

AVANTIX

Avancement Simulateur IQ



Problème de rognage des CW/FMCW

Rognage CW/FMCW

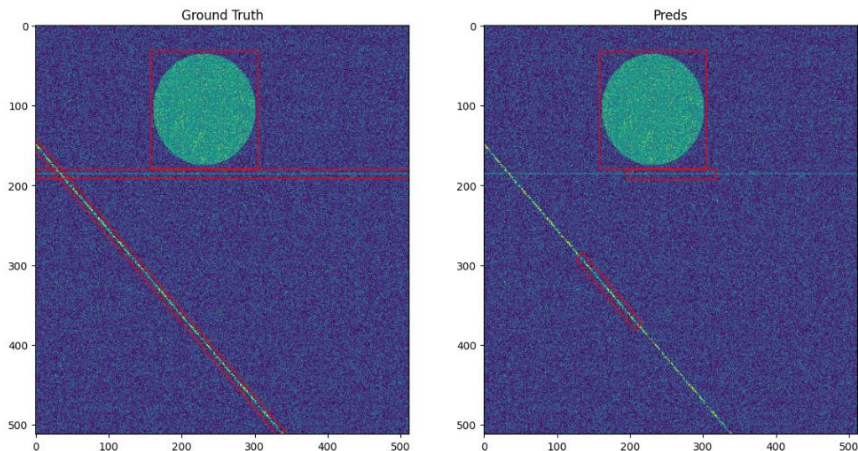
Bug

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



Discussion en cours avec le lead dev qui me conseille de faire du data augmentation et de rééquilibrer les classes

Ça fonctionne mieux mais toujours pas parfait

Essai avec Nano et Large sans HP tuning

Essai avec Nano avec HP tuning : en cours

Tentatives de résolutions

Attempts Made to avoid issue:

- ✓ IoU threshold set to 1 and 0: Unsatisfactory results observed (refer to previous post).
- ✓ Experimented with P2-P5 to assess if lower feature maps could assist with larger object detection: No discernible improvement over P3-P5.
- ✓ Current attempt involves rebalancing class proportions. Initially, long and thin objects comprised less than 20% of all rectangles. Currently adjusting to make long and thin objects approximately 80% of all classes. Training in progress; expecting results by tomorrow.

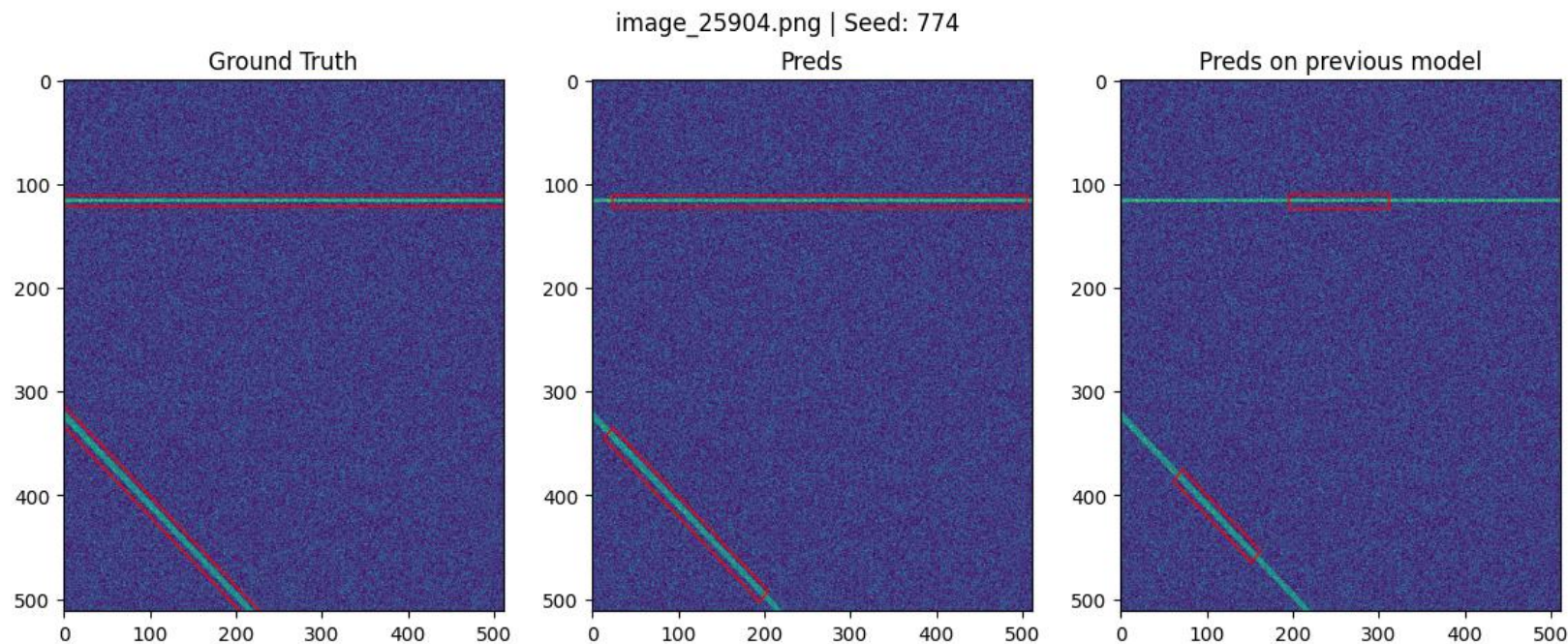
Pending Investigation:

- ☐ Deactivate completely NMS not just play with threshold
- ☐ Determine if aspect ratio filtering can be implemented within the existing codebase. Requesting confirmation from @glenn-jocher regarding the absence of such filtering.
- ☐ Explore the possibility of integrating a distinct loss function in the YOLO-OBb implementation to potentially address the issue.

