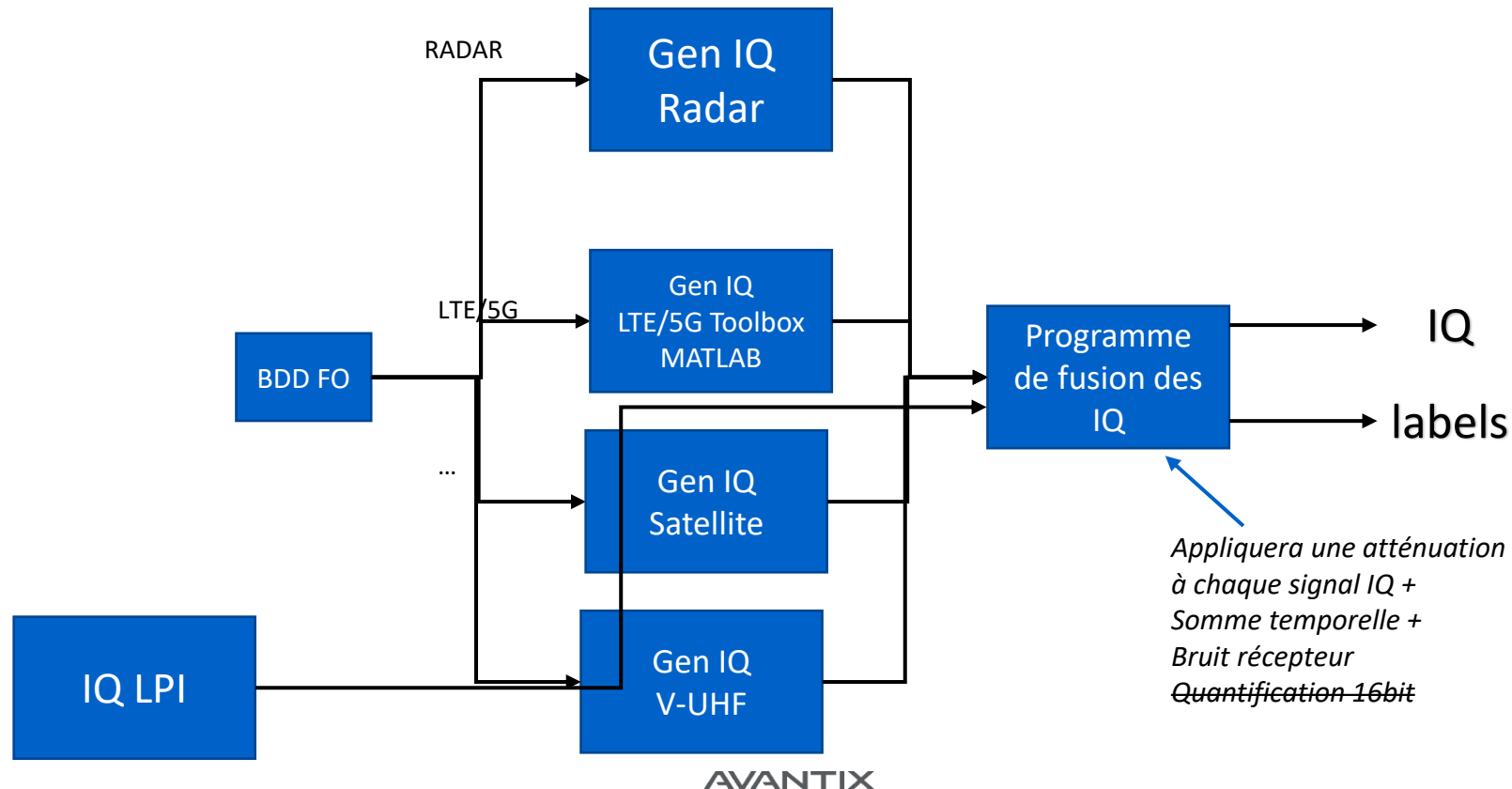
Ce que le programme fera :

