

**AIRBUS**

**ECE**  
ÉCOLE D'INGÉNIEURS  
ENGINEERING SCHOOL



## Rapport de projet

### SHIPMARK

<2675>-<2022/11/29>

29/11/2022

## Table des matières

<b>Abstract</b>	<b>3</b>
<b>Remerciements</b>	<b>5</b>
<b>1. Exigences / Opportunités</b>	<b>5</b>
<b>2. État scientifique &amp; technique</b>	<b>7</b>
a) Réseaux neuronaux convolutifs profonds	8
b) Masque RCNN	8
c) YOLO (You Only Look Once)	11
d) COCO	12
e) ResNet 50	12
f) CenterNet	12
<b>3. Solution proposée (valeur ajoutée)</b>	<b>13</b>
<b>4. Résultats</b>	<b>15</b>
A. Annotation manuelle	15
B. Annotation automatique	16
C. Les modèles de machine learning	18
<b>5. Discussion et analyse</b>	<b>21</b>
<b>6. Mise en oeuvre du projet</b>	<b>23</b>
<b>7. Conclusion</b>	<b>25</b>
<b>Bibliographie</b>	<b>26</b>



## Abstract

De nos jours, le trafic maritime augmente rapidement. Plus de navires implique une augmentation des risques d'infractions en mer comme les accidents de navires dévastateurs pour l'environnement, la piraterie, la pêche illégale, le trafic de drogue et le mouvement illégal de cargaisons. Cela a obligé de nombreuses organisations, des agences de protection de l'environnement aux compagnies d'assurances et aux autorités gouvernementales nationales, à surveiller de plus près les mers ouvertes.

Airbus propose des services complets de surveillance maritime en créant une solution pour une large couverture, une surveillance intensive et détaillée, une réactivité premium et une réponse d'interprétation. En combinant ses données propriétaires avec des analystes hautement qualifiés, Airbus aide à soutenir l'industrie maritime pour accroître les connaissances, anticiper les menaces, déclencher des alertes et améliorer l'efficacité de la surveillance en mer.

Beaucoup de travail a été fait au cours des 10 dernières années pour extraire automatiquement des objets à partir d'images satellites avec des résultats significatifs mais sans effets opérationnels effectifs. Désormais, Airbus se tourne vers nous pour augmenter la précision et la rapidité de la détection automatique de navires.

Ainsi est lancé Shipmark, un projet de la part d'Airbus Defence and Space qui vise à détecter les bateaux en mer pour identifier leur position et leur direction. En utilisant la technologie du Machine Learning, cet algorithme sera entraîné grâce à un dataset d'environ 300 000 images satellites fournies par Airbus.

Nous avons créé pour cela des algorithmes basés sur le modèle de détection d'objet Yolov7 ainsi que le modèle de détection de keypoints Centernet. Après avoir entraîné les différents algorithmes, ces derniers sont aujourd'hui en mesure de détecter plus de 90% des bateaux et de placer correctement plus de 80% des keypoint.

Shipping traffic is growing fast. More ships increase the chances of infractions at sea like environmentally devastating ship accidents, piracy, illegal fishing, drug trafficking, and illegal cargo movement. This has compelled many organizations, from environmental protection agencies to insurance companies and national government authorities, to have a closer watch over the open seas.

Airbus offers comprehensive maritime monitoring services by building a meaningful solution for wide coverage, fine details, intensive monitoring, premium reactivity and interpretation response. Combining its proprietary-data with highly-trained analysts, they help to support the maritime industry to increase knowledge, anticipate threats, trigger alerts, and improve efficiency at sea.



A lot of work has been done over the last 10 years to automatically extract objects from satellite images with significant results but no effective operational effects. Now Airbus is turning to us to increase the accuracy and speed of automatic ship detection.

The Shipmark project was initially a proposal from Airbus Defence and Space. In order to complete its former "Airbus Ship Detection Challenge" competition, this project aims to detect ships at sea to identify their position and direction. Using Machine Learning technology, this algorithm will be trained using a dataset of around 300,000 satellite images provided by Airbus.

For this, we have created algorithms based on the Yolov7 object detection model as well as the Centernet keypoint detection model. After training the different algorithms, they are now able to detect more than 90% of boats and correctly place more than 80% of keypoints.



# Remerciements

Nous souhaitons remercier quelques personnes qui nous ont grandement aidé lors de ce projet, notamment Denis Marraud qui nous a guidés tout le long de ce projet ainsi qu'Arthur Desjardins qui l'a assisté sur la partie technique. Nous voulions aussi remercier Geoffroy Lardeux qui nous a accompagnés notamment sur l'organisation et les rendus pour l'ECE. Nous tenions aussi à remercier Prabhakar Sivanesan pour la mise à disposition de son travail sur tensorflow object detection API qui nous a grandement aidé.

## 1. Exigences / Opportunités

Le projet Shipmark vise à permettre l'identification de landmarks (points d'intérêts) de bateaux présents sur des images satellites. En effet, avec la multiplication des capacités d'acquisition d'images satellitaires et à l'inverse les limitations sur les ressources disponibles en matière d'analystes images, il est crucial de disposer de traitements automatiques d'analyse d'images afin de focaliser l'attention des analystes sur les zones et images d'intérêt. Une meilleure caractérisation des bateaux présents sur une zone donnée contribue à cet objectif.

C'est avec l'objectif, dans un premier temps, de mettre en avant les images contenant des bateaux que le projet est lancé lors du concours "Airbus Ship Detection Challenge" par Airbus il y a maintenant 4 ans. À la suite de ce concours, Airbus possédait un premier algorithme de deep learning capable de détecter les bateaux présents dans une image, mais ils n'étaient pas pleinement satisfaits du résultat. En effet, le concours a permis de créer un algorithme de deep learning qui reconnaît des zones d'intérêt sur une image, mais ne donne pas plus d'information. Cependant, avoir la possibilité de déterminer des caractéristiques précises des bateaux est très bénéfique, permettant de mieux appréhender le but du bateau à l'endroit et au moment précis où a été prise la photo.

Airbus souhaite aujourd'hui ajouter une nouvelle fonctionnalité telle que l'extraction, sous forme de point d'intérêt pour chaque bateau reconnu dans l'image, de sa proue, poupe tribord et poupe bâbord. Pour implémenter cette fonction, nous devrons utiliser un algorithme de deep learning que nous aurons choisi, configuré et implanté en python. Les images satellites qui serviront à l'entraînement de notre IA seront fournies par Airbus, mais nécessiteront un tri au préalable afin d'identifier les images avec de l'intérêt ou non. L'objectif suivant est aussi de déduire les trois points d'intérêt des bateaux même si toutes les parties des bateaux ne sont pas visibles. Toutes les images qui seront transmises à notre algorithme ne seront pas toujours parfaites. Parfois, il n'y aura pas de bateaux dans l'image ou alors, ils seront cachés par des nuages ou seront coupés par les bords de l'image. Il se peut également que la qualité de l'image ou la taille du bateau sur l'image le rende difficile à identifier. Nous devons donc prendre en compte ces éventualités pour rendre notre système le plus performant possible.



L'identification des 3 points d'intérêts permet d'isoler différents aspects du bateau comme sa taille et de pouvoir en déduire à quel type de bateau nous avons à faire. De plus, cela permettrait de déterminer l'orientation du bateau, et donc de déterminer son sens de navigation. Ces fonctionnalités auraient alors plusieurs applications, à la fois civiles et militaires, telles que de combattre la pêche illégale pour la préservation des espèces, ou encore le dégazage / déballastage d'hydrocarbures dans la protection de l'environnement ou encore de mieux appréhender les enjeux géopolitiques de certaines zones comme par exemple le cas du blocus d'un port.