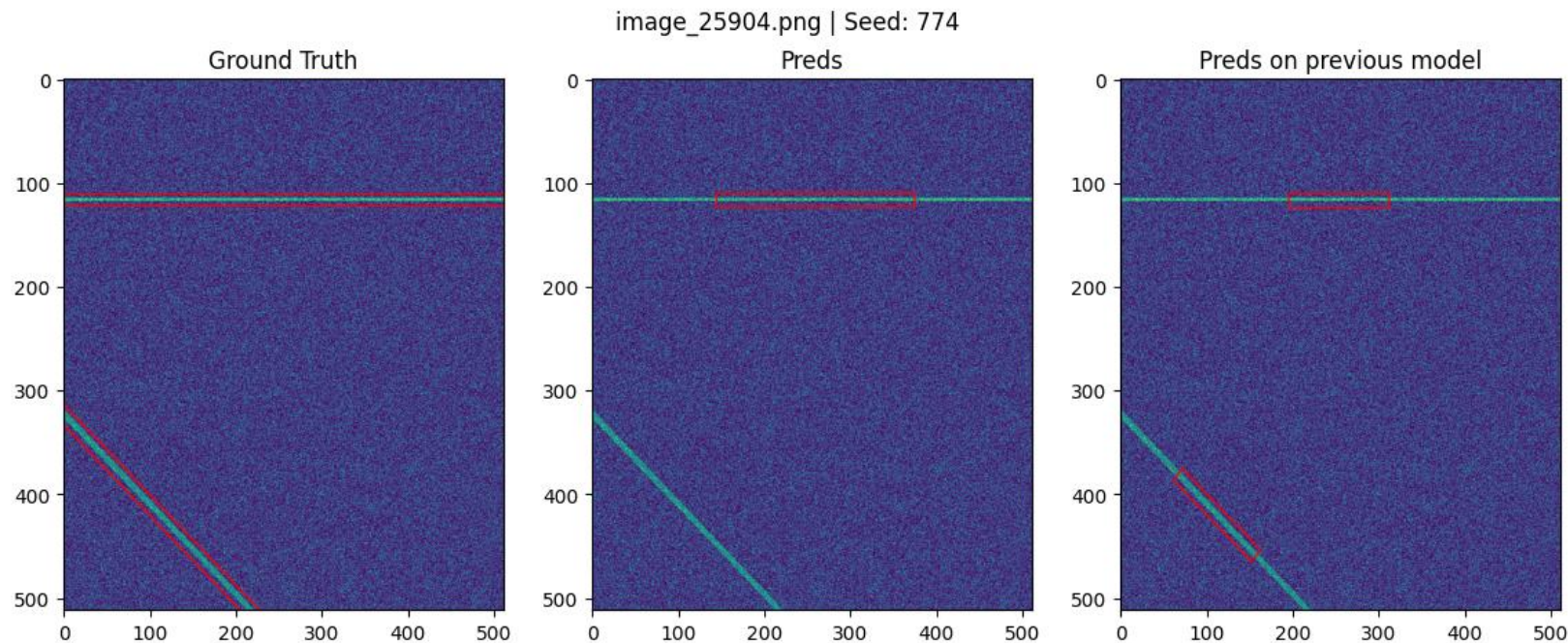


A apporter une amélioration

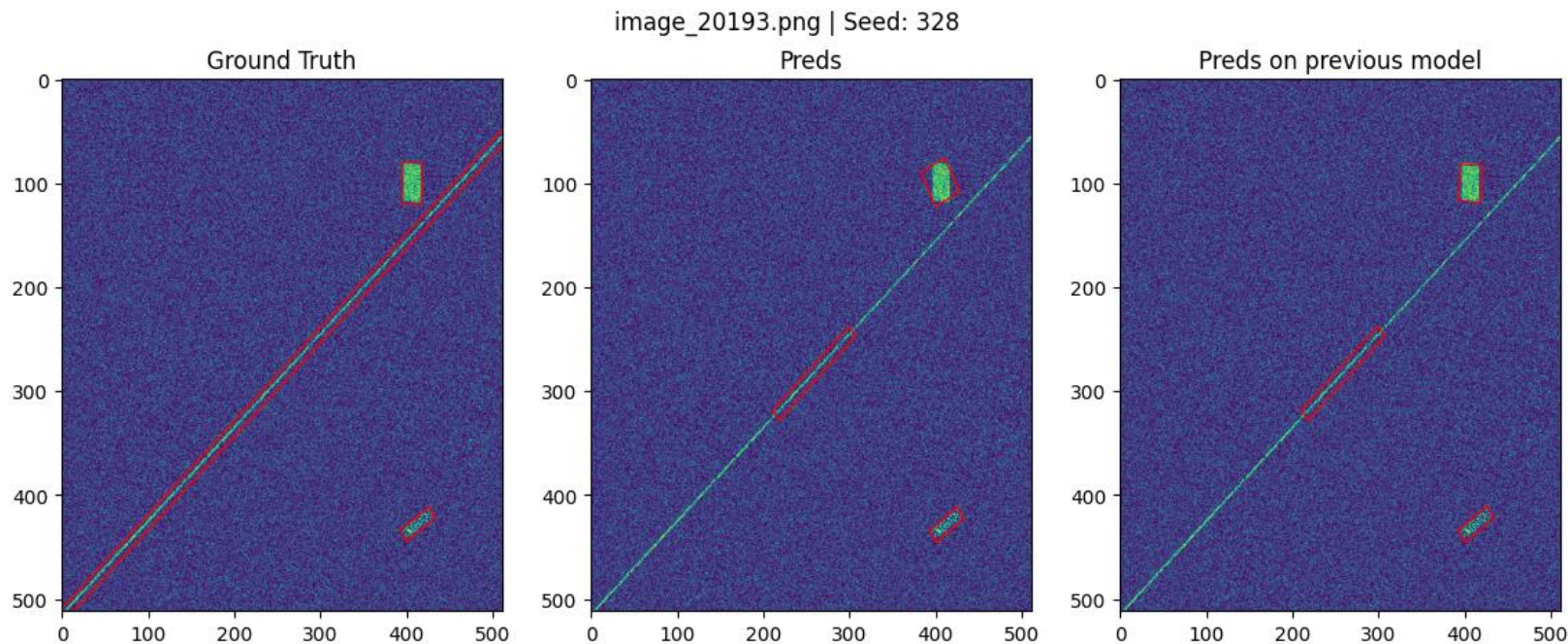
Rognage CW/FMCW : YOLO nano P2-P5



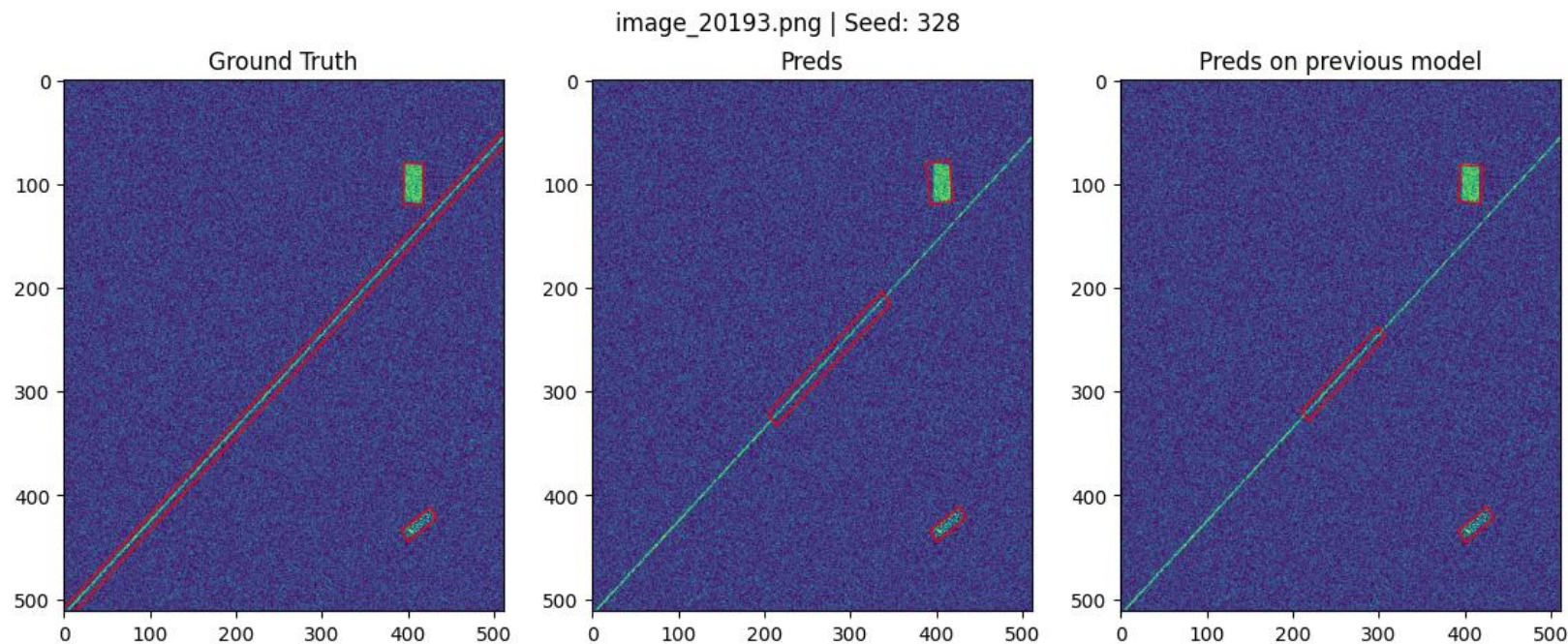
Rognage CW/FMCW : YOLO large P2-P5