- Capable de générer un signal IQ des FO suivantes :
  - **RADAR**
  - **LTE/5G**
  - V-UHF
  - LoRA bande ISM
  - Satellite :
    - THURAYA
    - INMARSAT
    - IRIDUM
    - OneWeb
    - Starlink

Ce que le programme ne fera pas :

- Gérer les cas spécifiques à chaque protocole de communication
- Préfiltrer des FO pour des raisons d'occupation spectrale irréaliste
- Pondérer de manière réaliste les puissances de réception

# Spécifications fonctionnelles



# Spécifications fonctionnelles

```
import pickle
from iq_sigint_simulator import iq_sigint_simulator

sig_types=['cellular', 'radar']
iq_rec, labels, detailed_labels = iq_sigint_simulator(fc_rec, fs_rec, dt_rec,
sigma_noise, sig_types= sig_types)

pickle.dump((record, labels), open('record.pkl', 'wb'))
```

- `fc_rec` : Fréquence centrale d'écoute
- `fs_rec` : Fréquence d'échantillonnage de la bande d'écoute
- `dt_rec` : Durée en secondes de l'écoute
- `sig_types` : Familles des formes d'ondes à simuler
- `sigma_noise` : écart-type de l'amplitude crête du bruit en Volt

Spécification en cours de rédaction :

[\\srv-filer01.global.ad\AFFAIRES TELECOM\1-10-R&D-2200\TECHNIQUE\DOCUMENTS\2\\_Spécification\DP078293STB004\\_00-Specification Simulateur R&D ELINT 2023.docx](\\srv-filer01.global.ad\AFFAIRES TELECOM\1-10-R&D-2200\TECHNIQUE\DOCUMENTS\2_Spécification\DP078293STB004_00-Specification Simulateur R&D ELINT 2023.docx)

# Spécifications fonctionnelles

L'utilisateur renseigne les familles de formes d'ondes (cellulaire ou RADAR) qu'il souhaite générer sur une fenêtre temps-fréquence.

La fonction *iq\_sigint\_generator* va ensuite piocher aléatoirement parmi les FO éligibles et générer la fusion de tous les IQ.

Contraintes d'éligibilités :

- RADAR: Génère entre 200k et 1M d'impulsions par seconde pour une bande d'écoute de 500 MHz
- Cellulaire : Un seuil d'occupation spectrale est renseigné par l'utilisateur en ratio

# SNR

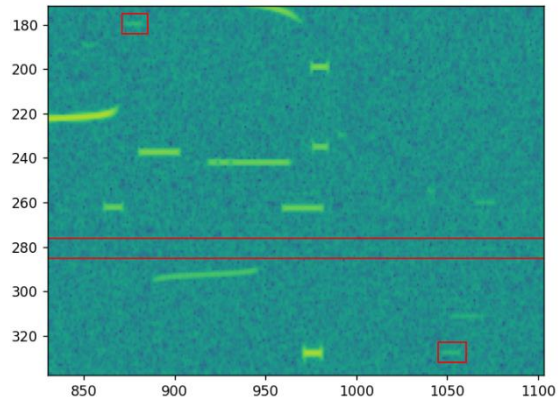
Pour chaque signal, un SNR est choisi aléatoirement entre 3dB et 40dB en apprentissage. On peut envisager sur la base de test entre -10dB et 40dB.

A partir de la puissance de bruit et du SNR, l'amplitude crête du signal est calculé.

