

Mer veille

Surveillance en mer

Olivier Ricou
ricou@lrde.epita.fr

Laboratoire de Recherche de l'EPITA

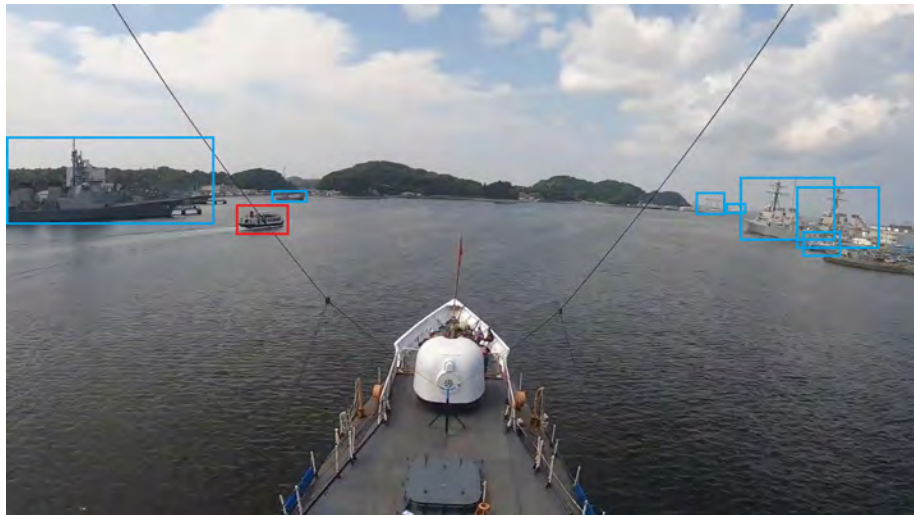
Gestion et Analyse des données Maritimes (GAM' 23)

Détecter, reconnaître et suivre les objets en mer

- **Détection** : obtenir les coordonnées des objets
- **Reconnaissance** : attribuer une classe aux objets
- **Suivi** : estimer la position future d'un objet cible, risque

Projet lancé en 2018, il n'y avait pas de solution pour la plaisance :

- un radar consomme trop
- rien ne détecte les OFNI ou les hommes à la mer



C'est un problème de traitement d'image → l'IA est bonne maintenant.



Renaud Dupuy de la Grandrive



Offshore ONE

The starting point for AI-based situational awareness on the water. 3 cameras (1 RGB, 2 thermal) and AI processing unit, all contained in one single vision unit, mounted at masthead.

€ 10.990,00 incl. VAT*

Offshore 320

Same setup as Offshore One but complemented with a powerful external processing unit. Larger detection angle. Higher detection frequency improves performance in bad sea conditions.

€ 20.990,00 incl. VAT*

Offshore 640

Same processing power as Offshore 320 but higher resolution thermal cameras. Superior detection performance (longer range, smaller objects). Higher resolution and better e-zoom capabilities.

€ 33.990,00 incl. VAT*

Produit sorti en 2022

IA entraînée sur 16 millions d'images maritimes (et vérité terrain).

Comment faire un tel système libre ?

- Avec des petites mains (trop lourd, trop cher)
- Avec des caméras qui filment la mer couplées à l'AIS (difficile à Paris)
- En générant les images

Toute suggestion est la bienvenue.



Maya semblait offrir l'outil idéal.

Générer des vidéos maritimes avec Maya (2/2)



Code en Python pour générer de nombreuses vidéos (sans GUI).

Problèmes :

- La zone de vagues demande beaucoup de mémoire
- Nos grandes vagues n'étaient pas réalistes
- Le ciel est une image fixe

Générer des images ou des films reste un métier.

Travail fait avec Charles Ginane.

Chercher les navires

Utiliser des propositions de région et détecter si on voit un navire.



YOLO v4 entraîné sur la base COCO

Échec : notre détecteur de navire ne marche pas sur des petites images de mauvaise qualité.