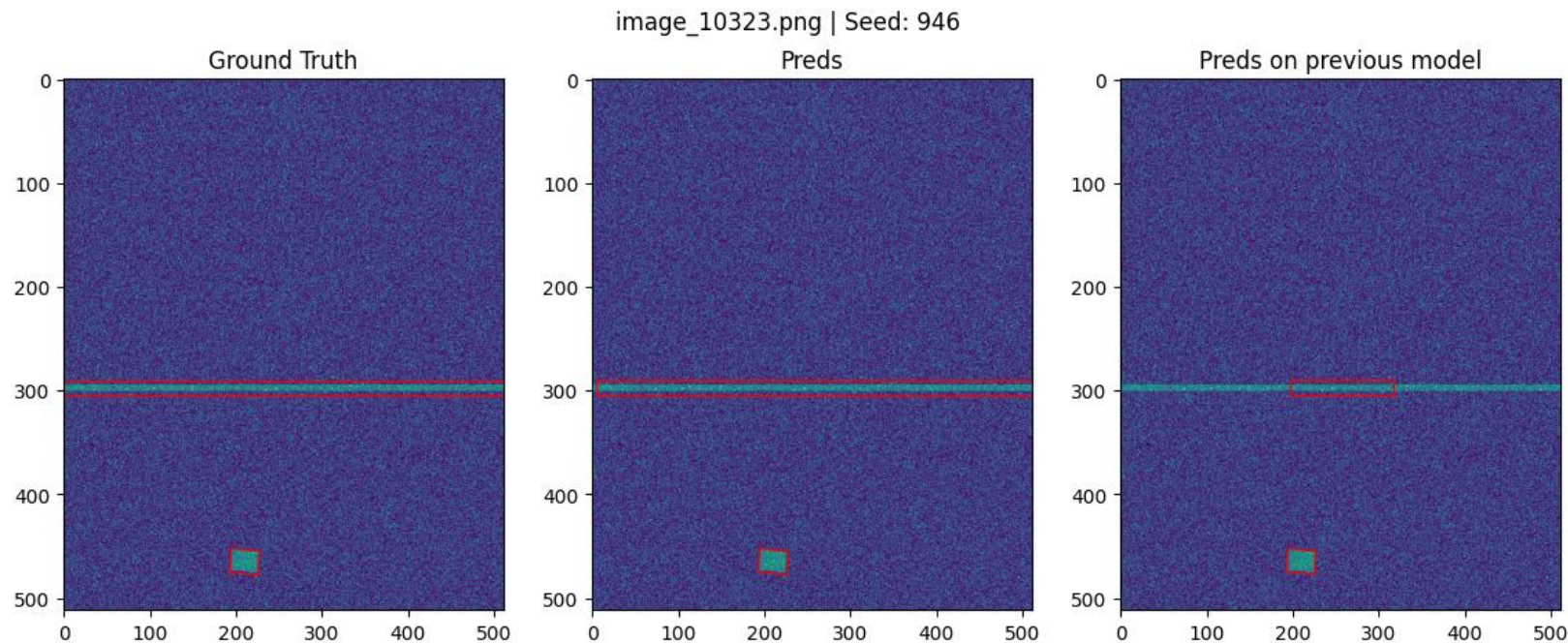
Rognage CW/FMCW : YOLO nano P2-P5



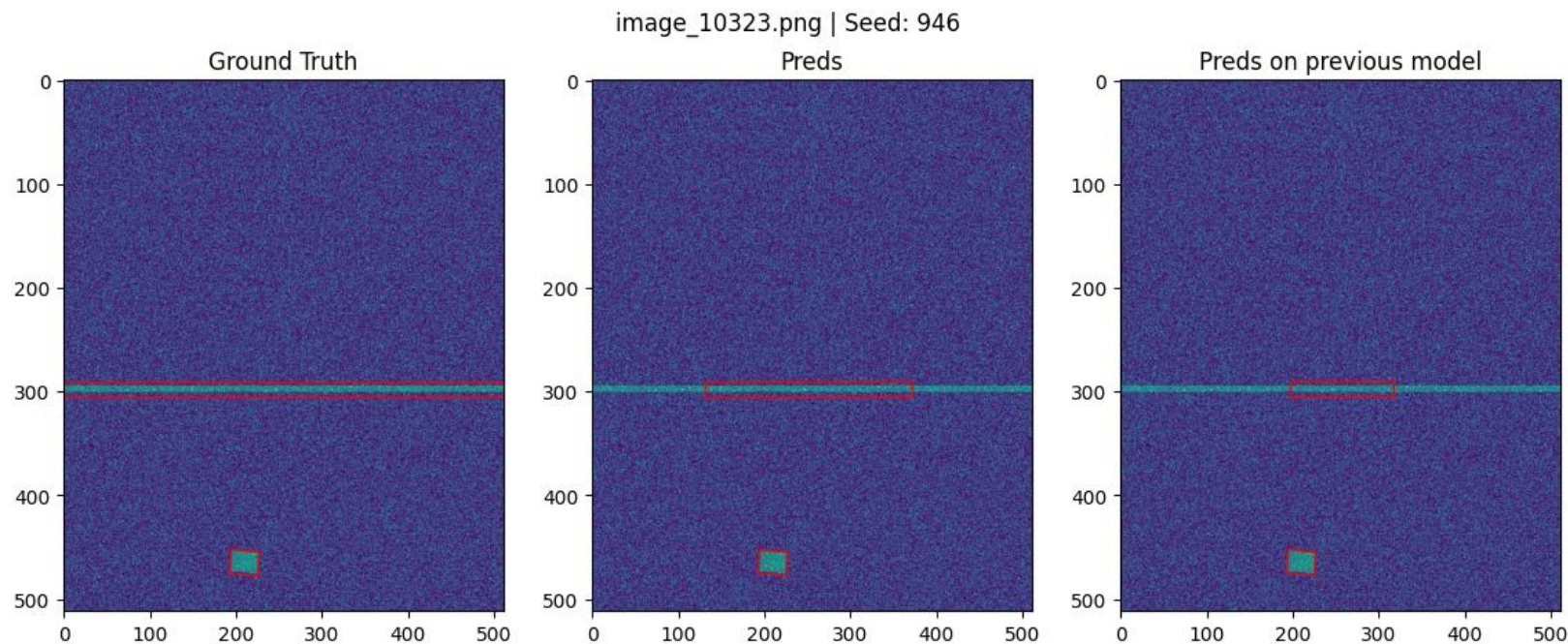
Rognage CW/FMCW : YOLO large P2-P5



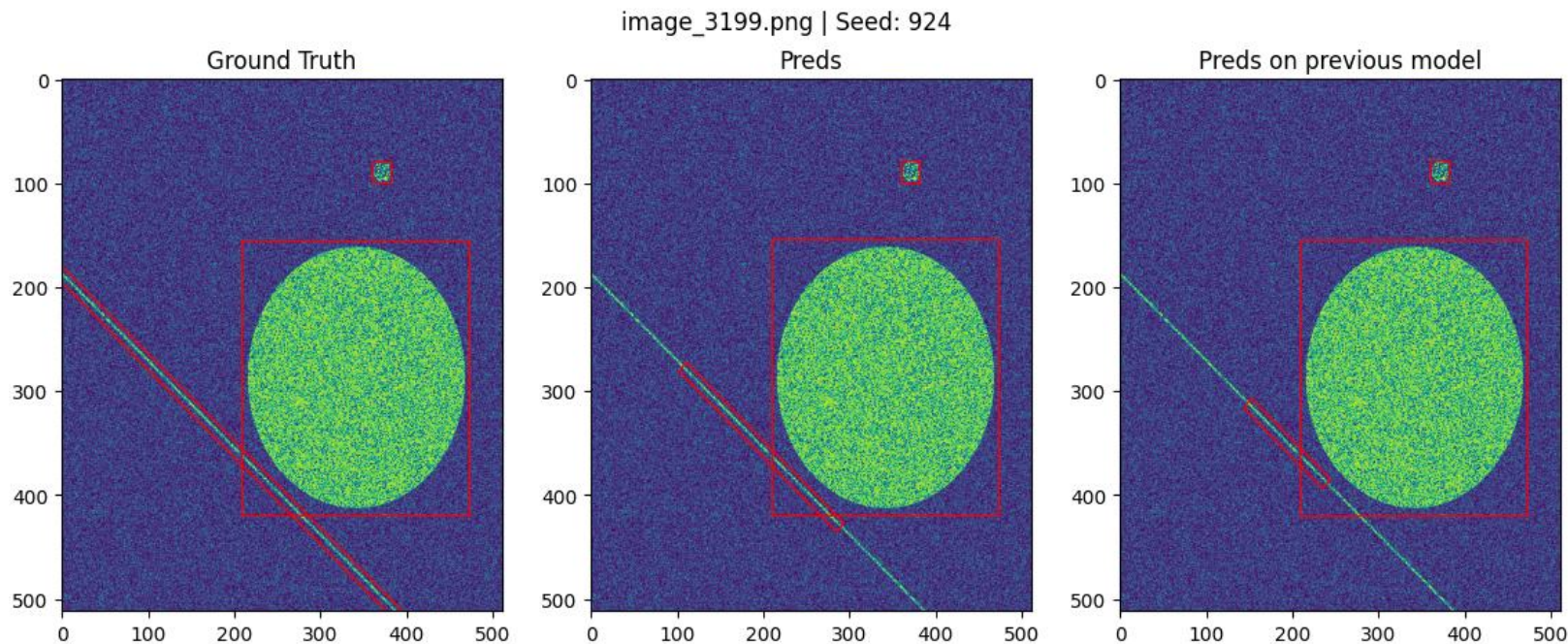
Rognage CW/FMCW : YOLO nano P2-P5



