

# Airbus Ship Detection Challenge

## Final project oral presentation

T. Rogy & A. Boissier

Deep Learning for Computer Vision

August 31, 2021



- ① Motivation (in English)
- ② Problems and Implementations
- ③ Results & Figures
- ④ Conclusion and room for improvement

- ① Motivation (in English)
- ② Problems and Implementations
- ③ Results & Figures
- ④ Conclusion and room for improvement

# The shipping industry and innovations

- 1960s - : Shipping has become the main driving force for international commerce
- Example of innovations in the shipping sector : containers (1950s, *Malcom McLean*), *Liquified Natural Gas* (LNG) carriers (1959) ...

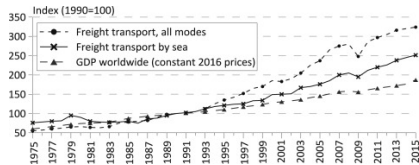


Figure 1: International freight transport

Blind spot : data analytics and computer vision, still underexploited

# Tensions and the need for efficient surveillance

Ships have become the center for economic, geopolitical and environmental tensions, which have contributed to increase :

- crimes **targetting** ships (e.g : international piracy)
- crimes **committed by** ships (e.g : used oil spills, drug smuggling ...)

## Study: In Nigeria, Illegal Fishing is Linked With Crime and Piracy



## DRUGS

## Meth haul on Thai cargo ship in Australia linked to Hong Kong drug trafficking boss



Published 3 weeks ago on Thursday, May 26, 2021  
by Maya Taylor



Figure 2: Examples of crimes related to shipping traffic

→ Need for a way to quickly monitor and detect ships through satellite imagery (ship surveillance)

# Why this challenge ?

Huge added value in fast and reliable ship monitoring

Airbus is one of the leading actors in the ship surveillance industry



Figure 3: Wanted image segmentation model

**Goal :** produce an efficient image segmentation model to create bounding boxes around detected ships



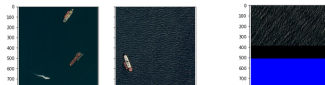
- ① Motivation (in English)
- ② Problems and Implementations
  - Problématiques
  - Building blocks principaux
- ③ Results & Figures
- ④ Conclusion and room for improvement



# Jeu de données

## Images :

- images satellites issues de l'océanographie (pleine mer, ports)
- $\approx 200\,000$  images (70 % unique)
- 2000 images corrompues (e.g lignes bleues ou vertes)



## Run-Length Encoding (RLE) :

- Encodage pour la compression de données
- Encode le nombre de pixels " actifs " se suivant



- ① Motivation (in English)
- ② Problems and Implementations  
Problématiques  
Building blocks principaux
- ③ Results & Figures
- ④ Conclusion and room for improvement

# Problèmes avec le jeu de données

- 1 Jeu de données **non équilibré** en défaveur de la classe intéressante (bateaux)
- 2 Jeu de données très lourd (  $> 29$  GB) : API de **kaggle** incontournable ...
- 3 Traitement des données : corruption + bateaux/images identiques ( $\rightarrow$  overfit ?)

# Problèmes avec l'IDE (Kaggle)

- ❶ Pas créé pour la collaboration en temps réel (→ colab ?)
- ❷ Problèmes liés à l'utilisation d'une IDE free (limitation GPU, RAM ...)

```
RuntimeError: CUDA out of memory. Tried to allocate 256.00 MiB (GPU 0; 15.90 GiB total capacity  
1 GiB already allocated; 129.75 MiB free; 15.01 GiB reserved in total by PyTorch)
```

Figure 6: Erreur lié à la limitation de RAM de kaggle `model.summary()`

- ① Motivation (in English)
- ② Problems and Implementations
  - Problématiques
  - Building blocks principaux
- ③ Results & Figures
- ④ Conclusion and room for improvement



