

AVANTIX

Avancement Simulateur IQ

Avancement

Bt200 :

- Réflexion d'une nouvelle base de données qui convient mieux à notre chaine de traitement
- En cours de génération

iq2pdw :

- Découpe des IQ par tranches de 2ms avec un overlap de 50%
- Exclusion des impulsions qui commencent avant ou finissent après une écoute de 2ms car seront vu dans leurs entières dans une différente vue
- Adapter max_detection selon la taille du spectrogramme

YOLO :

- Améliorer modèle 2019-03-19 → en cours via tuning sur P2-P5 large
- Utiliser Module + Angle → non débuté
- Bug NAN durant l'apprentissage
 - Ignorer les $t3 < 0$ → n'a pas fonctionné (plus aucun NaN mais explosion des bbox loss)
 - YOLOv5 ultralytics → n'a pas fonctionné (même fct de coût que v8 donc même pb)
 - Implémentation alternative de YOLOv5 → en cours
- Bug CW rogné → Discussion avec le lead dev en cours

YOLO : Bug durant l'apprentissage

Ignorer les valeurs de t3 qui sont à NaN ne suffit pas :

CBE / D102 / runs / 000 / _after_fix_PCH_tune_float16_F2-F3-nano / results.csv / data									
1					epoch,	train/box_loss,	val/box_loss,	train/cls_loss,	
12					11,	1.7050,	52.722,	1.750,	
13					12,	1.6602,	42.668,	1.31,	
14					13,	1.644,	53.759,	1.2,	
15					14,	1.6264,	78.297,	1.28,	
16					15,	1.6057,	90.672,	1.27,	
17					16,	1.587,	46.209,	1.26,	
18					17,	1.5662,	92.854,	1.25,	
19					18,	1.5589,	79.7,	1.2,	
20					19,	1.5493,	67.194,	1.2,	
21					20,	1.5381,	48.897,	1.24,	
22					21,	1.5221,	90.519,	1.23,	
23					22,	1.5104,	77.836,	1.22,	
24					23,	1.5044,	81.785,	1.22,	
25					24,	1.4906,	48.435,	1.21,	
26					25,	1.4816,	94.05,	1.21,	
27					26,	1.472,	37.504,	1.26,	
28					27,	1.4743,	60.744,	1.26,	
29					28,	1.4639,	81.131,	1.19,	
30									

YOLOv5 : Bug durant l'apprentissage

J'ai testé YOLOv5 d'ultralytics → bug tjrs présent car utilise la même fonction de coût

CBE > bt102 > runs > obb > _yolov5 > results.csv > data

	epoch,	train/box_loss,	val/box_loss,	train/cls_loss,	val/cls_loss,	train/df1_loss,	val/df1_loss,
34	33,	1.5238,	nan,	1.1671,	1.1333,	2.1468,	2.2237,
35	34,	1.5084,	nan,	1.1562,	1.1318,	2.1452,	2.2245,
36	35,	1.5171,	nan,	1.1609,	1.1312,	2.1469,	2.2247,
37	36,	1.5084,	nan,	1.1531,	1.13,	2.1457,	2.2246,
38	37,	1.5113,	nan,	1.157,	1.1293,	2.1418,	2.2248,
39	38,	1.5067,	nan,	1.1528,	1.1297,	2.143,	2.2256,
40	39,	1.4883,	nan,	1.1411,	1.1296,	2.1336,	2.2258,
41	40,	1.4906,	nan,	1.1409,	1.1299,	2.1362,	2.2269,
42	41,	1.4808,	nan,	1.1359,	1.1306,	2.1269,	2.2272,
43	42,	1.4838,	nan,	1.1371,	1.1315,	2.1334,	2.2284,
44	43,	1.4849,	nan,	1.1342,	1.1321,	2.132,	2.2293,
45	44,	1.4717,	nan,	1.1291,	1.1334,	2.1283,	2.2303,
46	45,	1.4687,	nan,	1.1251,	1.1349,	2.1263,	2.2317,
47	46,	1.4603,	nan,	1.1209,	1.1371,	2.1217,	2.2338,
48	47,	1.4647,	nan,	1.1233,	1.1393,	2.1271,	2.2363,
49	48,	1.4523,	nan,	1.1158,	1.142,	2.1244,	2.2386,
50	49,	1.4607,	nan,	1.1211,	1.1448,	2.1229,	2.2411,
51	50,	1.456,	nan,	1.1157,	1.148,	2.1147,	2.243,
52	51,	1.4579,	nan,	1.1176,	1.1512,	2.1165,	2.2449,
53	52,	1.4471,	nan,	1.1103,	1.1552,	2.1202,	2.248,

YOLOv5 OBB : Evaluation d'une autre implémentation

Evaluation d'une autre implémentation d'OBB : https://github.com/hukaixuan19970627/yolov5_obb

Fonction de coût différente de la version d'ultralytics → Possibilité d'éviter les NaN

YOLO

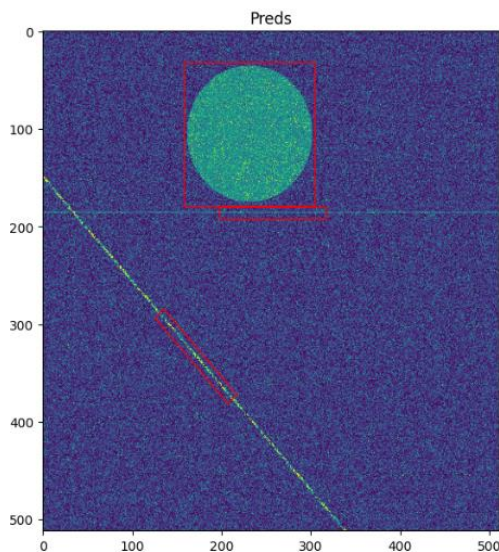
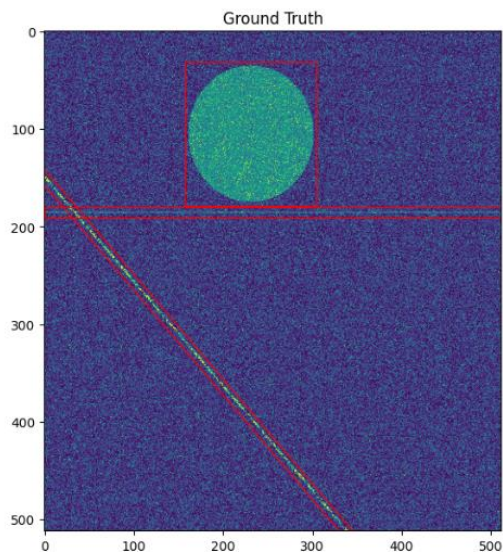
Bug

Description:

During model inference, there is a noticeable issue with the detection of thin and long objects. These objects tend to be poorly detected, with significant cropping occurring along their edges. This problem persists despite the availability of ample artificial data for training purposes.

Example:

image_12104.png | Seed: 4



AVANTIX

Discussion en cours
avec le lead dev qui
me conseille de faire
du data
augmentation

AVANTIX

MERCI