Cela nous a poussé à caractériser un navire quelque soit l'échelle.

Travail fait avec Tanguy Desgouttes.

Fabriquer la vérité terrain du pauvre. Idée d'origine :

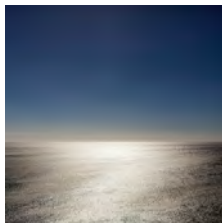
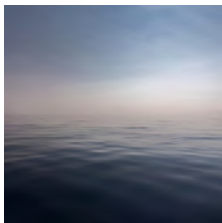
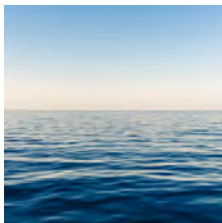
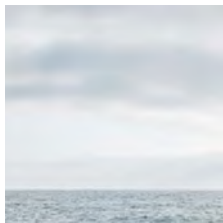
- 1 Distinguer la mer du ciel → segmentation images mer / ciel
- 2 Trouver les navires → sur-impression de navires
- 3 Trouver la côte et les rochers → sur-impression + inconnu ?
- 4 Détecter les OFNI → ce qui n'est pas reconnu ?

Bonus :

- apprentissage auto-supervisé (l'horizon est une ligne, un OFNI est dans l'eau...)
- entraînement prof / élève (un petit réseau (prof) apprend avec peu, puis il génère une vérité terrain qui permet d'entraîner un réseau plus gros (élève), etc.)

Un petit jeu de données

- 60 images de mer / ciel



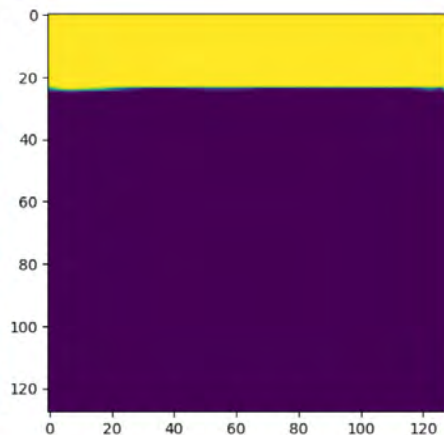
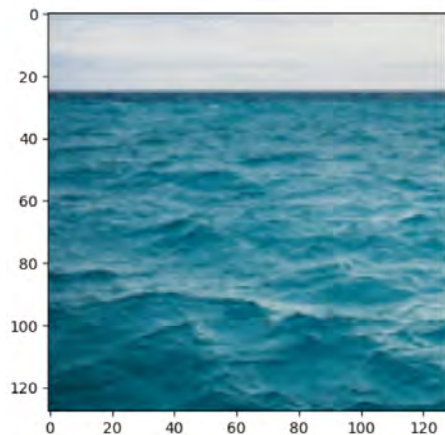
- 20 images de navires



Détection mer / ciel – segmentation

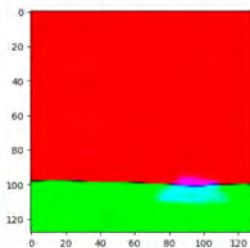
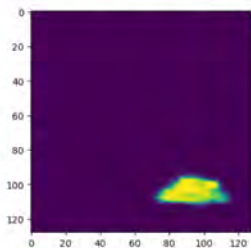
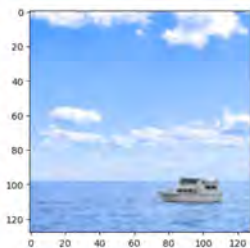
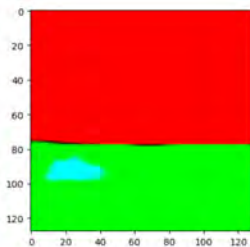
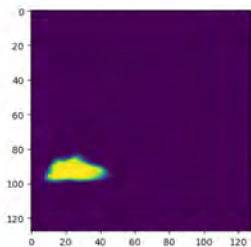
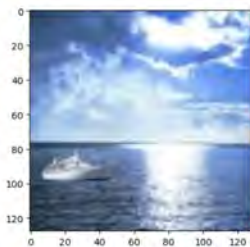
La segmentation mer / ciel marche avec un U-net 32 filtres en entrée.

