Avancement Simulateur IQ

BT1.0.2

Paramètres :

```
nperseg = [128, 256, 512]
noverlap = nperseg / 2
nfft = 512

fc_rec = 2.5 GHz
fs_rec = [4, 20, 100, 500, 1e3, 4e3] MS/s
dt_rec = 513 * nperseg / (2*fs_rec)
sigma_noise = 1mV
sig_types = [[‘radar’], [‘radar’, ‘cellular’]]
snr_db_opt_radar : friend / no friend
snr_db_opt_com = 35 dB
snr_db_eff = -2 dB

>100k enregistrements
```

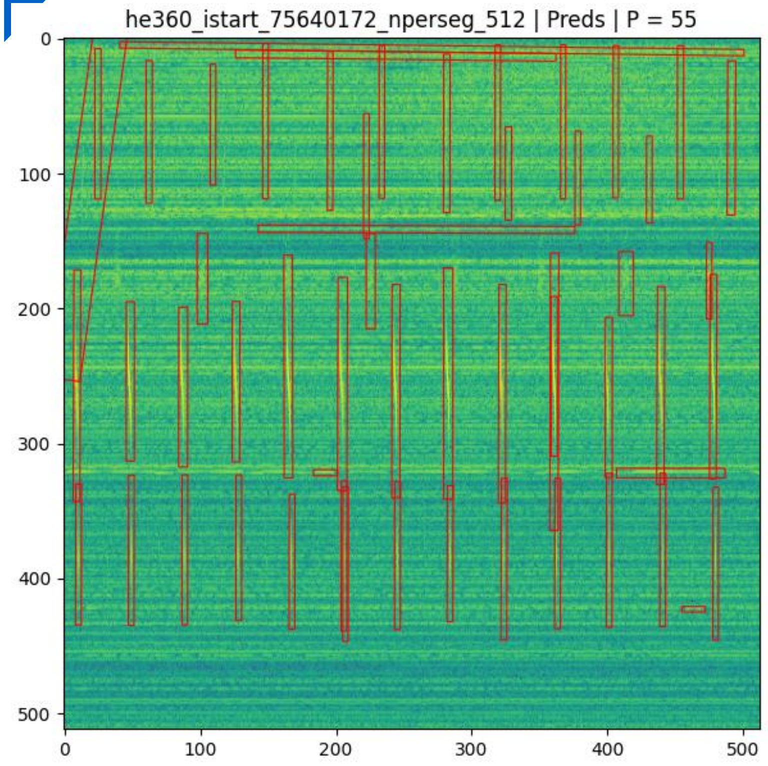
DI (ns)	*Sans Modulation*	*Modulations de Fréquence (MHz/µs)*		*Modulations de Phase Phase / Nombre de moments*	
10-100	Oui	-	-	-	
100-1_000	Oui	-	-	-	
1_000-10_000	Oui	20	100	180°	7, 8
10_000-100_000	Oui	2	10	180° 60° 45°	7, 8, 13, 25, 33, 39 25, 30 8, 35, 70
100_000-1_000_000	Oui	10^{-3}	1	180° 60° 45°	7, 8, 13, 25, 33, 39 25, 30 8, 35, 70
1_000_000-10_000_000	Non	0,001	0,1	-	
10_000_000-100_000_000	Non	0,0001	0,01	-	
100_000_000-inf	Non	0,00001	0,001	-	
CW	Oui	-	-	-	