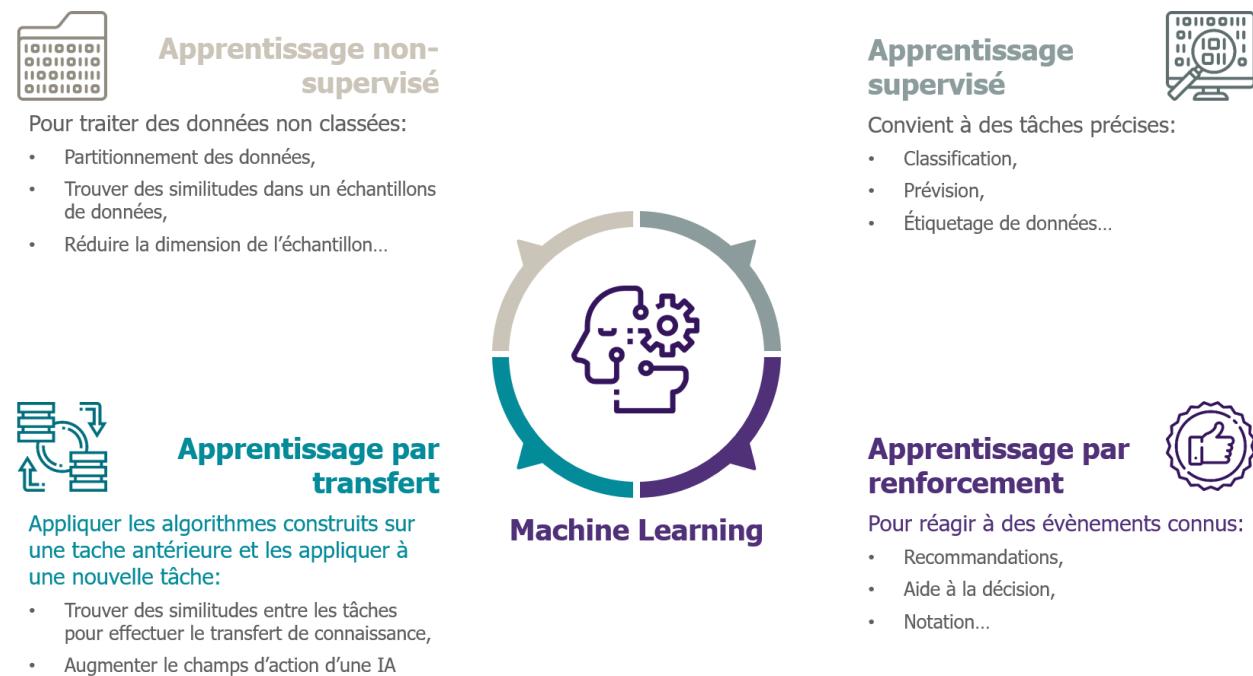
Les attendus d'Airbus pour le projet sont :

- Un dataset trié qui permettra de correctement entraîner notre algorithme et de le tester. Les datasets (ou jeux de données) regroupent un ensemble de données cohérentes qui peuvent se présenter sous différents formats (textes, chiffres, images, vidéos, etc...). Ils sont utilisés pour entraîner des algorithmes de machine learning. Le but est de correctement entraîner l'IA pour optimiser au maximum ses performances.
- Un algorithme de deep learning écrit en python avec les scripts de test et d'apprentissage. L'algorithme devra implémenter les fonctionnalités décrites dans le paragraphe précédent.
- Une présentation de la méthode retenue, du dataset, des performances obtenues.



## 2. État scientifique & technique

Une fois que les objectifs et les buts de notre projet ont bien été définis, nous avons commencé nos recherches sur la manière d'aborder le problème. Pour répondre à notre problématique, nous devons utiliser une intelligence artificielle car la reconnaissance doit être automatique sur des photos satellites. L'intelligence artificielle désigne un ensemble de théories et de techniques réunies dans des programmes informatiques complexes capables de simuler certains traits de l'intelligence humaine (raisonnement, apprentissage...). Nos recherches ont donc débuté sur le type d'intelligence que nous allons utiliser. Étant donné que nous devons reconnaître et placer des points sur des bateaux, cela signifie que nous connaissons déjà notre sortie et notre entrée. En entrée, nous aurons une image satellite et en sortie la même image avec des points d'intérêts placée sur le bateau. Nous aurons donc besoin d'entraîner une intelligence artificielle à reconnaître ces points d'intérêt grâce aux machines learning. Parmi toutes les techniques qui existent dans le machine learning, nous utiliserons l'apprentissage supervisé, car c'est cette technique qui convient le mieux aux travaux d'étiquetage. Il existe déjà aujourd'hui des intelligences artificielles qui savent reconnaître des points d'intérêts sur une image. La bibliothèque OPENCV fournit par exemple un outil qui permet de reconnaître les points d'intérêt sur un visage tels que les yeux, la bouche, le nez... Nous pourrions nous inspirer de son fonctionnement pour l'appliquer à notre objectif.

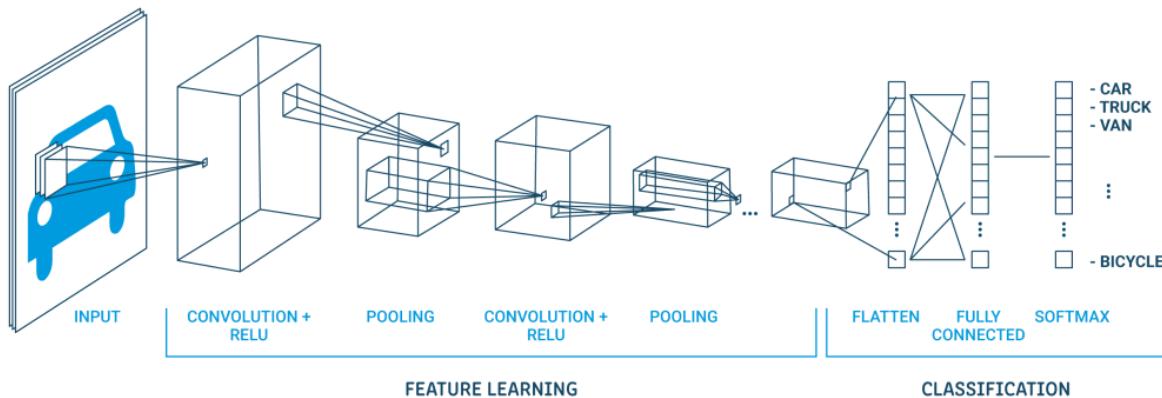


## a) Réseaux neuronaux convolutifs profonds

La force des DCNN (Deep Convolutional Neural Network = réseaux neuronaux convolutifs profonds) réside dans leur superposition. Un DCNN utilise un réseau neuronal tridimensionnel pour traiter les éléments rouge, vert et bleu de l'image en même temps. Cela réduit considérablement le nombre de neurones artificiels nécessaires au traitement d'une image, par rapport aux réseaux neuronaux à action directe traditionnels.

Les réseaux neuronaux convolutifs profonds reçoivent des images en entrée et les utilisent pour former un classificateur. Le réseau utilise une opération mathématique spéciale appelée "convolution" au lieu de la multiplication matricielle.

L'architecture d'un réseau convolutionnel se compose généralement de quatre types de couches : convolution, pooling, activation et entièrement connectée.



## b) Masque RCNN

Il y a néanmoins certains projets qui ont un but similaire au nôtre, mais qui sont appliqués dans un autre domaine que la surveillance de bateau en mer. Nous avons une problématique semblable à ce projet. L'outil le plus important utilisé ici est le masque RCNN.