# Architecture pour la segmentation

Nombreux candidats pour les architectures de segmentation

Historical use case proche du notre - 2015, *Freidburg University*

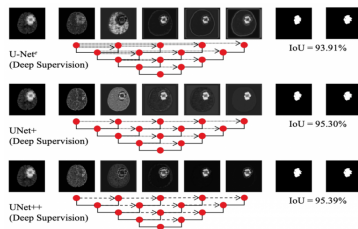


Figure 8: Historical use case du U-Net

→ choisir le U-Net le plus simple et améliorer de manière incrémentale

## Quelques améliorations

Essais empiriques sur la structure du réseau montrent que :

- ① La structure initiale répond plutôt bien au problème
- ② Intéressant de complexifier la structure mais attention car augmenter le nombre de blocs "up" ou "down" peut affecter la performance + le temps d'entraînement
- ③ Finalement : ajout de 3 blocs up et down + modification structure interne des blocs

<pre> ===== Total params: 31,390,721 Trainable params: 31,390,721 Non-trainable params: 0 ===== Input size (MB): 6.75 Forward/backward pass size (MB): 8905.50 Params size (MB): 119.75 Estimated Total Size (MB): 9032.00 ===== </pre>	<pre> ===== Total params: 81,829,857 Trainable params: 81,829,857 Non-trainable params: 0 ===== Input size (MB): 6.75 Forward/backward pass size (MB): 2027.95 Params size (MB): 312.16 Estimated Total Size (MB): 2346.86 ===== </pre>
---	---

Figure 9: Comparaison des paramètres du réseau initial (*gauche*) vs réseau amélioré *droite*



# Choix d'une fonction de loss

$$Dice = \frac{2|X \cap Y|}{|X| + |Y|}$$

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$BCEDice(w) = -\frac{1}{N} \sum_{n=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log (1 - \hat{y}_n)]$$

Certaines loss génèrent des meilleures formes de bateaux, mais obligent la création de formes plus homogènes, rendant leur séparation plus difficile. Les loss basées sur les BCE aboutit à des formes moins bonnes, mais une meilleure chance de séparation.

- ① Motivation (in English)
- ② Problems and Implementations
- ③ Results & Figures
  - Building blocks
  - Convergence and overfitting issues
- ④ Conclusion and room for improvement

- ① Motivation (in English)
- ② Problems and Implementations
- ③ Results & Figures
  - Building blocks
  - Convergence and overfitting issues
- ④ Conclusion and room for improvement

# Epochs

## Rendements décroissants :

- croissance du score qui diminue
- score sur le ladder public et privé décroît même après la 6ème époque

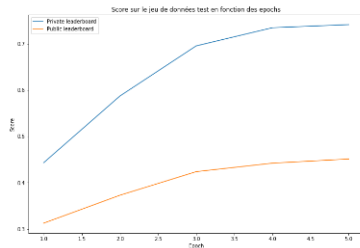
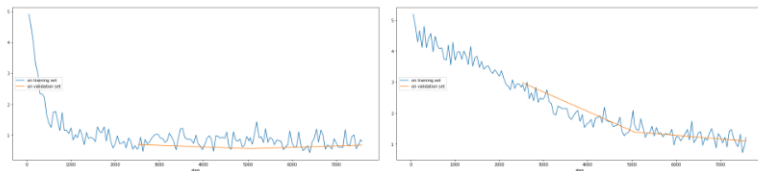


Figure 10: Influence du nombre d'époques sur les résultats finaux

# Architecture du réseau

"Convergence" plus tôt d'une époque par rapport à notre modèle → **performance satisfaisante.**



**Figure 11:** Comparaison de la loss des réseaux : préentraîné vs. notre architecture

Quant à la comparaison U-net vs U-net amélioré, la loss décroît elle aussi plus rapidement sur le U-net amélioré sans overfitting

# Augmentation d'image

## Gain de performance très léger :

- pas d'augmentation de la taille du jeu de données
- modifications (trop) légères pour pouvoir traiter des situations variées ?