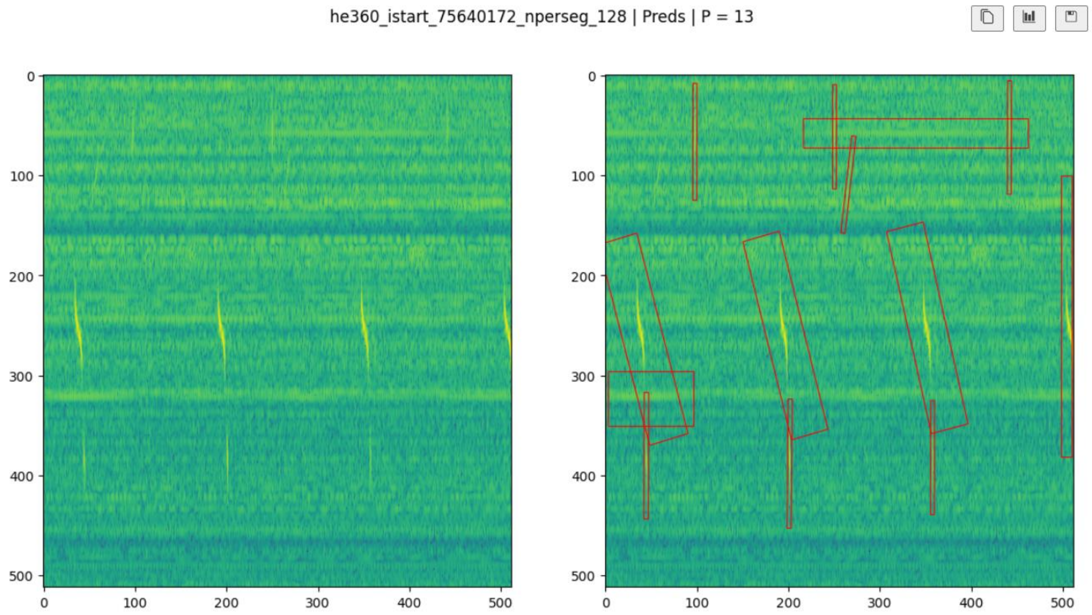
Meta-formes d'ondes

3 classes

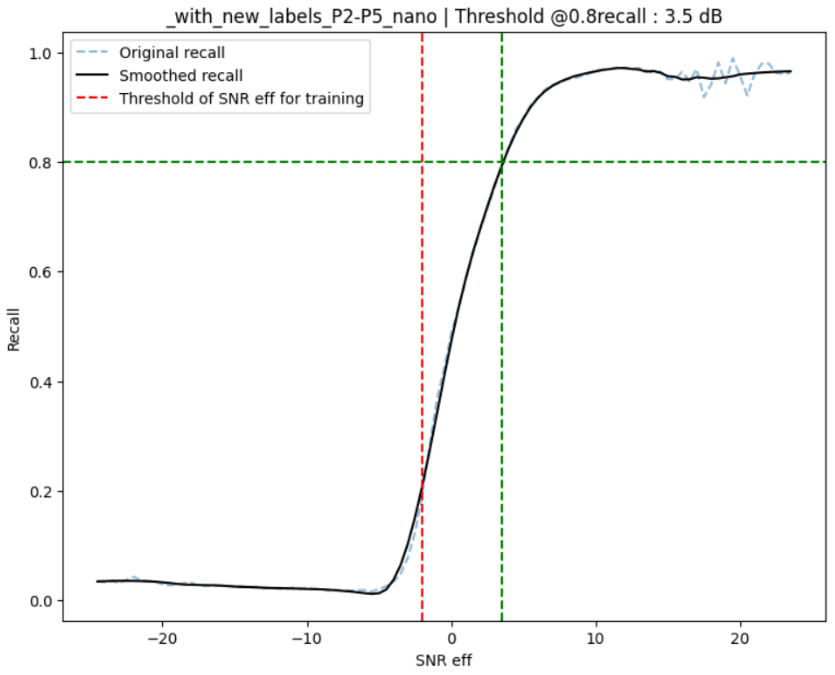
- NO-MOD + CW + P1 + Frank (qq fois doucle chirp) + P4 + Biphasique (Barker + Random)
- FMOD + FMCW + P2 (ressemble au symbole %)
- P3 (double chirp)