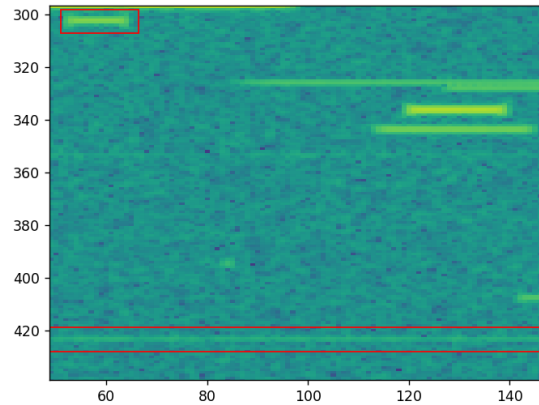
$$\begin{aligned} SNR_{dB} &= DSP_{x_{dB}} - DSP_{N_{dB}} \\ \Rightarrow DSP_{x_{dB}} &= SNR_{dB} + DSP_{N_{dB}} \\ \Rightarrow DSP_x &= 10^{\frac{SNR_{dB}}{10}} * \frac{\sigma_N^2}{f_s} \\ \Rightarrow \frac{A^2}{2} * k_{type} &= 10^{\frac{SNR_{dB}}{10}} * \frac{\sigma_N^2}{f_s} \\ \Rightarrow A &= \sqrt{\frac{2}{f_s * k_{type}} 10^{\frac{SNR_{dB}}{10}} * \sigma_N} \end{aligned}$$

$$\text{Avec } k_{type} = \begin{cases} \sqrt{\frac{pw}{\Delta f}} & \text{si chirp radar} \\ pw_{moment} & \text{si pmop radar} \\ pw & \text{si radar no mod ou cw} \\ \frac{1}{\Delta f} & \text{sinon} \end{cases}$$

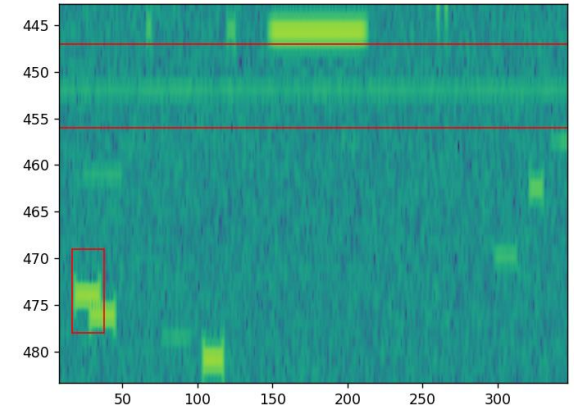
# SNR : Exemple



```
no_mod 7.989360320811331  
no_mod 7.538338205415122  
cw 7.535235953020623  
no_mod 7.862971690719021  
no_mod 7.881505883690628
```



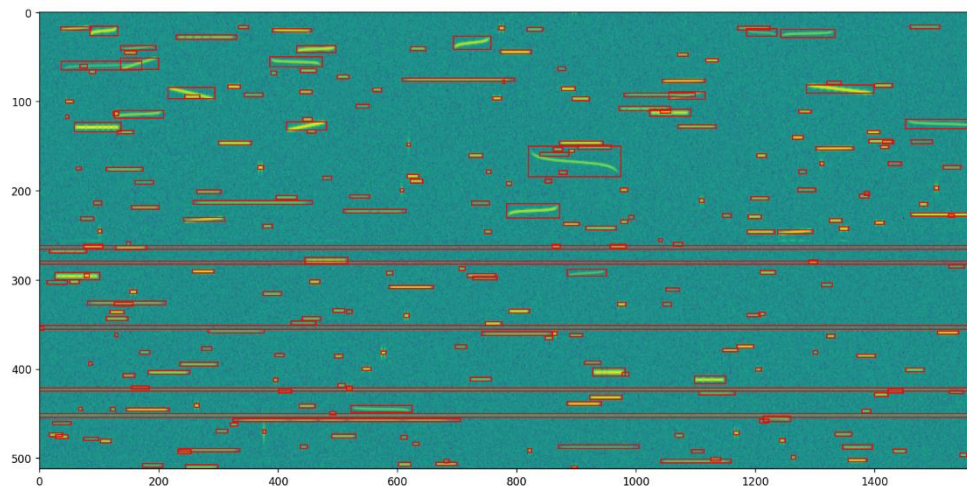
```
no_mod 26.239489169210728  
no_mod 26.296589289997957  
cw 26.135956525095562  
no_mod 26.29549689178736
```



```
cw 29.71672951365484  
no_mod 29.746267947122526  
no_mod 29.92441421158403  
p_mod 29.563106294909332  
no_mod 29.692284037158846
```