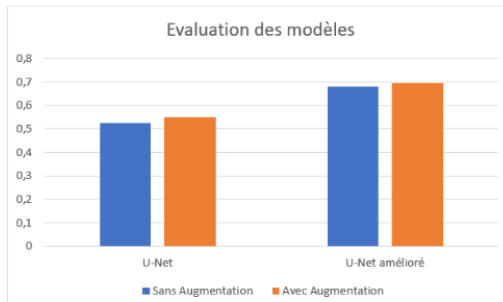


Figure 12: Impact de l'augmentation sur le score final

- ① Motivation (in English)
- ② Problems and Implementations
- ③ Results & Figures
  - Building blocks
  - Convergence and overfitting issues
- ④ Conclusion and room for improvement

# Convergence du réseau

## Comportement des loss :

- Descente rapide de la loss de train sur la première epoch
- Stagnation des loss après la 2nde epoch

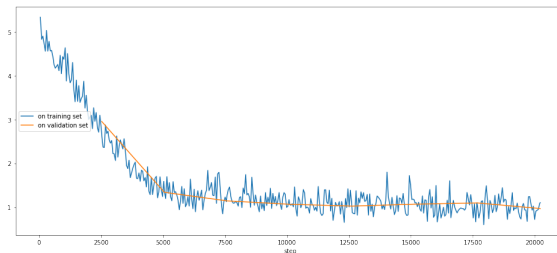


Figure 13: Graphique de la loss sur 7 epochs



# Overfitting

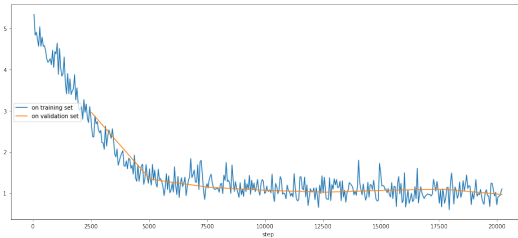


Figure 14: Graphique de la loss sur 7 épochs

Submission and Description	Private Score	Public Score
<a href="#">submission-4.csv</a> 18 minutes ago by <a href="#">Theophile Rogy</a>	0.64755	0.39322

Figure 15: Après 6 epochs, le score décroît à la fois sur le ladder publique et privé

- ① Motivation (in English)
- ② Problems and Implementations
- ③ Results & Figures
- ④ Conclusion and room for improvement

# Conclusion

Application pure et simple des méthodes du cours ne fonctionnait pas : il a fallu trouver d'autres architectures pour la segmentation

Le U-Net a été un candidat idéal : même si son implémentation est plus complexe (blocks ups vs down), ses performances sur ce problème sont satisfaisantes :

- 42,4 % sur le leaderboard publique
- > 69,9 % sur le leaderboard privé

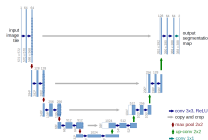
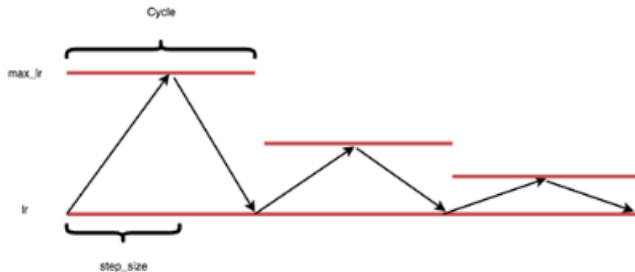


Figure 16: Architecture du U-Net

# Room for improvement

- 1 Bibliothèques complémentaires pour contrôler nos hyperparamètres
- 2 Réseau pré-entraîné sur jeu de données plus généraliste qu'ImageNet
- 3 Variation du learning rate
- 4 Architecture différente (eg : Mask R-CNN)



## Room for improvement (2)

- 1 Post-traitement des résultats : "rectangularisation" + séparation
- 2 "Stacker" avec un classifieur ships/no ships pour éviter les Faux Positifs ("wave glare")

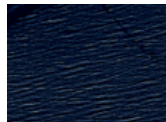


Figure 17: Wave glare is a common cause of False positives

3

Merci de votre attention !