Résultats HE360 sur YOLOv8-OBB nano

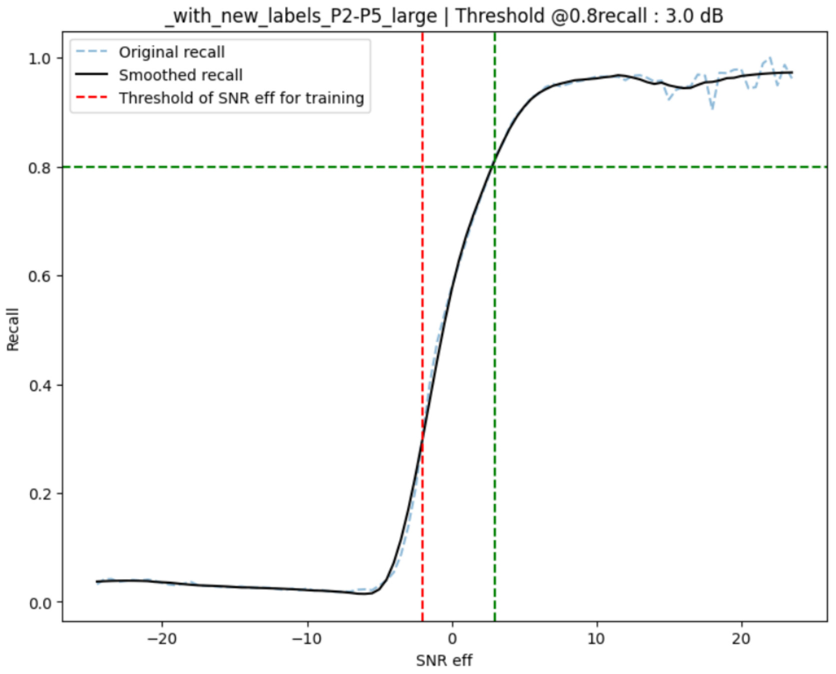




Résultats BT102 : Stats sur train

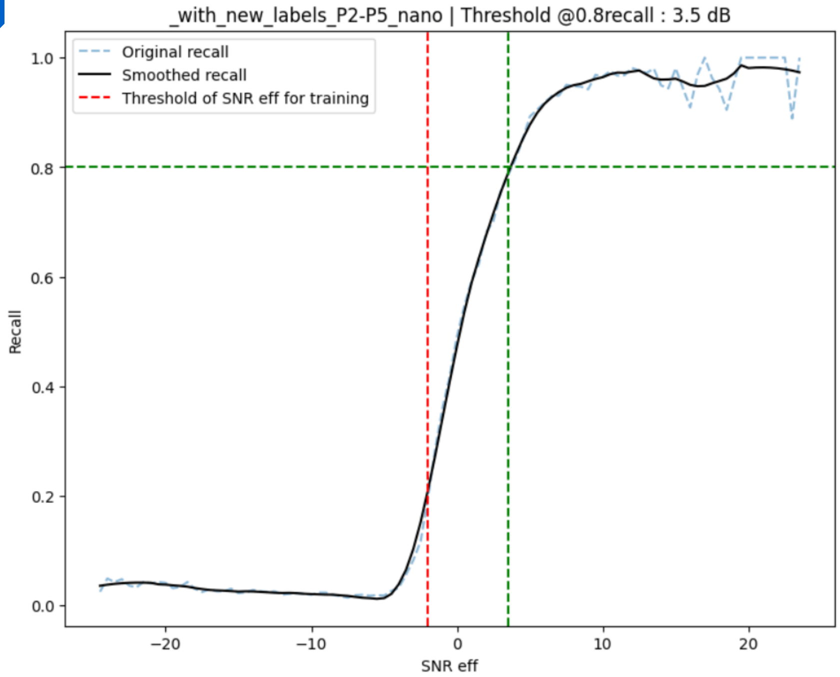


Agg result
recall = 31.04% | precision = 91.75% | multidetection ratio = 2.79%

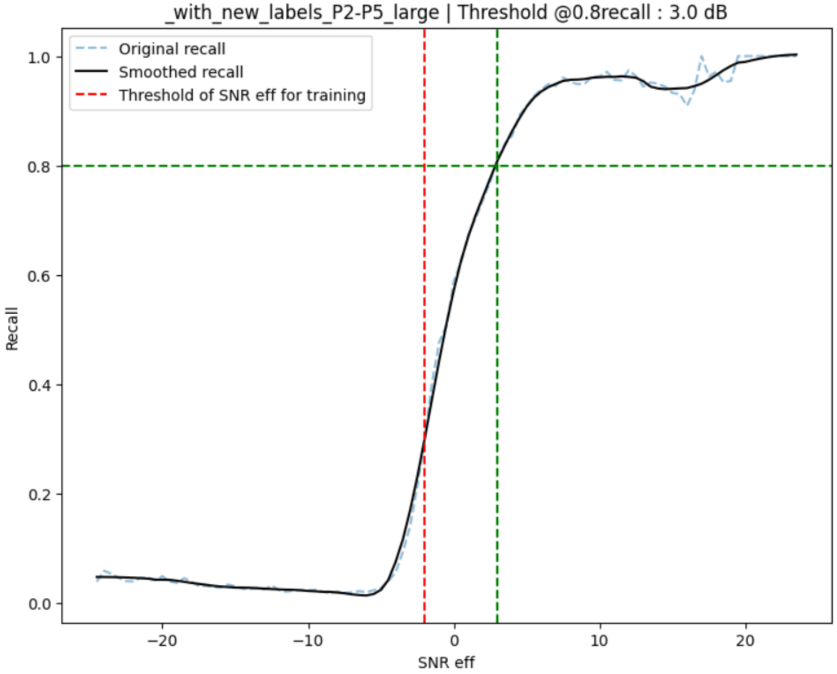


Agg result
recall = 33.86% | precision = 91.54% | multidetection ratio = 3.13%

Résultats BT102 : Stats sur test



Agg result