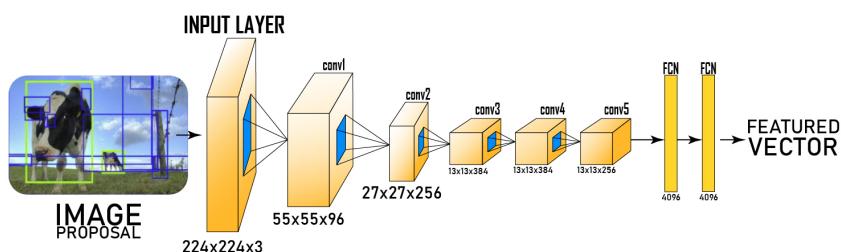
Le masque RCNN est un réseau neuronal visant à résoudre le problème de la segmentation des instances dans le machine learning ou computer vision. En d'autres termes, il peut



séparer différents objets dans une image ou une vidéo. Vous lui donnez une image, il vous donne les boîtes de délimitation des objets, les classes et les masques.

Le masque RCNN comporte deux étapes. Premièrement, il génère des propositions sur les régions où il pourrait y avoir un objet en se basant sur l'image d'entrée. Ensuite, il prédit la classe de l'objet, affine la boîte englobante et génère un masque au niveau du pixel de l'objet sur la base de la proposition de la première étape. Les deux étapes sont connectées à la structure dorsale.

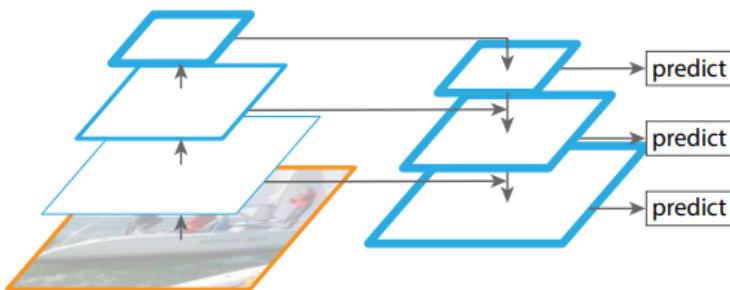
La détection d'objets à différentes échelles est un défi, en particulier pour les petits objets. Nous pouvons utiliser une pyramide de la même image à différentes échelles pour détecter les objets. Cependant, le traitement d'images à plusieurs échelles prend du temps et la demande de mémoire est trop élevée pour être entraînée. Alternativement, nous créons une pyramide de caractéristiques et les utilisons pour la détection d'objets.



Un réseau pyramidal de caractéristiques (Feature Pyramid Network, ou FPN) est un extracteur de caractéristiques qui prend en entrée une image à échelle unique de taille arbitraire et produit des cartes de caractéristiques de taille

proportionnelle à plusieurs niveaux. Il s'agit donc d'une solution générique pour construire des pyramides de caractéristiques à l'intérieur de réseaux convolutifs profonds, à utiliser dans des tâches telles que la détection d'objets.

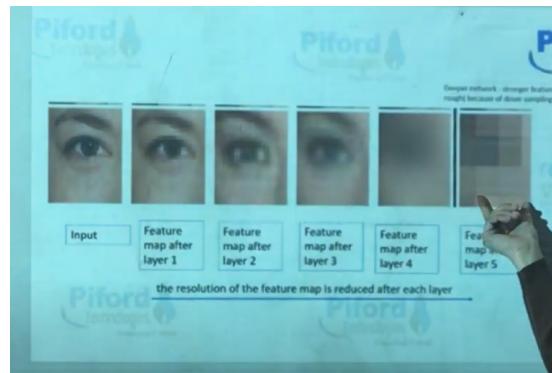
La construction de la pyramide implique un cheminement ascendant et un cheminement descendant.



Le cheminement ascendant (the bottom-up pathway) vient appliquer des filtres sur l'image d'entrée pour réduire sa taille en pixel. L'image va se dégrader au fil des couches, la résolution sera moins grande jusqu'à avoir une image de basse résolution sur la dernière couche.

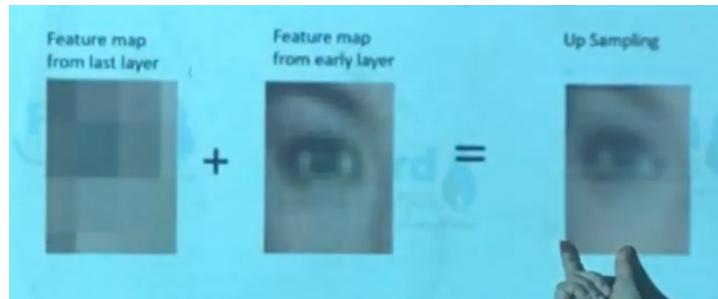
Cette image de faible résolution de la dernière couche ne donne pas de bons résultats lorsqu'on vient l'utiliser pour la reconnaissance d'objet. Or c'est une image importante qui contient beaucoup d'informations. Exemple :





Ensuite, nous appliquons le cheminement descendant (the top-down pathway). Dans ce procédé, nous allons fusionner les couches de basse résolution avec les couches de haute résolution pour obtenir un meilleur résultat lors de la reconnaissance d'objet.

Exemple :



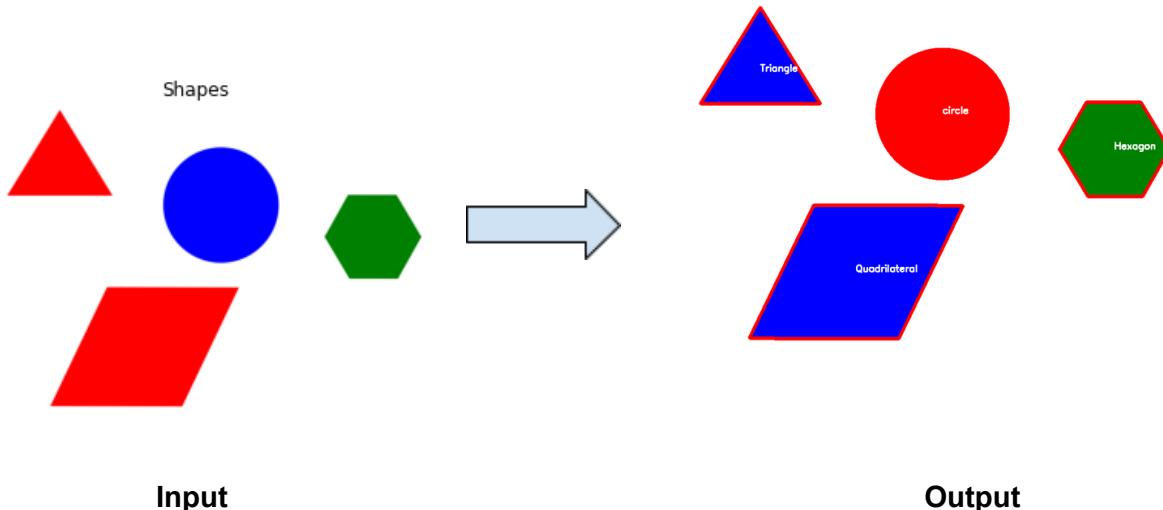
Examinons maintenant la première étape. Un réseau neuronal léger appelé RPN analyse toutes les voies haut-bas des FPN et propose des régions susceptibles de contenir des objets. Ces régions sont entourées par des rectangles.

Lors de la deuxième étape, un autre réseau neuronal prend les régions proposées par la première étape et les attribue à plusieurs zones spécifiques d'un niveau du FPN, analyse ces zones et génère des classes d'objets, des boîtes de délimitation et des masques.