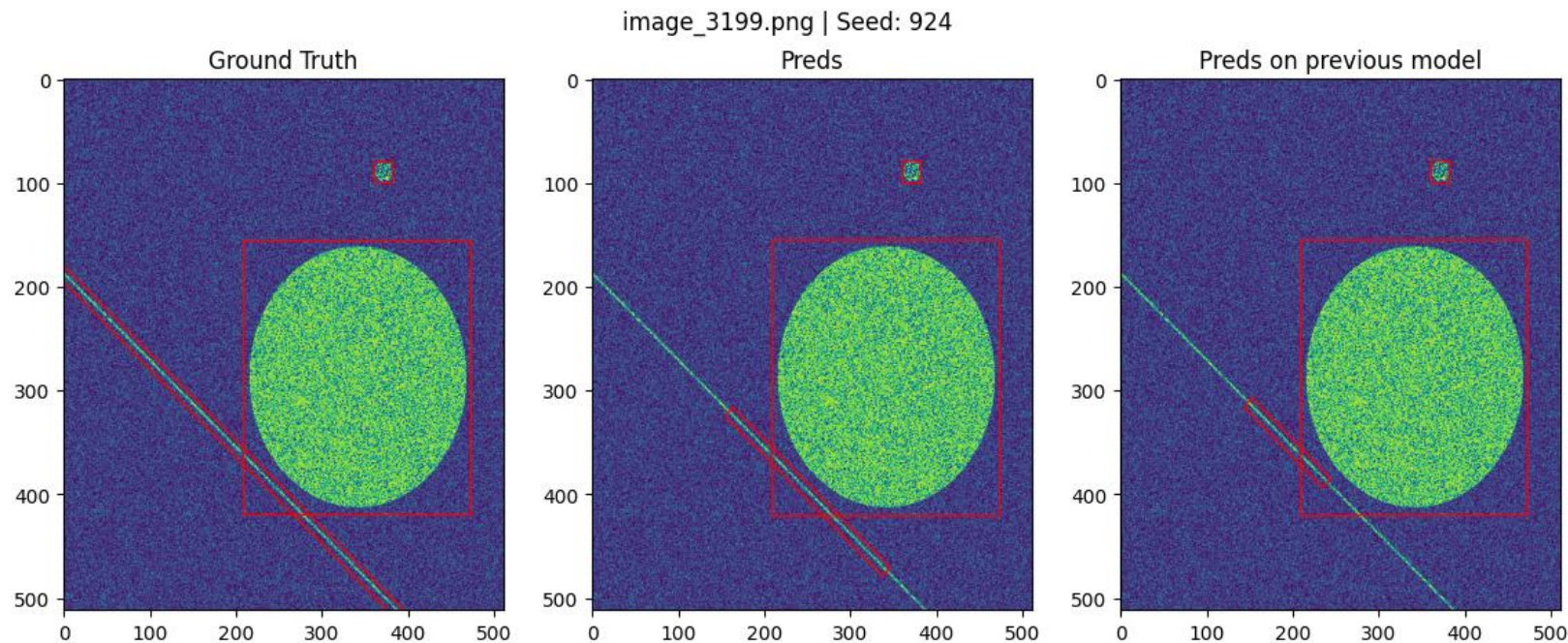
Rognage CW/FMCW : YOLO large P2-P5



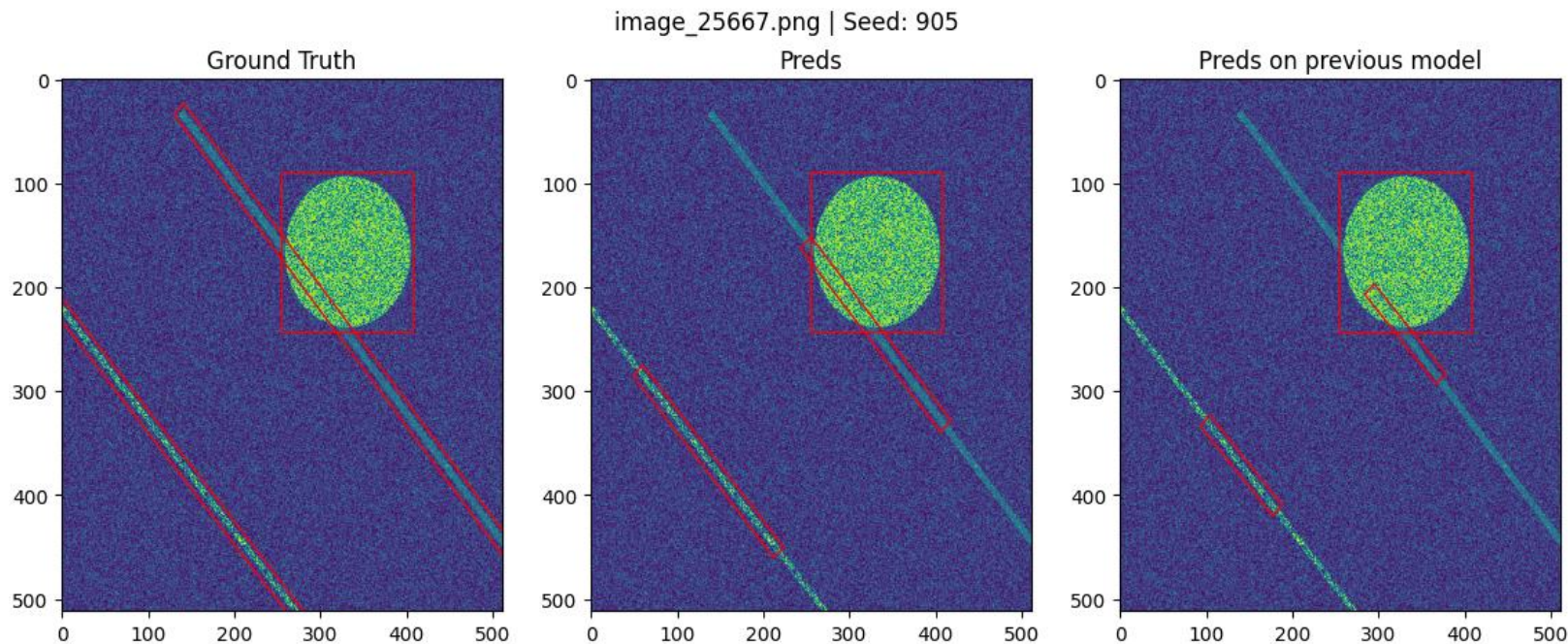
Rognage CW/FMCW : YOLO nano P2-P5



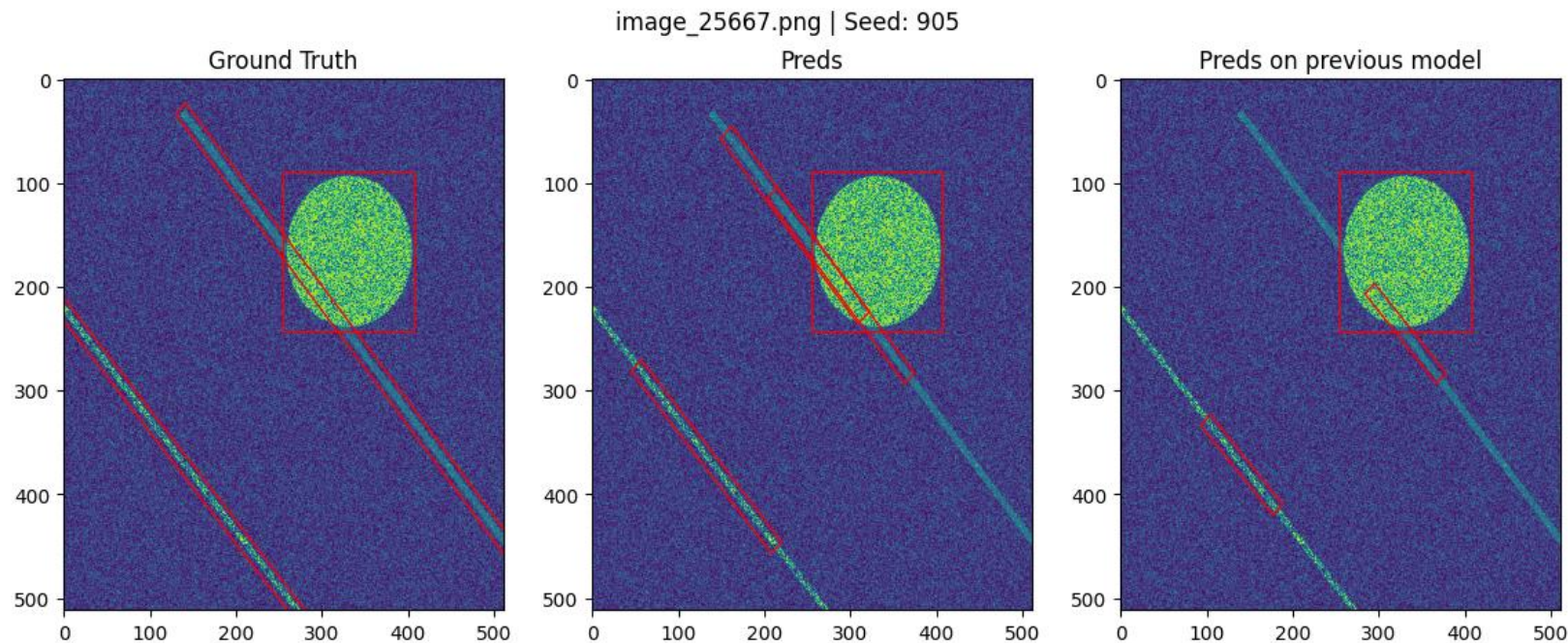
Rognage CW/FMCW : YOLO large P2-P5



