



COMMERCIAL AIRCRAFT

Soutenance Finale Master 2

Sujet : L'Atterrissage Basé Vision

Alya Zouzou

Encadrée par Mélanie Ducoffe et Jérôme Pasquet

27 Août 2025

AIRBUS

- I. Présentation de l'équipe
- II. Gestion de projet
- III. Contexte Industriel : Le VLS
- IV. Les données : LARD
- V. Mes 3 projets
- VI. Perspectives et conclusions

Sommaire

CRT (Center of Research and Technology) d'Airbus

- **Domaines d'études diverses** : Computer Vision, NLP, Recherche Opérationnelle, contraintes fiabilité/embarqué
- **Encadrante** : Mélanie Ducoffe - chercheuse en IA spécialisée en Vérification Formelle

Concernant la **gestion de projet** ...

1. Travail sur une **problématique unique** tout au long de l'année → saisie des **enjeux globaux**
2. **Autonomie**
3. Réunions **Hebdomadaires** et aides au **débuggage**

Contexte Industriel : l'atterrissage Basé Vision (VLS)

Une Ambition : créer un système autonome basé sur l'IA qui permet d'assurer un atterrissage sécurisé.

Assistance au pilote

1. Détection de la Piste

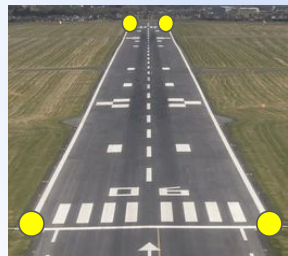
Input Image



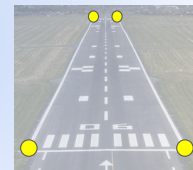
Crop



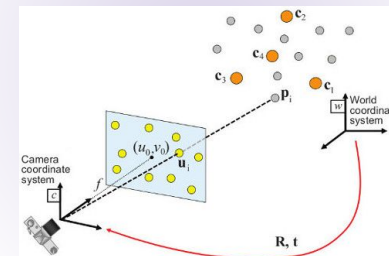
2. Extraction des coins 2D de la piste



Keypoints
Features
Extraction



3. Estimation de la position de l'avion



- **Contraintes computationnelles** : 20 images par secondes + pas de GPU
- **Robustesse** aux conditions difficiles et **Explicabilité** pour la certification.

Certification aux normes de l'EASA

Les données : Présentation du Dataset LARD

LARD : Landing Approach Runway Detection

- Dataset public pour la détection de pistes d'atterrissage **MONO classe**
- Vues **en approche** à différent niveaux :



Différentes conditions d'atterrissage :



- Images Synthétiques (via Google Earth Studio) et réelles :



Set	Type	Images
Train	Synthetic	11,546
Validation	Synthetic	2,886
Test	Real+Synth	2,315
Test Synth	Synthetic	2,212
Test Real	Real	103



Projet 1 : Benchmark de YOLOv5 vs YOLOv6 sur LARD

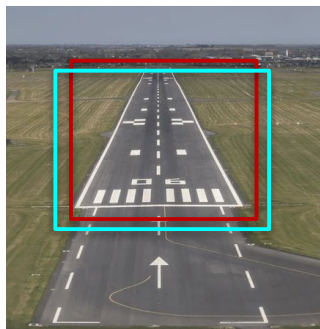
Établir un benchmark de certains modèles constituant **l'état de l'art** en OD sur le dataset **LARD**

⇒ **modèles YOLO (You Only Look Once)**

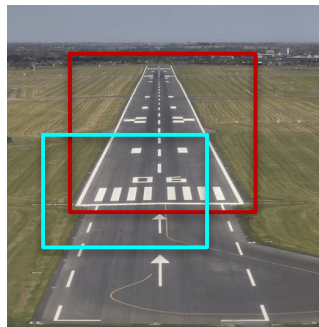
Mes Missions :

- Lire des articles pour comprendre leur fonctionnement → YOLOV5 et YOLOV6
- Construire la base de données associée
- Prendre en main certains tutoriels et les adapter à nos données
- Comprendre l'optimisation de paramètres (Optuna)
- Effectuer les entraînements