La vérité terrain est la position de la ligne d'horizon.



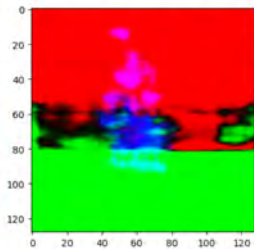
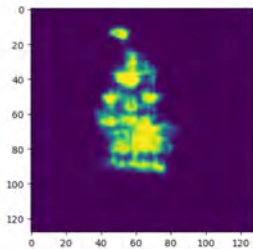
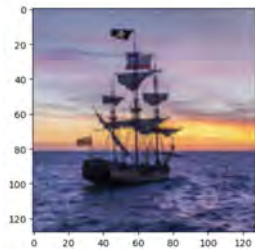
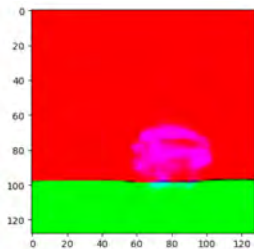
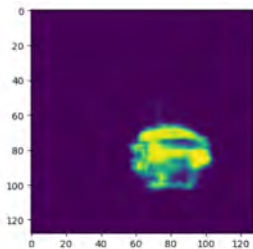
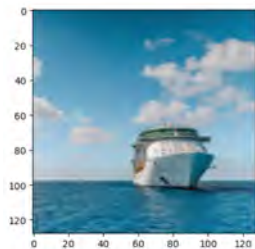
Détection des navires – collage

On ajoute des navires sur des images de ciel et mer
puis on entraîne un U-net 32 (6,3 M paramètres) :



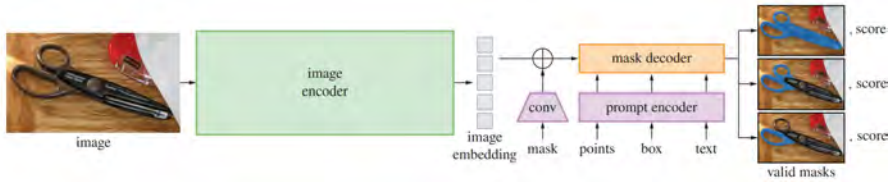
Détection des navires – image réelle

À améliorer mais l'idée semble fonctionner :



Sur les épaules d'un géant

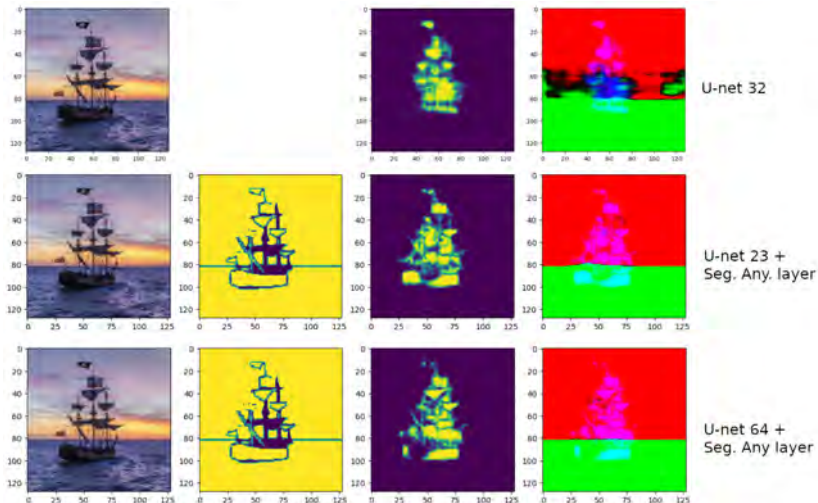
Segment Anything par Meta Research – avril 2023