Une autre solution à laquelle nous pourrions penser est la library OpenCV pour la détection d'objets. OpenCV est une bibliothèque open source utilisée principalement pour le traitement des images et des vidéos afin d'identifier des formes, des objets, du texte, etc. Elle est surtout utilisée avec Python.

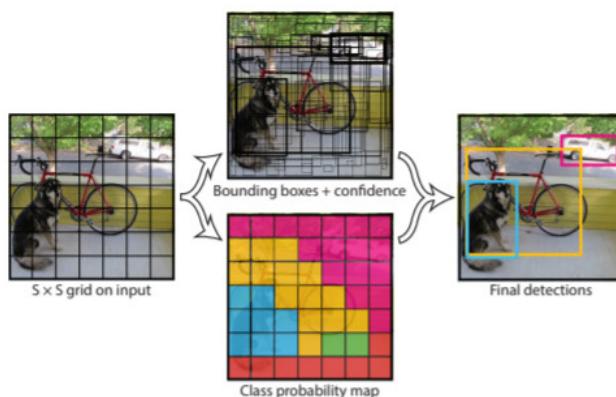
Pour ce faire, nous avons deux fonctions qui sont particulièrement intéressantes. Cv2.findContours() : cette méthode permet de trouver tous les points limites d'une forme dans une image, et cv2.drawContours() : cette méthode dessine un contour. Elle peut également dessiner une forme si vous fournissez des points limites.



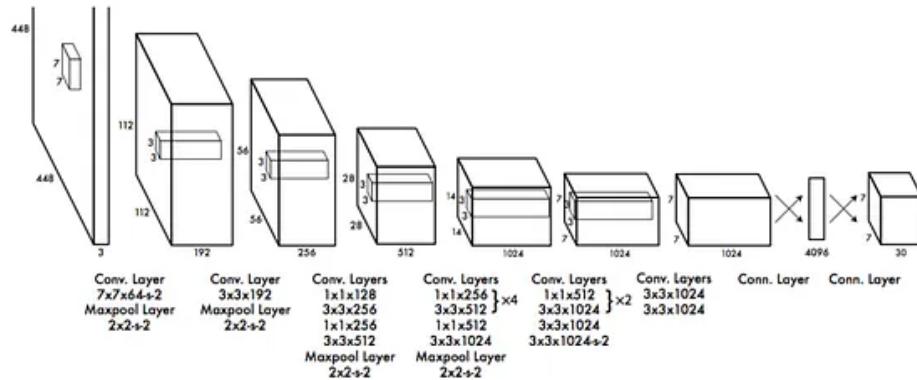


### c) YOLO (You Only Look Once)

YOLO est un réseau de neurones assez connu pour la détection d'objets, notamment pour sa fluidité d'exécution. En effet, cet algorithme, une fois entraîné, permet de faire de la détection d'objets en temps réel. Comparé à ce qui a été présenté ci-dessus est la façon d'aborder le problème à un niveau élevé. Plutôt que de considérer le problème comme étant purement un problème graphique (détection sur des images), YOLO tend à réduire la détection d'objet à un “simple problème de régression”. Un second élément différenciant entre YOLO et la technologie RCNN (keypoints-RCNN, Masque RCNN, Fast-RCNN) est dans la façon de conceptualiser une image. En effet, plutôt que de considérer l'image comme des zones et de la traiter grâce à une fenêtre glissante, on considère ici l'image dans son ensemble, permettant notamment de réduire les faux positifs dans l'arrière-plan des images. Enfin, un autre avantage de cet algorithme est sa flexibilité qui lui permet de détecter assez précisément des objets sur des photos différentes de celles sur lesquelles il a été entraîné. YOLO fonctionne en divisant l'image en une grille de taille  $S \times S$ . Des bounding box sont ensuite créées et un indice de confiance est aussi généré pour chaque box. Une carte des classes est aussi générée. En combinant les informations générées, on obtient une détection d'objets (cf photo ci-dessous).



D'un point de vue architectural, YOLO utilise un réseau de neurones à convolution profonde, à savoir 24 couches de convolution suivies de 2 couches complètement connectées. Le problème principal pour notre application est que YOLO peut avoir du mal à détecter plusieurs objets accolés.



## d) COCO

COCO signifie "Common object in context", littéralement en français "des objets communs dans un contexte". Bien qu'il soit possible de l'implémenter pour différents objets, COCO est à la base conçu pour la segmentation d'objets et la détection d'objets en contexte, et principalement du corps humain.

## e) ResNet 50

ResNet signifie "Residual Network" et est un type spécifique de réseau de neurones à convolution profonde introduit en 2015. Il est composé de 50 couches de neurones : 48 couches de convolution, une couche de MaxPool et enfin une couche d'average pool.

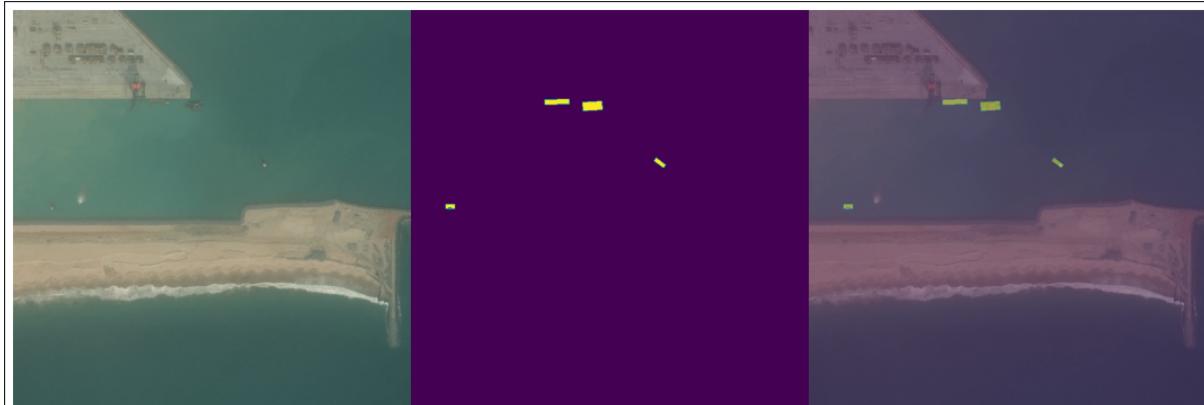
## f) CenterNet

CenterNet est une méthode d'extraction de keypoints sur des images plus précises que les méthodes habituelles. Comme la majorité des algorithmes, il a été conçu pour déterminer des keypoints sur le corps humain. Il permet également, à partir de la détermination de keypoints, d'extraire la bounding box d'un objet. CenterNet représente chaque objet comme un point unique et vient ensuite réaliser une régression depuis les features de l'image. Lorsqu'on lui transmet une image, CenterNet transforme l'image en une heatmap dont les pics de chaleur correspondent au centre d'un objet. Une autre particularité de CenterNet est sa rapidité d'exécution.



### 3. Solution proposée (valeur ajoutée)

Pour commencer ce projet, Airbus nous a donné un grand nombre d'images satellites, certaines contenant des bateaux, certaines non, avec ce set d'images, nous avons eu un excel résultant des recherches du *airbus ship detection challenge*. Grâce à cela, nous avons pu isoler seulement les images contenant des bateaux. Nous avons donc eu 42 000 images satellites contenant des bateaux.