recall = 30.82% | precision = 91.50% | multidetection ratio = 2.86%



Agg result
recall = 33.59% | precision = 91.43% | multidetection ratio = 3.15%

AVANTIX

Résultats BT102 : Exemple haut SNR mal prédit

TODO

Résultats BT102 : Problème de SNR effectif

TODO

Bug YOLOv8

Apprentissage instable : Bbox Loss qui diverge → A débuguer

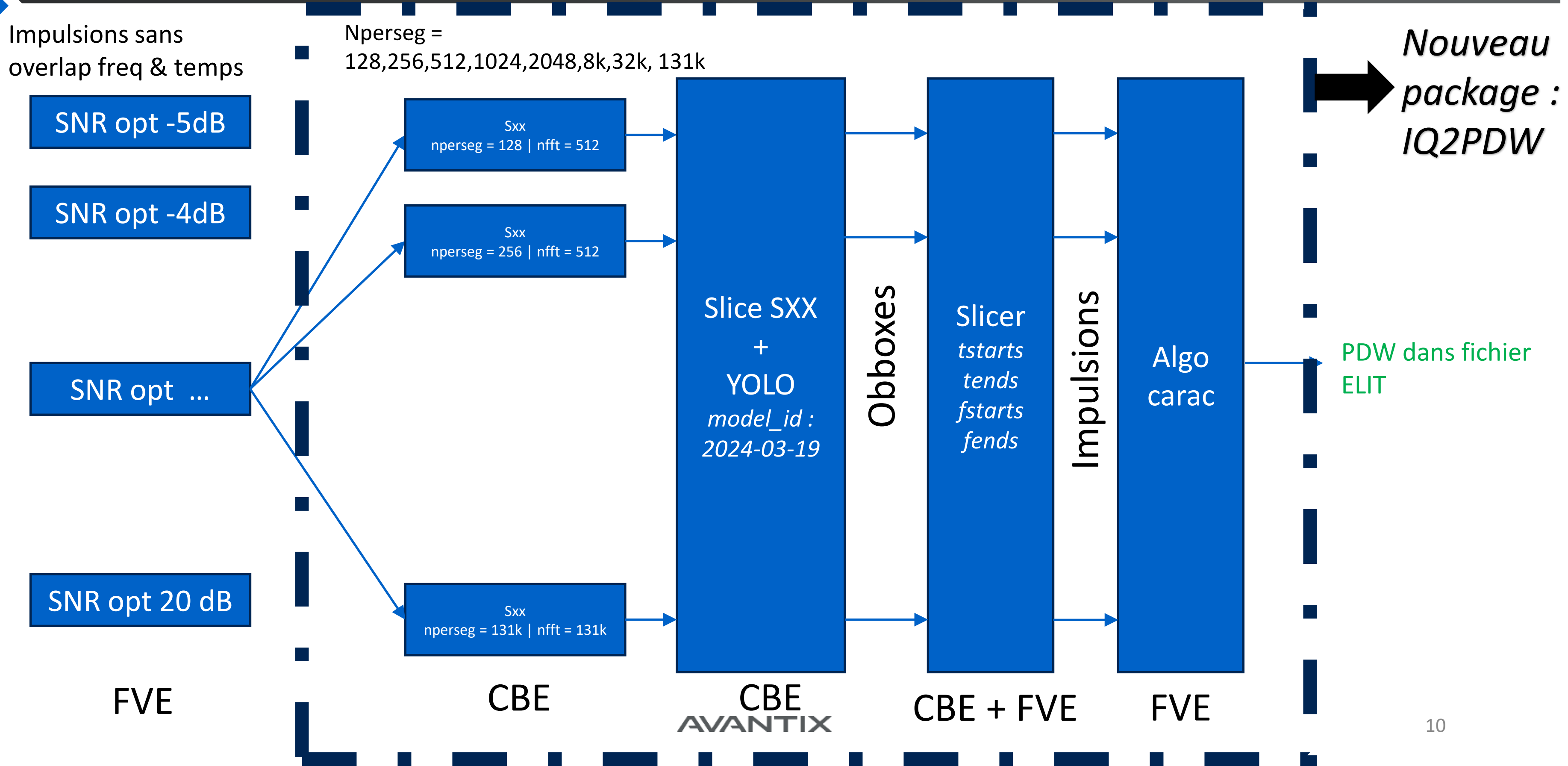
Sous apprentissage → Plus gros modèle nécessaire (après résolution du pb précédent)