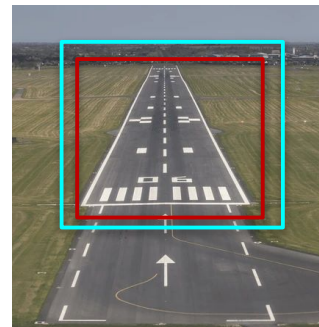
Recueillir des métriques permettant la comparaison des modèles → **mAP50** dans notre cas



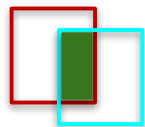
Bon IoU !



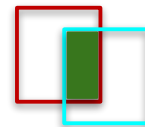
**Mauvais IoU
et IoA !**



Bon IoA !

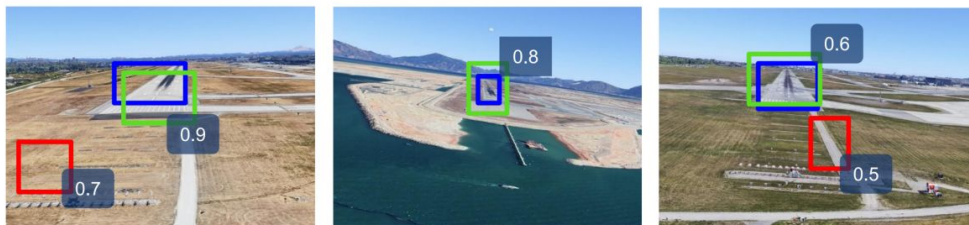


⇒ Plus la **zone commune**
est grande, plus l'**IoU** est
proche de **1**.
⇒ Valeur entre **0** et **1**



⇒ Plus la vérité terrain est
couverte par la prédiction,
plus l'**IoA** est proche de 1.
⇒ Valeur entre **0** et **1**

mAP (Mean Average Precision) = la qualité globale des détections, grâce à la précision moyenne sur toutes les classes, avec une valeur comprise entre **0 (mauvaise détection)** et **1 (détection parfaite)**.



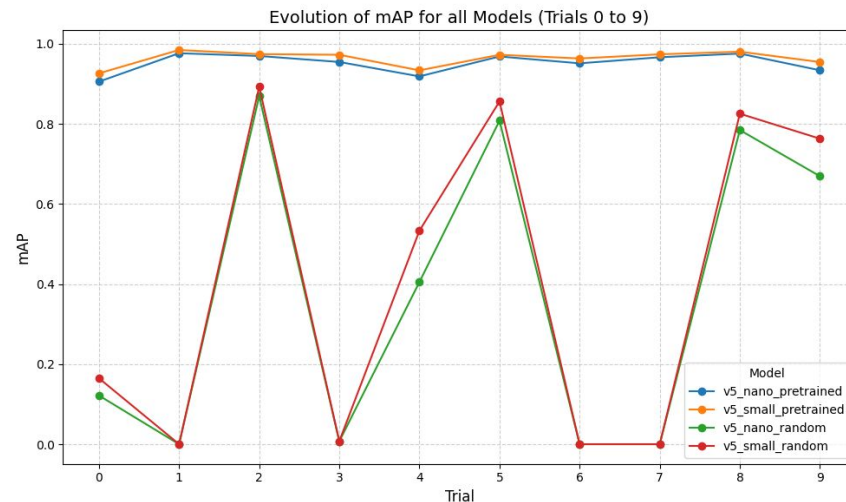
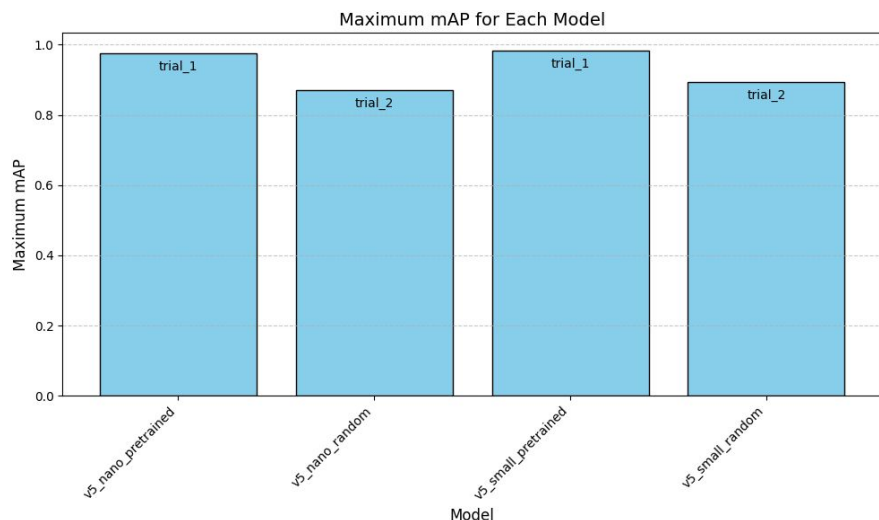
Confidence	0.9	0.8	0.7	0.6	0.5
b_{pred}					
$\text{IoU}(b_{\text{pred}}, b_{\text{gt}}) \geq \tau$	1	1	0	1	0
TP	1	2	2	3	3
FP	0	0	1	1	2
FN	2	1	1	0	0
Precision = $\frac{TP}{TP+FP}$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{3}{5}$
Recall = $\frac{TP}{TP+FN}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{3}{3}$	1	1