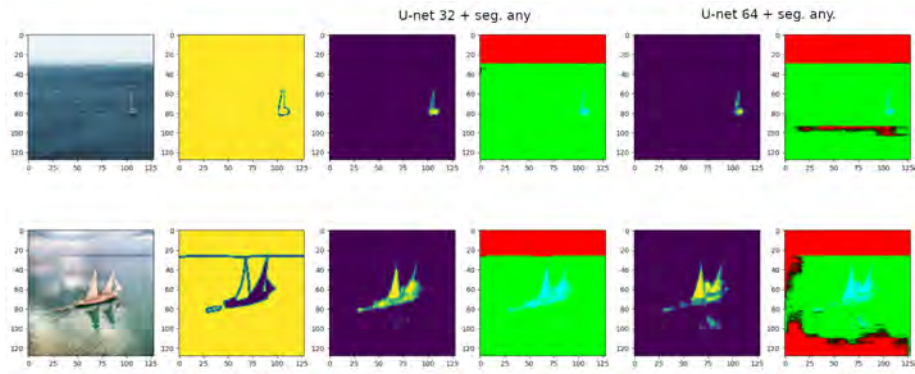
SA-1B Dataset : 11 M d'images, 1,1G masques

U-net + segment anything

Segment Anything → un canal de plus



Lorsque le modèle est trop grand



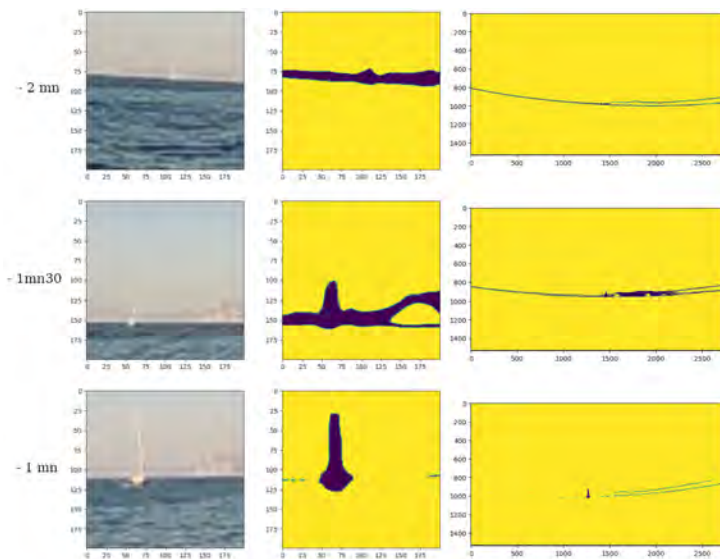
Parfois U-net 64 (l'original) réussit nettement moins bien que U-net 32 (l'inverse n'arrive pas sur une dizaine d'images réelles).

- Jeu de données trop petit pour U-net 64 ?
- U-net 32 est moins précis et donc généralise mieux ?

L'horizon est à 2 mn 30



Segment Anything marche bien



Mais le navire n'est jamais détecté en tant qu'objet (contour des autres).

Segment Anything est lourd

640 M de paramètres

Extraction des masques d'une image 2K : 23s sur une Quadro P6000

Temps réel = 1 image / seconde

NVIDIA propose des cartes GPU à embarquer :

- Jetson Orin Nano : 40 TOPs pour une conso entre 5 et 15 W.
- Jetson Orin AGX : 200 TOPs pour une conso entre 15 et 40 W¹.

Quadro P6000 : environ 80 TOPs

Il manque un facteur 50 pour faire du temps réel (avec un Jetson Nano).

¹Un pilote automatique consomme 25 W, un Raspberry Pi consomme 6 W

Segment Anything est lourd

640 M de paramètres

Extraction des masques d'une image 2K : 23s sur une Quadro P6000

Temps réel = 1 image / seconde

NVidia propose des cartes GPU à embarquer :

- Jetson Orin Nano : 40 TOPs pour une conso entre 5 et 15 W.
- Jetson Orin AGX : 200 TOPs pour une conso entre 15 et 40 W¹.