# Label → Bounding Box

Implémentation d'une fonction permettant de générer des bounding box quelques soit les paramètres du spectrogramme.



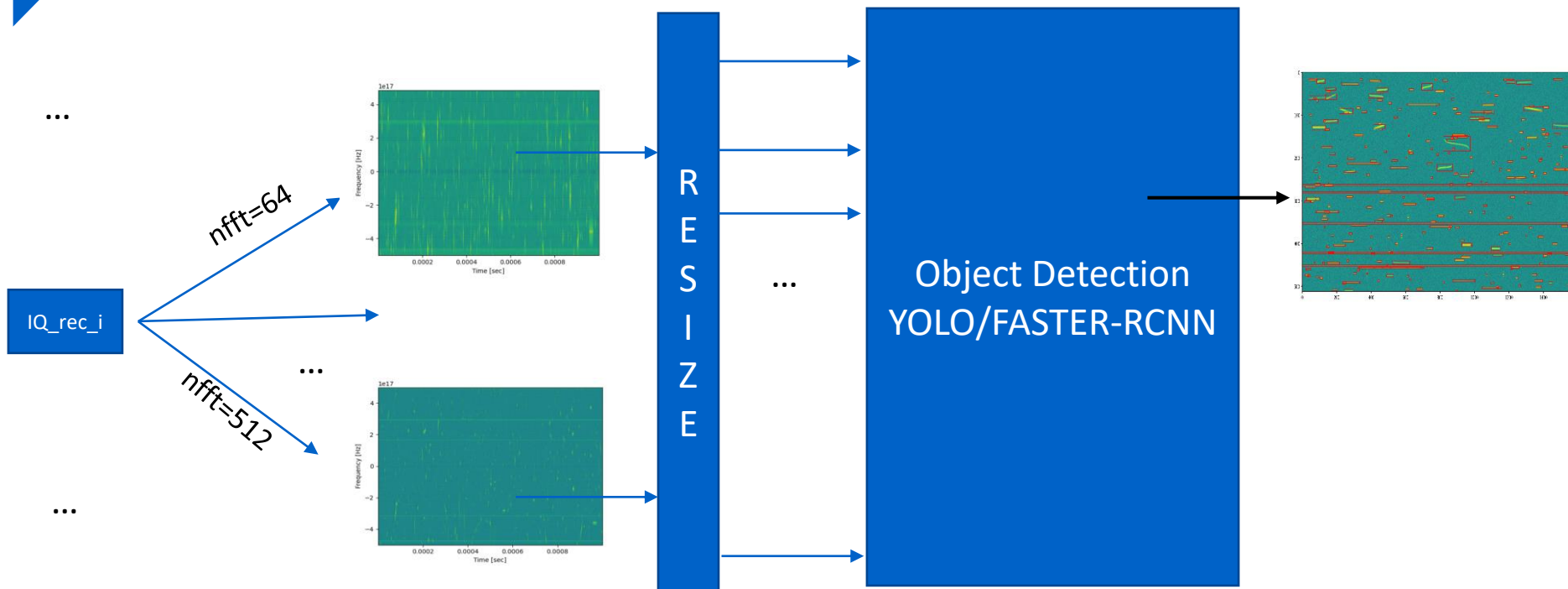
# Génération BT 0.1.0 : en cours

## Paramètres :

```
fc_rec = 1GHz  
fs_rec = [1, 5, 25, 100, 500, 1e3, 2e3, 4e3] MS/s  
dt_rec = 1ms  
dt_total = 1min → 60k enregistrements  
sigma_noise = 1mV  
sig_types = 'radar'  
snr_db = 3 à 40dB
```