*Une image avec les informations que l'on a reçues de la part d'Airbus*

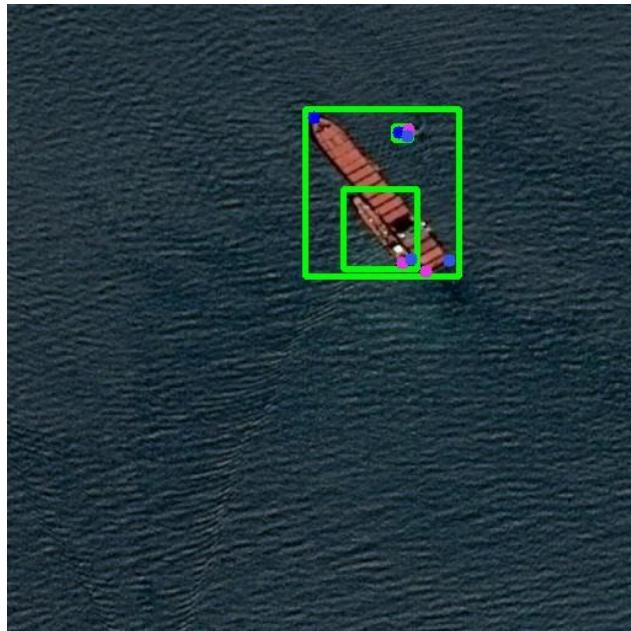
Le but du projet était de pouvoir déterminer la direction grâce à la position de l'avant et l'arrière des bateaux sur les images satellites. Pour ce faire, nous avons besoin de 3 points, la proue, la poupe arrière gauche et droite. Cela permet de calculer la longueur ainsi que la largeur du bateau à l'arrière. On peut aussi trouver la direction du bateau avec ces 3 points. Cela devra être fait par un algorithme de deep-learning, celui-ci devra alors identifier les bateaux et leurs keypoints (points d'intérêts avant arrière). L'algorithme de deep-learning a été entraîné avec un grand nombre d'images satellites annotées.



*Image simple en sortie de l'algorithme de deep-learning*



Il devra aussi être capable d'opérer lorsque deux bateaux sont accolés ou lorsqu'ils sont situés dans un port. Aussi, il faudrait que l'on puisse savoir si certains des points ne sont pas visibles sur l'image, car le bateau est au bord de l'image.



*Image de deux bateaux accolés en sortie de l'algorithme de deep-learning*

Un des facteurs différenciant du projet est la reconnaissance de points sur des bateaux, en effet, il existe un grand nombre d'algorithmes de reconnaissance d'objet qui trouvent des rectangles au sein desquels se trouve différents objets. Dans notre cas, nous cherchons des points précis sur une image, il existe peu d'algorithmes capables de placer des points sur une photo, la plupart de ceux qui existent servent à reconnaître le corps humain ou des parties de celui-ci.



## 4. Résultats

Pour ce projet de computer vision, il nous semblait approprié de baser notre solution sur un algorithme de deep learning, utilisant donc un réseau de neurones artificiel pour la reconnaissance. Afin d'entraîner ce dernier, nous avions besoin d'une grande quantité de données, dans notre cas, des images satellites contenant des bateaux ainsi que les coordonnées de leur avant et arrière (proue et poupe).

L'annotation des images est cruciale pour une IA qui doit reconnaître des points dans des images. Lors de l'annotation, des informations supplémentaires sont ajoutées aux images pour les aider à comprendre le contenu et la structure des images. Cela permet à l'IA d'apprendre les caractéristiques clés des points à reconnaître, telles que leur forme, leur taille, leur position et leur orientation. Plus il y a d'images annotées, plus l'IA aura de données pour s'entraîner et améliorer sa capacité à reconnaître les points de manière précise. De plus, une variété d'images annotées avec différentes perspectives, couleurs et bruit aide l'IA à mieux généraliser et à être plus robuste face à des variations dans les images réelles. En somme, l'annotation de nombreuses images est cruciale pour garantir la précision et la fiabilité de l'IA dans la reconnaissance des points dans des images. C'est dans ce but que nous avons cherché à automatiser les annotations pour fournir rapidement une grande quantité d'images pour l'entraînement de l'intelligence artificielle. Le but est donc d'annoter dans un premier temps des images manuellement afin de fournir un premier jeu de données à l'IA afin de vérifier que les entraînements se passent bien. Dans un second temps, nous voulons associer les images annotées manuellement avec des images annotées automatiquement une fois que notre algorithme d'annotation automatique sera au point.

### A. Annotation manuelle

L'annotation manuelle est l'étape la plus chronophage du projet. Il a fallu annoter un maximum d'images afin de les fournir à notre réseau de neurones artificiels. Nous avons dû faire face à plusieurs problèmes lors de cette étape d'annotation.

Le problème important est la manière d'annoter. En effet, il n'est pas toujours évident de savoir où placer exactement chaque point sur les images. Certains bateaux sont si petits qu'il est très compliqué de savoir où se trouvent la proue et la poupe. Des bateaux sont parfois coupés par les bords des images ou cachés par des nuages. Il nous a fallu du temps pour savoir comment annoter ces images pour, malgré tout, entraîner l'IA dessus.

Le deuxième problème que nous avons rencontré portait sur le format des annotations. Nous avons utilisé plusieurs outils comme Roboflow ou Coco Annotator. Le problème était que lorsque nous avons terminé les annotations, il faut les exporter. Cela se présente sous la forme d'un fichier ".json" contenant toutes les informations d'annotation pour coco annotator. Ce fichier peut ensuite être donné au modèle de machine learning pour qu'il puisse s'entraîner dessus. Le problème étant que certains modèles d'IA ne supportent pas le format d'annotation Coco ou ne supportent pas les annotations de Roboflow. Nous avons

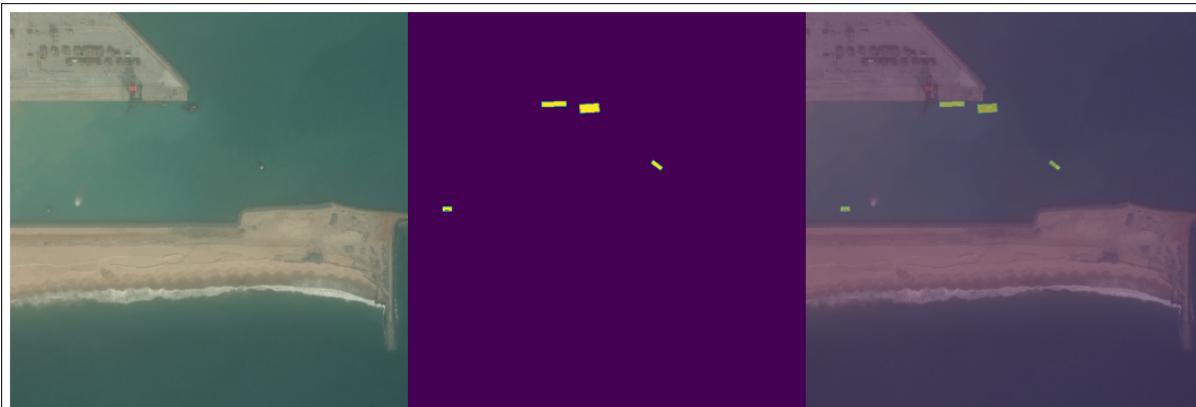


donc dû annoter plusieurs fois les mêmes images sur les deux outils pour tester différents algorithmes.

## B. Annotation automatique

Prenant la suite d'un projet existant, nous devions prendre en entrée les données de sortie de l'algorithme du *Airbus ship detection challenge*. C'est-à-dire un grand nombre d'images satellites de l'océan, ainsi qu'un document Excel nous donnant pour chaque image, le nombre de bateaux et leur position.