Quadro P6000 : environ 80 TOPs

Il manque un facteur 50 pour faire du temps réel (avec un Jetson Nano).

Segment Anything peut permettre d'entraîner un réseau plus léger.

¹Un pilote automatique consomme 25 W, un Raspberry Pi consomme 6 W

Difficulté de voir et reconnaître un navire qui se confond avec l'horizon.

La vérité terrain peut être faite automatiquement si

- on a la détection de navires bien visibles
- on sait caractériser le navire suivi
- on a le suivi

Dans ce cas on passe le film à l'envers.



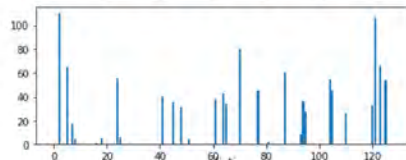
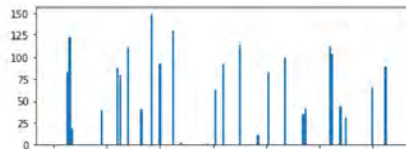
images séparées de 5 secondes

Signature d'un navire – 1/2

On veut identifier un navire pour le suivre → signature.

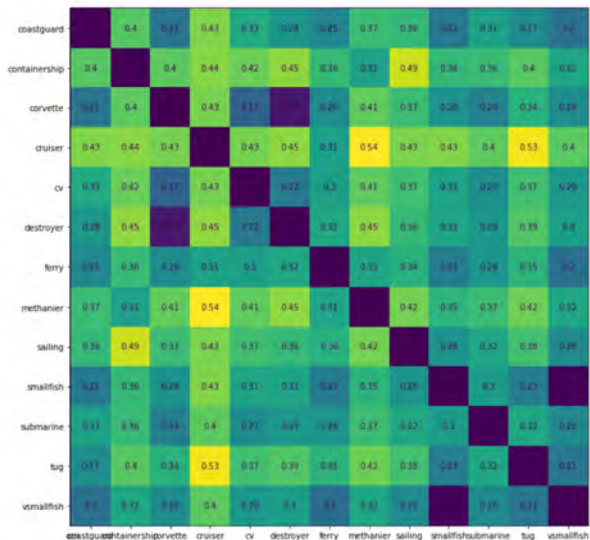
Base de 65 000 navires dans 13 catégories.

Utilisation d'un réseau basé sur Efficient Net v2 + *Contrastive loss* pour maximiser la distance entre les catégories.



Travail fait avec Tanguy Desgouttes.

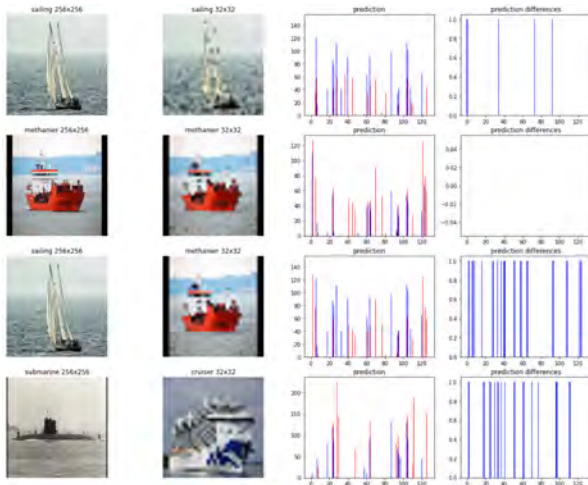
Signature d'un navire – 2/2



Moyenne distance cosinus de chaque navire à la référence de chaque classe.