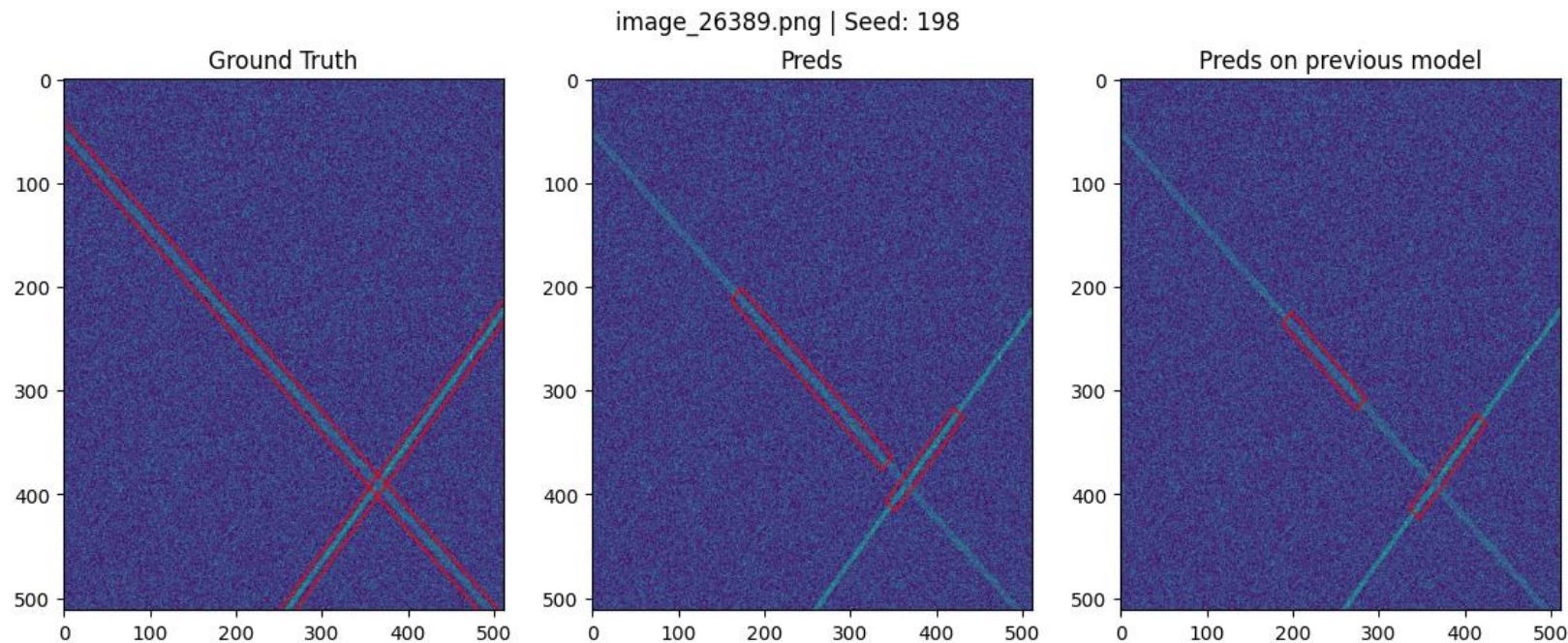
Rognage CW/FMCW : YOLO nano P2-P5



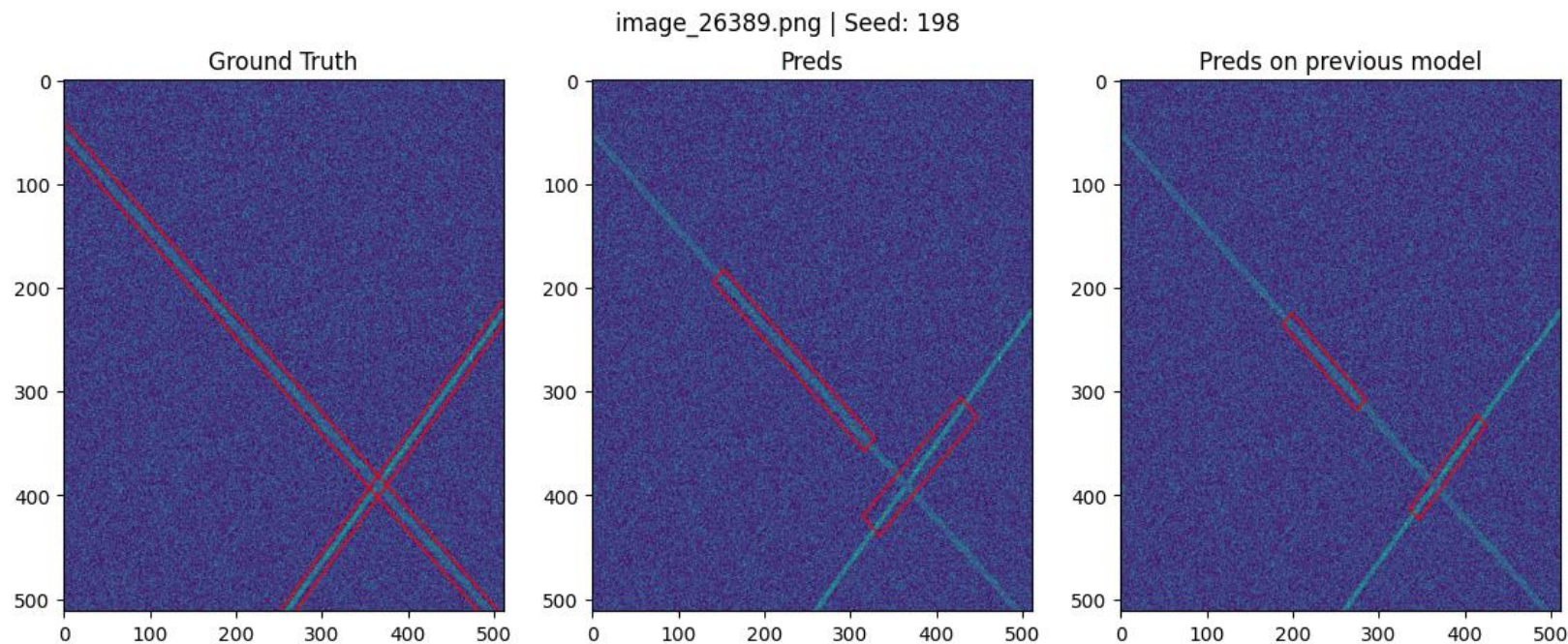
Rognage CW/FMCW : YOLO large P2-P5



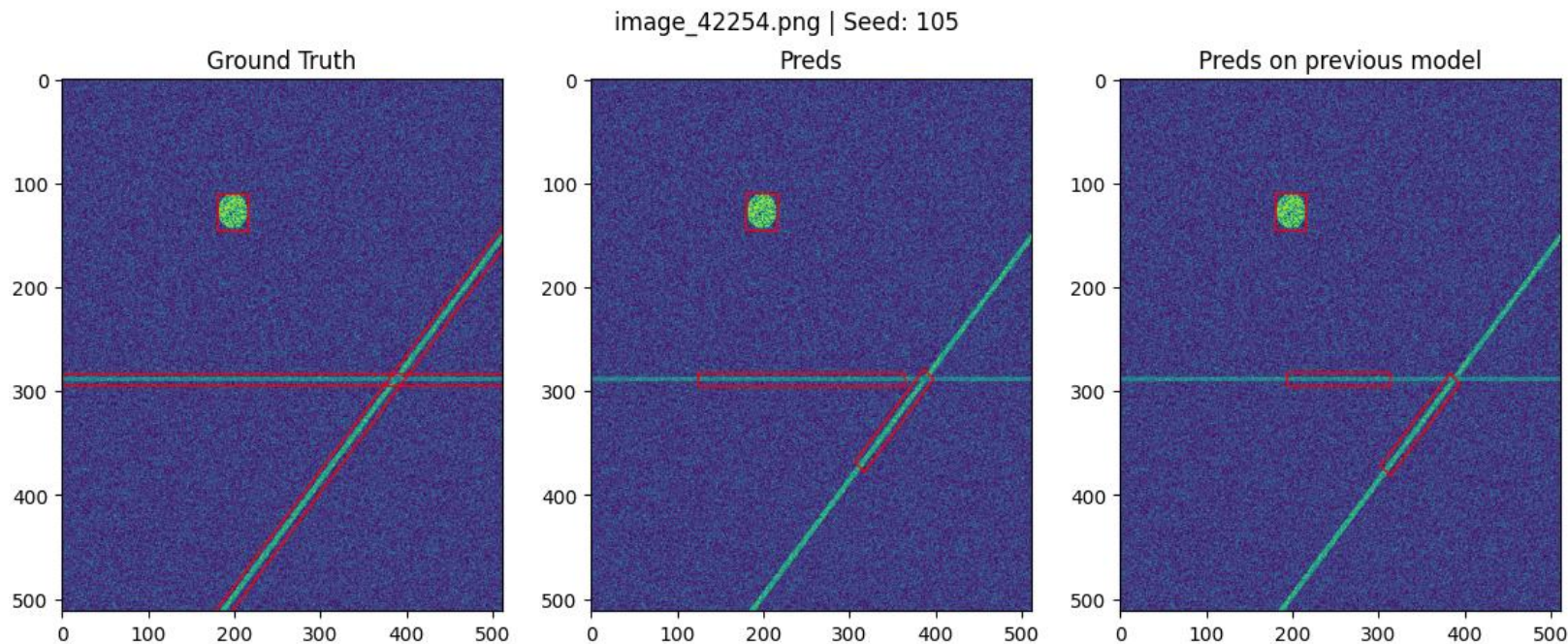
Rognage CW/FMCW : YOLO nano P2-P5