Nous pouvons voir sur l'image ci-dessous, à gauche l'image satellite d'origine sur laquelle on recherche la présence de bateaux, au centre les rectangles formés par les coordonnées extraites du document Excel (on remarque en jaune les bateaux détectés) et à droite la superposition des deux.



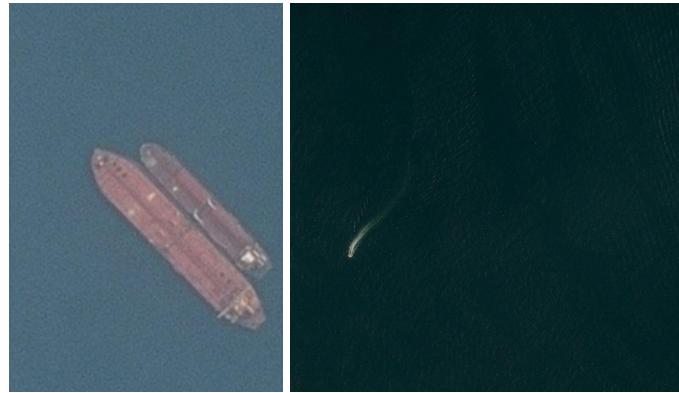
Nous cherchons à extraire la position de la proue ainsi que la poupe tribord et bâbord. Afin d'entraîner le réseau de neurones, nous avons besoin d'un grand nombre d'images avec les coordonnées de ces trois points pour chaque bateau.

Disposant de 42 000 images contenant des bateaux, il était inenvisageable de toutes les annoter à la main, nous avons cherché une manière de les annoter automatiquement.

Grâce aux données de l'Excel, il nous est possible de prendre seulement les images contenant des bateaux afin de les annoter. De plus, cela nous permet d'estimer plus facilement la direction des bateaux. En effet, avec un rectangle, l'avant est forcément sur un des côtés les plus courts, ce qui ne laisse que deux possibilités. L'avant est donc au milieu d'un des côtés les plus courts, là où la poupe tribord et bâbord sont dans les coins opposés.

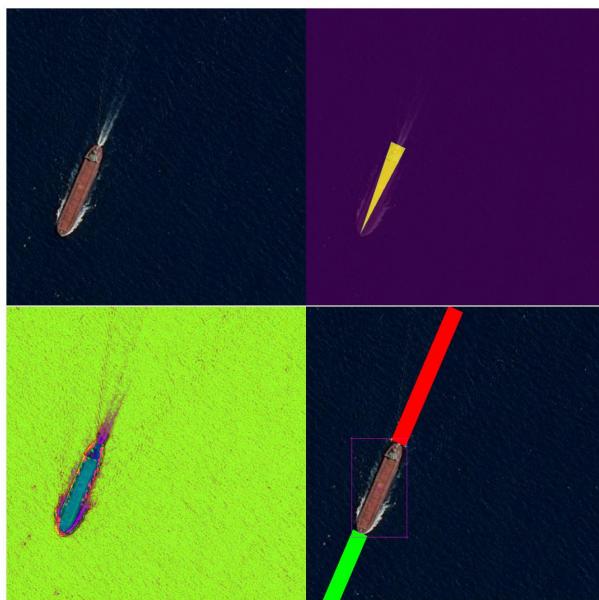
Afin de déterminer le sens du bateau, il a donc fallu réfléchir à un paramètre significatif visible, notre première idée fut de regarder la forme des bateaux, l'avant étant en général plus pointu, cependant un très grand nombre de bateaux nous semblait bien trop petit pour utiliser ce paramètre comme on peut le voir avec l'exemple ci-dessous.





*à gauche une image sur laquelle ce paramètre fonctionnerait, à droite un bateau trop petit pour analyser sa forme.*

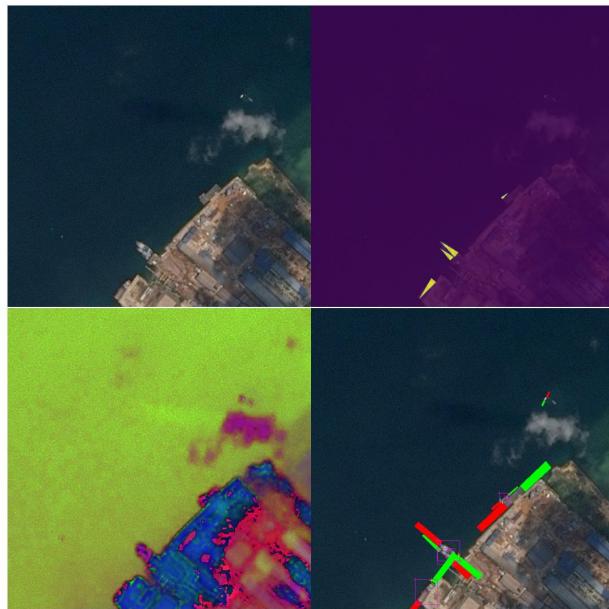
Notre autre idée pour trouver le sens des bateaux fut donc de regarder la traînée qui se situe forcément à l'arrière du bateau, cela ne permet pas non plus de connaître le sens de tous les bateaux, mais cela a pour mérite de s'appliquer aux bateaux de toutes les tailles.



Afin de voir plus significativement la différence de couleur entre la mer et la trainée, nous faisons une transformation de couleur comme on peut le voir en bas à gauche dans l'image ci-dessous. Nous comparons ensuite la couleur moyenne des deux côtés du bateau (dans la zone rouge et la zone verte). Afin de limiter les erreurs, nous ne prenons que les bateaux pour lesquels la différence de couleur est significative.

Aussi, nous avons observé que nous avions des problèmes pour déterminer le sens des bateaux se situant au bord de la terre, en effet la couleur des ports et de la terre est souvent confondue avec celle de la traînée.





Pour cette raison, nous n'annotons pas automatiquement les images contenant trop de terre ou de nuages. Cela nous laisse toujours 25% des images qui peuvent être annotées automatiquement, cela peut paraître peu, mais ayant 42 000 images cela nous donnerait déjà 10 000 images. Sur les images que nous arrivons à annoter automatiquement, nous obtenons une précision de 78% sur l'échantillon de 500 images que nous avions aussi annotées à la main. Cela nous permettra déjà d'avoir un nombre important d'images pour l'apprentissage de notre réseau de neurones.

## C. Les modèles de machine learning

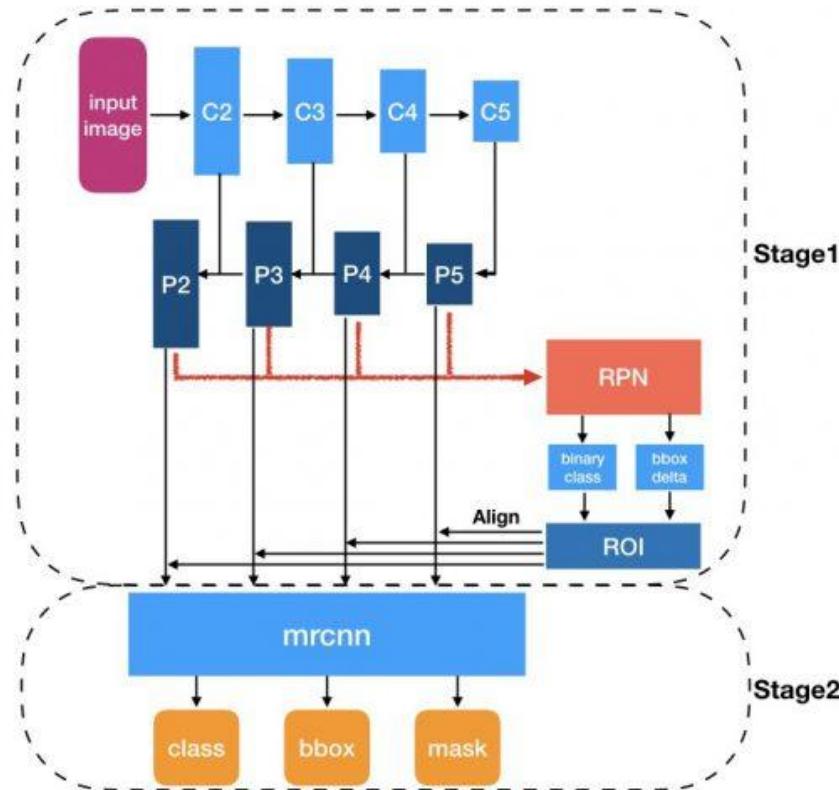
Pour entraîner nos premiers modèles et n'ayant pas encore d'images annotées automatiquement, nous avons dû commencer par annoter des images manuellement. Nous avons utilisé dans un premier temps le logiciel labelme, par la suite, nous nous sommes tournées vers le site roboflow qui nous permet, entre autres, d'importer des images, de se les répartir, de les annoter, mais aussi d'entraîner des réseaux de neurones pour la reconnaissance d'objet.