Invariance de la signature au zoom

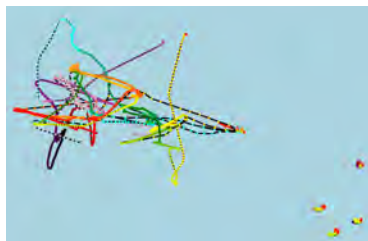
On veut identifier un navire quelque soit la distance.



$\text{Différence}[i] = 1$ si $\text{image1}[i] = 0$ et $\text{image2}[i] > 0$ ou l'inverse

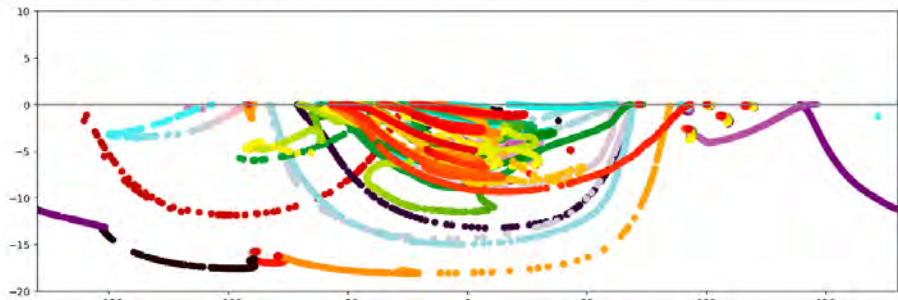
Suivi de trajectoire – l’AIS

timestamp	lon	lat	course	heading	speed
2022-05-23 21:17:59+00:00	-2.641332	47.153238	109.200000	107.000000	NaN
2022-05-23 21:18:59+00:00	-2.639733	47.152833	109.100000	106.000000	4.2
2022-05-23 21:19:59+00:00	-2.638110	47.152478	110.200000	109.000000	NaN
2022-05-23 21:20:59+00:00	-2.636467	47.152078	109.300000	112.000000	4.3
2022-05-23 21:21:59+00:00	-2.634807	47.151697	108.500000	109.000000	NaN

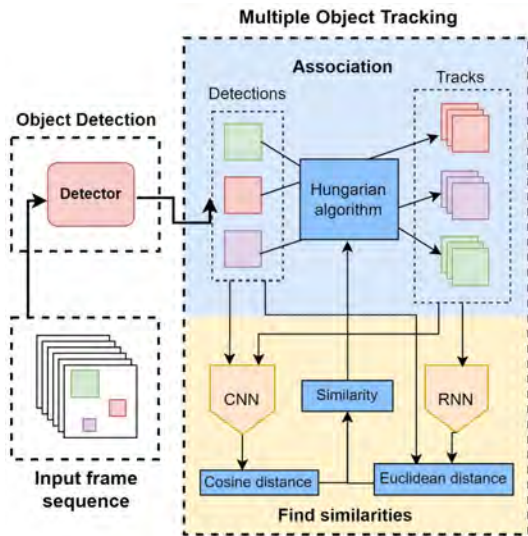


```
df = to_circular_vision(positions, np.array((47.127, -2.53)))
```

60634 position de navires vus en dessous de l'horizon, c.à.d à moins de 6736 m.



Suivi des trajectoires combiné aux détections



Le RNN prédit la position suivante. Elle peut être trouvée avec

- un calcul à l'ordre 1 :
$$x_{t+1} = x_t + \delta t_{t,t+1} * v_t$$
- un filtre de Kalman (utile si les mesures sont approximatives)
- un réseau neuronal récurrent (RNN)

Erreur de prédiction à l'ordre 1

En utilisant les traces d'AIS

- qui bougent (pas au port ou à l'encre)
- qui donnent leur vitesse

On a pour 644 traces :

- erreur médiane : 9,7 m
- erreur moyenne : 11,9 m
- écart type : 12,3 m
- temps moyen en 2 relevés : 16,4 s
- vitesse moyenne : 2 m/s

Note : si on calcule la vitesse à partir des 2 positions précédentes, on augmente l'erreur de 50 %.

Avez-vous des questions ?