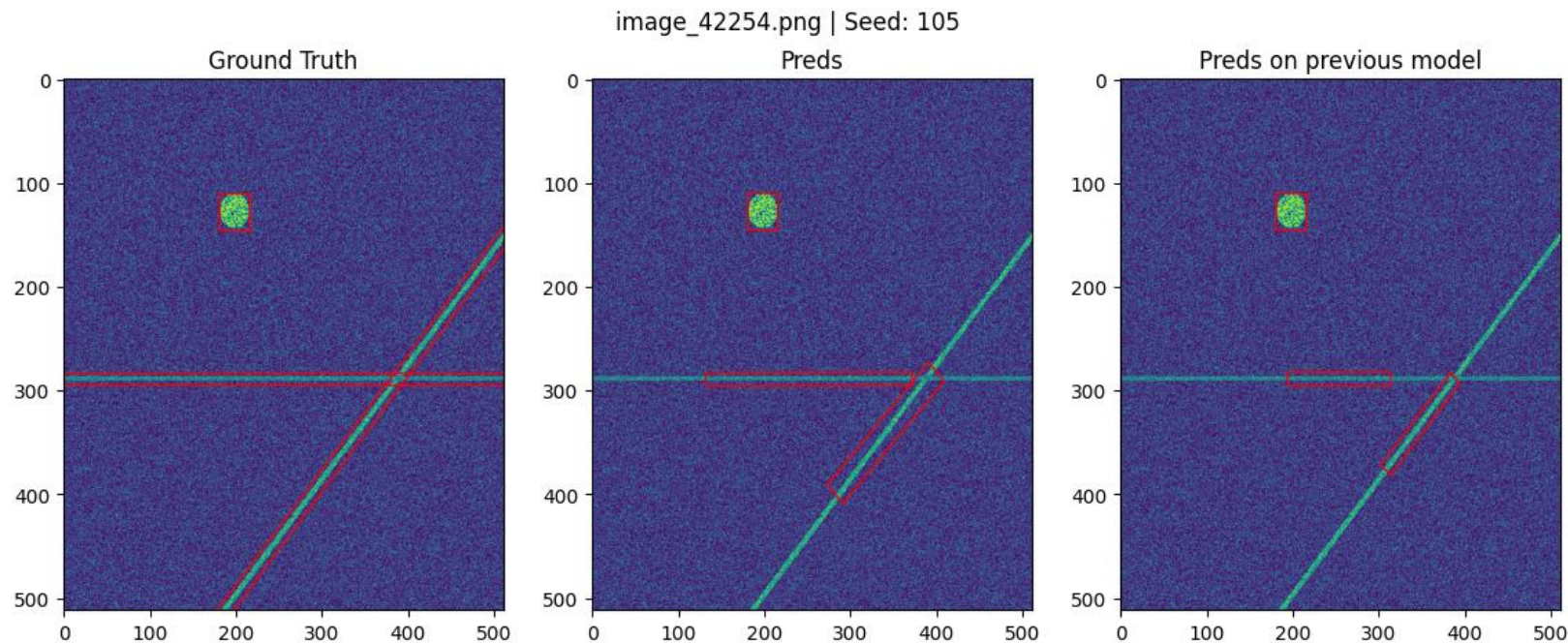
Rognage CW/FMCW : YOLO large P2-P5



Rognage CW/FMCW : YOLO nano P2-P5



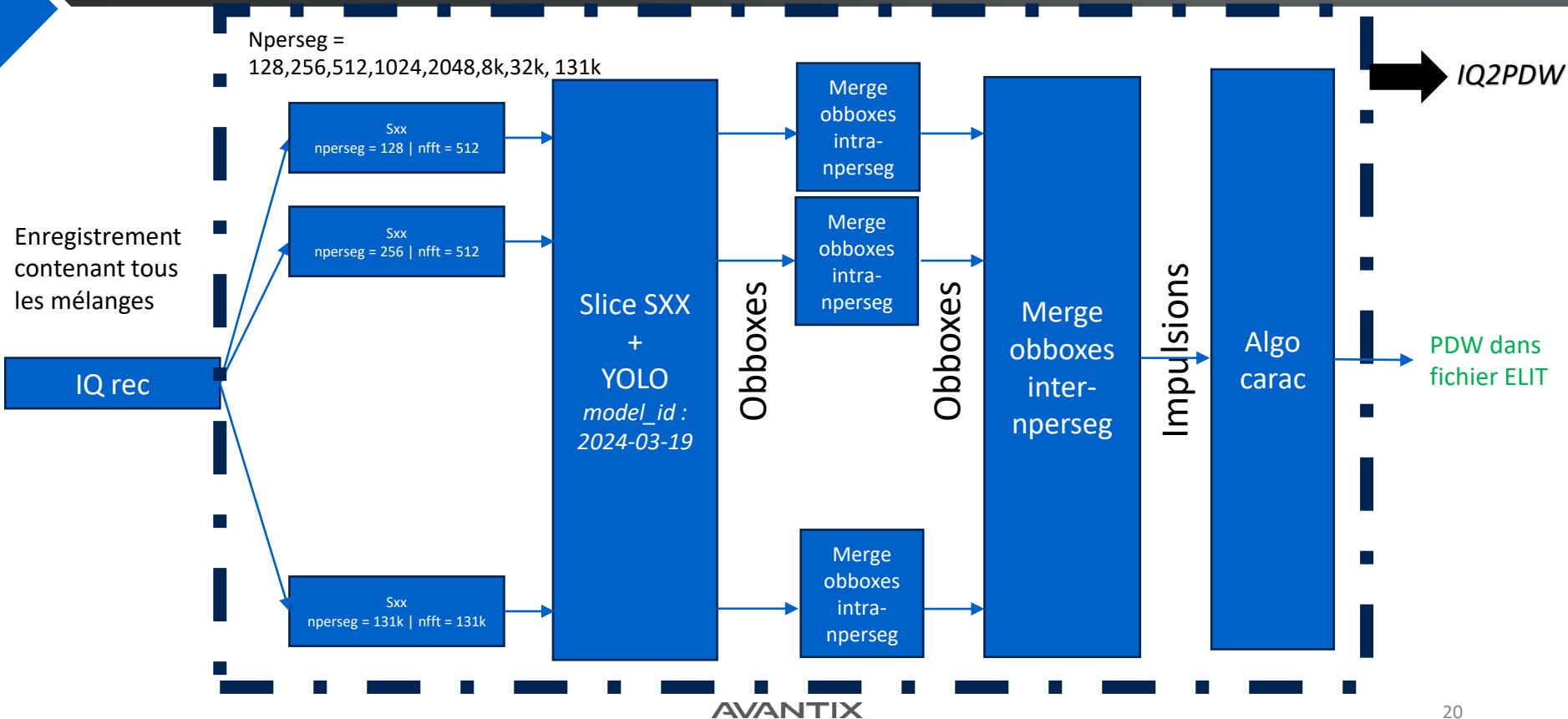
Rognage CW/FMCW : YOLO large P2-P5



AVANTIX

Fusion des obboxes intra-nperseg

Chaine de traitement IQ bout-en-bout



Proposition initiale

Avoir des spectrogrammes

1. Sans split en fréquence
2. Recouvrant au moins 2ms
3. Recouvrement temporelle de 50%

Avantage: N'avoir à réaliser que la fusion inter-nperseg

Inconvénients :