Ensuite, nous avons créé notre réseau de neurones en utilisant l'algorithme de keypoint RCNN, ce dernier est basé sur le Mask\_RCNN.

Le Mask R-CNN est une architecture qui fonctionne en trois grandes parties. On a tout d'abord un réseau convolutif appelé backbone qui extrait des primitives à partir de nos images de départ. À partir de ces primitives, une seconde partie (le Region Proposal Network ou RPN) va proposer et affiner un certain nombre de régions d'intérêts (sous la forme de boîtes englobantes rectangulaires) susceptibles de contenir un objectif. Enfin, la dernière partie va récupérer les meilleures propositions, les affiner de nouveau, et produire un masque de segmentation propre à chacune d'entre elles.



Un schéma récapitulatif du réseau se trouve ci-dessous. Il est à noter qu'au total, ce réseau possède une centaine de couches de convolution. Cela rend sa manipulation plus complexe, car il est plus difficile d'interpréter les résultats obtenus.

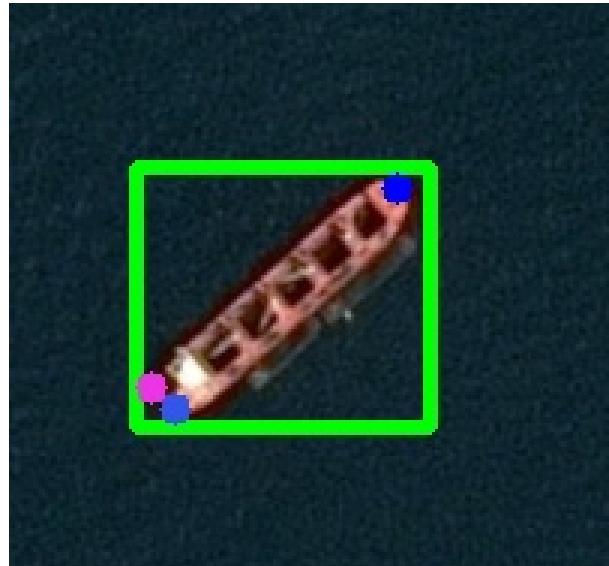


Le Mask RCNN est très efficace pour la détection de keypoints sur le corps humain comme les articulations, mais ne fonctionne pas très bien dans notre cas sur des bateaux. Nous avons testé plusieurs codes et plusieurs paramètres pour Resnet50 mais sans obtenir de résultat satisfaisant notamment au niveau du test de l'overfitting qui donne un indice sur l'apprentissage du réseau de neurone. Nous sommes alors passés sur un autre code basé sur Centernet et COCO qui est un algorithme de détection de keypoints dont le dataset d'image n'est pas fixe.

Centernet est un modèle de détection d'objets développé par Facebook AI. Il est utilisé pour trouver des points clés sur des images. Il prédit la position et la confiance de chaque point clé en utilisant un réseau neuronal. Centernet est entraîné avec des images annotées pour améliorer ses prédictions. Il a montré de bonnes performances dans la reconnaissance de formes dans des images et des vidéos.

Nous pouvons alors fournir à cet algorithme notre dataset de bateau et l'entraîner dessus. Les résultats de COCO sont tout de suite bien plus satisfaisants (plus de détails dans la partie 5 - Discussion et Analyse).





YOLO quant à lui, est spécialisé sur la détection d'objet. Nous avons utilisé cet algorithme pour notre projet pour détecter la proue et la poupe gauche et droite, mais ce n'était pas exactement ce qu'attendait AIRBUS. L'attendu était des points précis pour pouvoir effectuer des mesures comme la taille des bateaux. Les résultats de YOLO étaient bons pour environ la moitié des photos, ce qui n'est pas satisfaisant, néanmoins notre contact chez AIRBUS nous a demandé de continuer à explorer cette piste pour pouvoir comparer les différents outils.

Afin de disposer de plus de ressources pour faire fonctionner notre algorithme plus rapidement et de pouvoir partager les résultats entre nous, nous avons exécuté le code sur Google collab qui est un service de Google qui permet d'exécuter du code en python dans une machine virtuelle hébergée sur le cloud. Afin d'avoir accès à plus de ressources, nous avons pris la version payante collab pro + qui nous permet d'exécuter le code sur une carte graphique et d'accélérer grandement l'apprentissage.



## 5. Discussion et analyse

Comme nous l'avons vu plus tôt, nous avons utilisé le réseau CenterNet pour la détection des keypoints. À la fin de cette dernière semaine de sprint, le meilleur modèle détecte seulement 66% des bateaux présents sur les images, en effet de nombreux bateaux sont à peine visibles, comme on peut le voir sur l'image suivante. Ici seul le bateau à gauche serait détecté.



On voit aussi que l'on est capable de localiser correctement presque 90% des avant de bateau, ce qui est un bon score. En ce qui concerne les points arrière, nous obtenons 82% de point détecté au bon endroit, cela est plus complexe, car les deux points arrière sont très similaires, ils sont donc compliqués à différencier, pour savoir quel est le point arrière gauche ou droite, il faut s'orienter par rapport au bateau ce qui rend cela plus complexe.

La précision moyenne, pour trouver chaque point, est d'environ 1.9 pixels en X et 1.9 pixels en Y. En sachant que les images ne sont pas prises par le même satellite. Un pixel vaut en moyenne 80 cm. Si nous connaissons le satellite avec lequel les images ont été prises nous aurions une idée assez précise de la taille des bateaux, cependant cela reste assez peu précis pour les bateaux de petite taille.

En ce qui concerne le fait de trouver le sens des bateaux, nous obtenons un très bon score de 90%, cela veut dire que nous serions capables de trouver la direction des bateaux quasiment à tous les coups.

Quelques axes d'améliorations seraient d'améliorer la détection des points arrière ainsi que la précision, pour cela il nous faudrait un plus grand dataset d'images annotées en effet l'annotation automatique nous permet certes, d'avoir un grand nombre d'images annotées mais certaines de ces annotations ne sont pas correctes et limitent le pouvoir d'apprentissage du modèle. Il faudrait donc perfectionner ce système d'annotation automatique. Par la suite, nous pourrions également entraîner plus longtemps notre



algorithme. Pour les besoins du projet nous avons utilisé Google Colab. C'est une plateforme qui nous a permis d'entraîner l'ia, mais nous avons été limités dans le temps et par la puissance de calcul. La combinaison de ces facteurs ne nous a pas permis d'obtenir des résultats optimaux.