CBE > bt102 > runs > obb > _with_new_labels_P2-P5_large2 > results.csv > data

epoch,	train/box_loss,	val/box_loss,	train/cls_loss,	val/cls_loss,	trai
27,	1.364,	nan,	1.1113,	1.2613,	
28,	1.3588,	nan,	1.1069,	1.2643,	
29,	1.348,	nan,	1.1019,	1.301,	
30,	1.3423,	nan,	1.0969,	1.3273,	
31,	1.3268,	nan,	1.0833,	1.3589,	
32,	1.3157,	nan,	1.0742,	1.3726,	
33,	1.3189,	nan,	1.08,	1.3967,	
34,	1.3064,	nan,	1.0676,	1.4224,	
35,	1.3026,	nan,	1.0668,	1.4382,	
36,	1.2927,	nan,	1.0583,	1.4613,	
37,	1.2925,	nan,	1.0597,	1.459,	
38,	1.2816,	nan,	1.0516,	1.4648,	
39,	nan,	nan,	nan,	nan,	
40,	nan,	nan,	nan,	nan,	
41,	nan,	nan,	nan,	nan,	
42,	nan,	nan,	nan,	nan,	
43,	nan,	nan,	nan,	nan,	
44,	nan,	nan,	nan,	nan,	
45,	nan,	nan,	nan,	nan,	
46,	nan,	nan,	nan,	nan,	
47,	nan,	nan,	nan,	nan,	
48,	nan,	nan,	nan,	nan,	
49,	nan,	nan,	nan,	nan,	
50,	nan,	nan,	nan,	nan,	

Stand by

Implémentation traitement I/Q bout-en-bout



YOLOv9

https://arxiv.org/pdf/2402.13616.pdf

Résout le problème de la perte d'information dans les réseaux profonds en proposant le programmable gradient information (PGI) et le generalized ELAN (GELAN). Cela a permis d'obtenir un nouvel état de l'art des perfs sur MSCOCO.
Des branches sont ajoutées durant l'apprentissage uniquement ne pénalisant pas le temps d'inférence.

Table 1. Comparison of State-of-the-Art Real-Time Object Detectors

Model	size (pixels)	AP^{val} 50-95	AP^{val} 50	AP^{val} 75	params (M)	FLOPs (B)
YOLOv9-S	640	46.8	63.4	50.7	7.2	26.7
YOLOv9-M	640	51.4	68.1	56.1	20.1	76.8
YOLOv9-C	640	53.0	70.2	57.8	25.5	102.8
YOLOv9-E	640	55.6	72.8	60.6	58.1	192.5

Detection (COCO) Detection (Open Images V7) Segmentation (COCO) Classification (ImageNet) Pose (CO

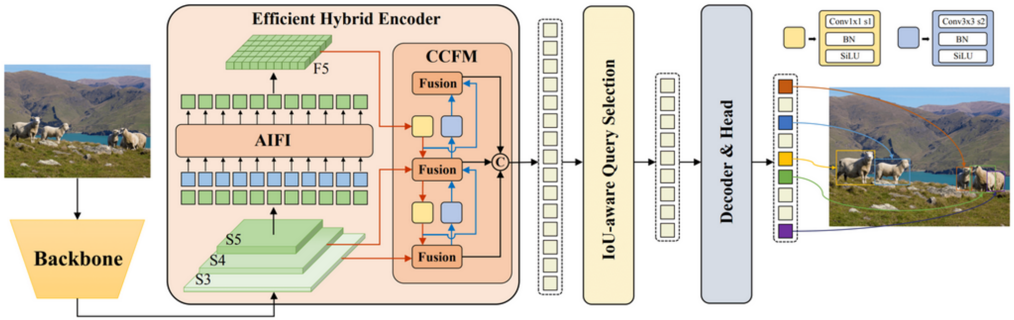
See Detection Docs for usage examples with these models trained on COCO, which include 80 pre-trained classes.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

RT DETR

Overview

Real-Time Detection Transformer (RT-DETR), developed by Baidu, is a cutting-edge end-to-end object detector that provides real-time performance while maintaining high accuracy. It leverages the power of Vision Transformers (ViT) to efficiently process multiscale features by decoupling intra-scale interaction and cross-scale fusion. RT-DETR is highly adaptable, supporting flexible adjustment of inference speed using different decoder layers without retraining. The model excels on accelerated backends like CUDA with TensorRT, outperforming many other real-time object detectors.



Modèle intéressant pour faire du temps-réel **mais n’a pas (pour le moment) d’implémentation OBB**

Est beaucoup moins considéré car il nécessite un gros préapprentissage or ce n’est pas un frein pour notre cas d’usage.

AVANTIX

MERCI