1. Chaque résolution doit être entraînée séparément (tuning très complexe)
2. Nécessite N fois plus de données avec N le nombre de résolutions différentes
3. Normalisation plus coûteuse (recherche min, max et quantile sur beaucoup plus de pixels)

Proposition exclue à cause de l'inconvénient n°1

Proposition 2

Proposition :

1. Inférer sur des images de taille fixes. Le choix se porte sur 1024x1024 car c'est une échelle mi-parcours entre tous les nperseg, il faut éviter les échelles trop grandes car les petits objets ne sont pas détectés et c'est une définition qui s'approche de celles communément utilisées
2. Créer un masque à partir des obboxes sur une définition plus grande
3. Générer les obboxes finales

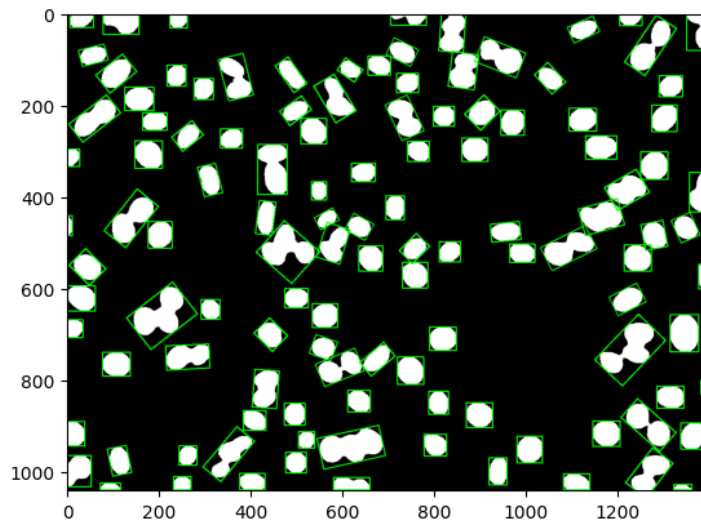
Avantage : La fusion intra-nperseg est gérée l'algo de traitement d'image ci-contre

Inconvénients :

1. Nécessité de gérer tous les cas de figure (fs + nperseg) afin de savoir combien de masque à carreler
2. Demande à réfléchir à la gestion d'un overlap de 50%
3. Solution potentiellement calculatoire
4. Ne gère pas le cas des CW/FMCW

Proposition exclue car il semble compliqué de gérer l'inconvénient 2

Algorithme permettant de recoller les obboxes



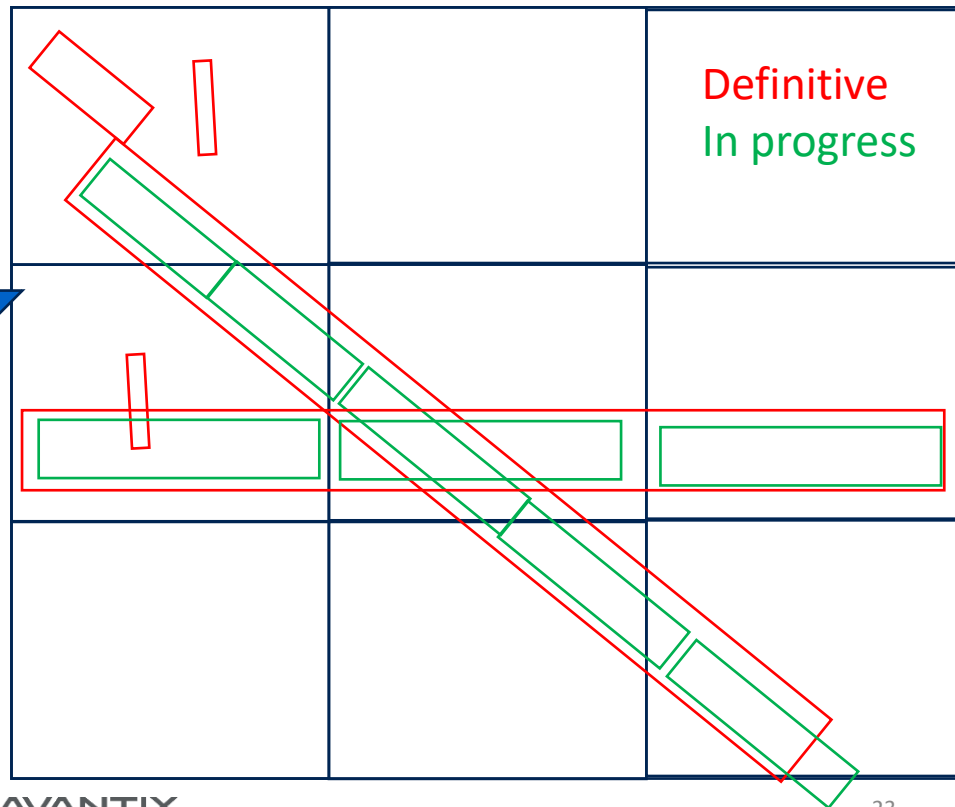
Proposition 3

1

Proposition :

1. Parcourir toutes les tuiles pour un nperseg fixée
2. Gérer 2 listes d'obboxes :
 - « Definitive » : Les obboxes dont on sait quelle délimite une impulsion
 - « In progress » : Les obboxes en bord d'images et qui peuvent tronquées des impulsions
3. A chaque passage d'une nouvelle tuile, on ajoute les obboxes « Definitive » et on vérifie que les obboxes « In progress » des précédentes tuiles ne sont pas fusionnables avec les courantes

2





Problème divergence apprentissage (NaN)

YOLO : Bug durant l'apprentissage

~~Standby : A voir si le bug persiste avec la BT200~~

Le bug s'est présenté sur les données des simples shapes

Hypothèse : Bug déclenché dû à la présence d'objets qui débordent de l'image (coordonnées des obboxes qui sortent du cadre)

Pour régler le problème :

- Soumettre le problème au dev
- Avoir des coordonnées qui dépasse le moins possibles l'image
- Padding avec zeros pour éviter les coordonnées extérieures
- Eviter d'avoir des obboxes qui sortent de l'image et en inférence gérer par du padding

Si malgré les tentatives précédents bug toujours présent → Tester avec YOLOv5 chinois

AVANTIX

MERCI