Voici un comparatif des résultats obtenu avec le temps:

	Modèle version 1	Modèle version 2	Modèle version 3	
Part des bateaux détectés dans les images	54%	65%	66%	Le plus haut est le mieux
Parmi les bateaux détectés, quelle est la propension de bateau dans les bons sens	90%	91%	90%	Le plus haut est le mieux
Précision moyenne des keypoints sur l'axe X	4,4 pixels	1,9 pixels	1,9 pixels	Le plus bas est le mieux
Précision moyenne des keypoints sur l'axe Y	3,5 pixels	1,9 pixels	2,3 pixels	Le plus bas est le mieux

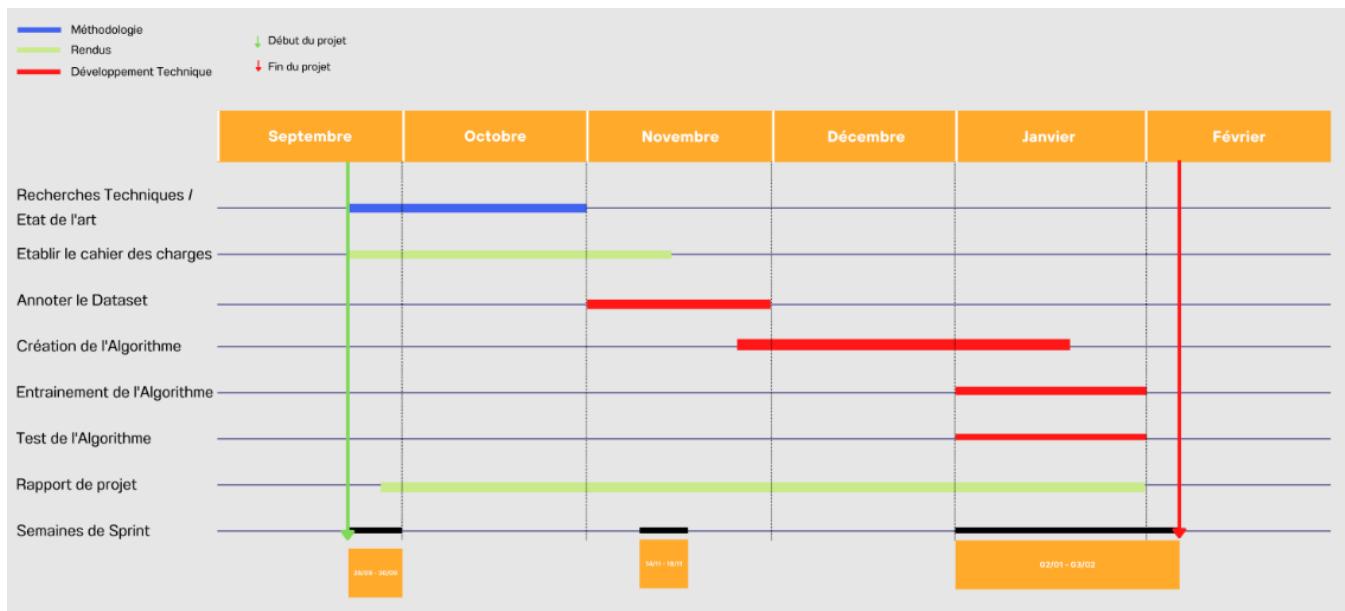
Nous pouvons voir que les deux dernières versions sont bien meilleures que la première version. Cela est dû au fait que nous avons beaucoup augmenté la taille du dataset. Avec les dernières versions, nous remplissons notre objectif qui était de détecter des keypoints sur des bateaux à partir d'images satellites.



## 6. Mise en oeuvre du projet

Rôle	Responsabilités	Membre de l'équipe
Data management	Algorithme d'annotation automatique / Algorithme de mesure des résultats	Clément Bouvard
Data management	Algorithme d'annotation automatique / Algorithme de mesure des résultats	Edouard Mignien
Team neural network	Recherche et développement d'algorithme de détection de keypoints	Quentin Maisonnier
Team neural network	Recherche et développement d'algorithme de détection de keypoints	Mathias Sericola
Team neural network	Recherche et développement d'algorithme de détection de keypoints	Pierre Langlois
Team neural network	Recherche et développement d'algorithme de détection de keypoints	Ibraguim Vadouev





## Planning détaillé du mois de janvier

Du 02/01 au 06/01 :

- Développement d'un algorithme de mesure des résultats
- Recherche d'algorithme de détection de keypoints sur des bateaux

Du 09/01 au 13/01 :

- Entraînement du modèle RCNN (1er modèle) (Keypoints detection)
- Mesure des résultats du modèle RCNN

Du 16/01 au 20/01 :

- Entraînement du modèle YOLO V7 (2ème modèle) (object detection)
- Mesure des résultats du modèle YOLO V7

Du 23/01 au 27/01 :

- Entraînement du modèle Center Net (3ème modèle) (Keypoints detection)
- Mesure des résultats du modèle Center Net



## 7. Conclusion

Le projet Shipmark est un projet dans un domaine totalement nouveau pour une grande partie du groupe. Ce projet était une opportunité de découvrir le secteur du machine learning et de la computer vision. Nous avons découvert plusieurs outils de détection d'objets et de points d'intérêts comme YOLO avec Resnet50 et COCO avec Centernet. Nous avons passé beaucoup de temps à chercher l'outil le plus efficace pour nos attendus et nous avons donc retenu COCO et Resnet50. L'annotation des images était aussi une partie problématique, car extrêmement chronophage, c'est pour cette raison que nous avons développé un algorithme d'annotation automatique. Pour valider notre algorithme de détection, il nous fallait une confirmation de la précision des points trouvés. L'algorithme de validation répond à cela, nous obtenons une moyenne et un écart type sur la précision des points en pixels, les pixels représentent 50 centimètres ou 1 mètre et 20 centimètres selon le satellite de provenance de la photo.

Les suites imaginables du projet sont l'utilisation des données des keypoints afin de déterminer avec précision la direction des bateaux. On pourrait de plus, sous réserve de savoir de quel satellite provient la photo, déterminer la taille du bateau en faisant correspondre un nombre de pixels avec la taille réelle qu'ils représentent. Une fonctionnalité possible est aussi la détermination de la taille, du type de bateau, sa direction, permettant alors de mieux appréhender le contexte géopolitique d'une zone du globe.

Du point de vue de notre partenaire Airbus, ce projet est une réussite. Il leur a en effet permis de tester plusieurs algorithmes et d'avoir une preuve de concept fonctionnelle d'une méthode de détection de keypoints sur leurs images satellitaires. Pour nous, ce projet est également une réussite. Nous avons pu développer de nombreuses compétences : deep learning, computer vision, labellisation, management, cohésion d'équipe, et bien d'autres encore. Le fait d'avoir réussi à réaliser un prototype fonctionnel qui convienne à notre partenaire est également une vraie réussite pour nous.



# Bibliographie

YOLO :

<https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006>

<https://pjreddie.com/darknet/yolo/>

COCO : <https://arxiv.org/abs/1405.0312>

<https://cocodataset.org/#home>

Centernet : <https://arxiv.org/abs/1904.07850>

Resnet 50 : <https://iq.opengenus.org/resnet50-architecture/>

<https://datagen.tech/guides/computer-vision/resnet-50/>

Base pour le training du model :

<https://github.com/prabhakar-sivanesan/Custom-keypoint-detection>

DCNN :

<https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks#Deep-Convolutional-Neural-Networks-Explained>