Durée simulation = 60h

# Chaine de traitement : Détection (Etat de l'art en cours)



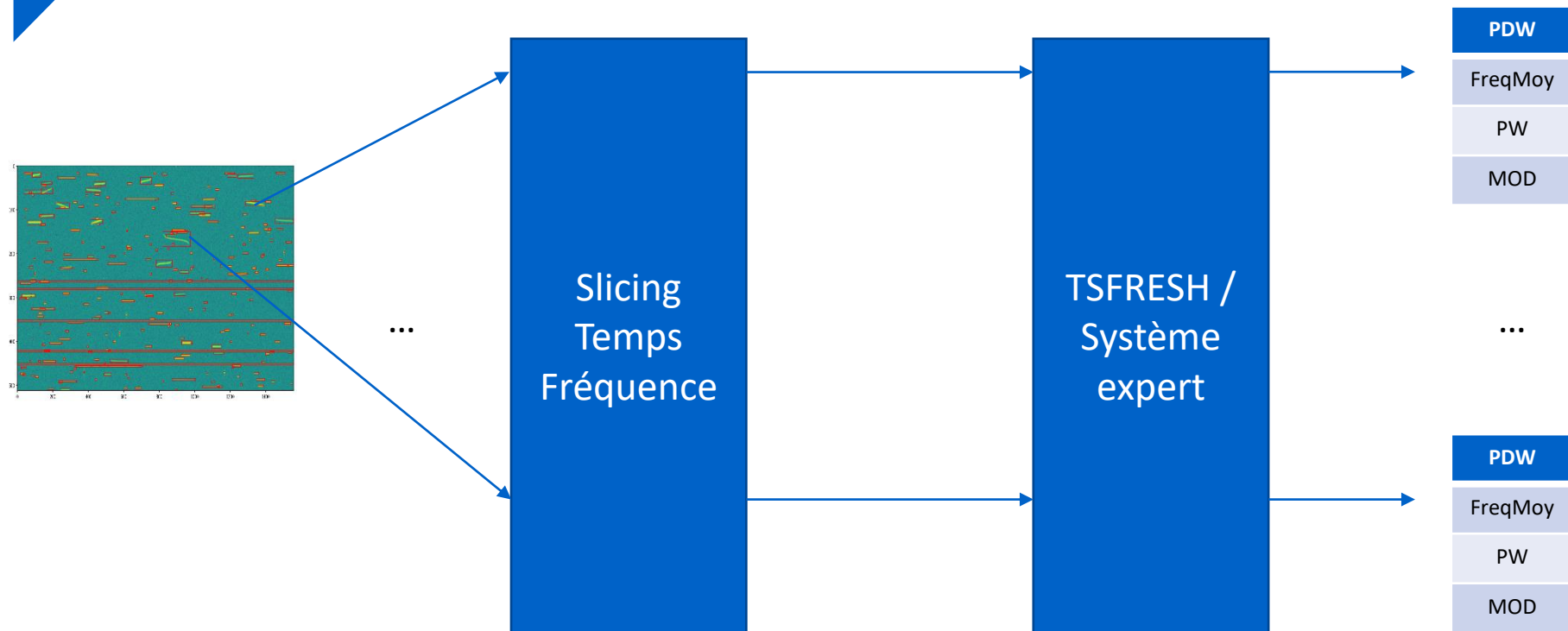
# Framework: Detectron2

<https://github.com/facebookresearch/detectron2>

- Framework le plus populaire en CV
- Basé sur PyTorch
- Permet de faire de l'object detection
- Permet de sélectionner des modèles pré-entraînés



# Chaine de traitement : Caractérisation



# AVANTIX

MERCI