$$\text{mAP} = \frac{1}{C} \sum_c \underbrace{AP}_{\text{Métrique pour évaluer le matching - IoU par classe}} \left(\underbrace{\{b_i^x\}_{\hat{c}_i^x=c}}_{\hat{c}_i^x = \arg \max_k p_i^x(k)} \underbrace{\{b_j^{gt,x}\}_{c_j^x=c}}_{\text{GT}} \right),$$

Nombre de classes - ici 1
 Boîtes prédites avec les probabilités les plus grandes

⇒ **mAP = AP** car on a qu'une classe

- **v5_small_pretrained** → meilleures perfs (mAP ≈ 1.0)
- **v5_nano_pretrained** → très proche, légèrement en dessous
- **Modèles random** → mAP plus faibles, convergence difficile
 - Random → fluctuations, instabilité, besoin de plus d'époques & réglages
- **Pré-entraînés** → progression rapide, stable, résultats fiables



Projet 2 : Prédiction Conforme appliquée à la détection d'objets

*Robust Vision-Based Runway Detection through Conformal Prediction and Conformal mAP,
ZOUZOU et al 2025, COPA*

Petit rappel ...

1. Détection de la Piste



Crop



2. Extraction des coins 2D de la piste



Si la boîte englobante ne contient pas **entièrement** la vérité terrain et donc au moins 3 points clés \Rightarrow VLS échoue.



- Conformal Box
- Ground truth
- YOLO Prediction

Adapter le modèle d'IA a posteriori afin que les boîtes englobantes prédites recouvrent la vérité terrain avec des garanties statistiques.

\Rightarrow Évaluer l'impact de la conformalisation avec des métriques pertinentes pour l'industrie.


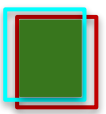

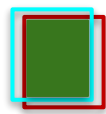




Mais comment ? en appliquant la prédiction conforme (e.g., De Grancy et al.) aux sorties du YOLO.

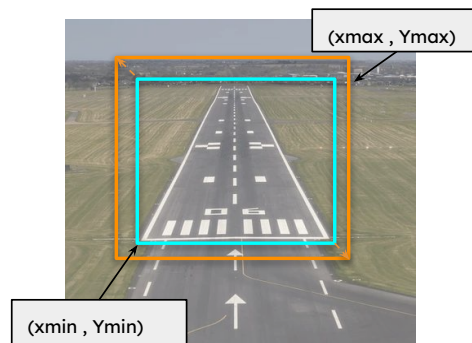
HYPOTHÈSE :

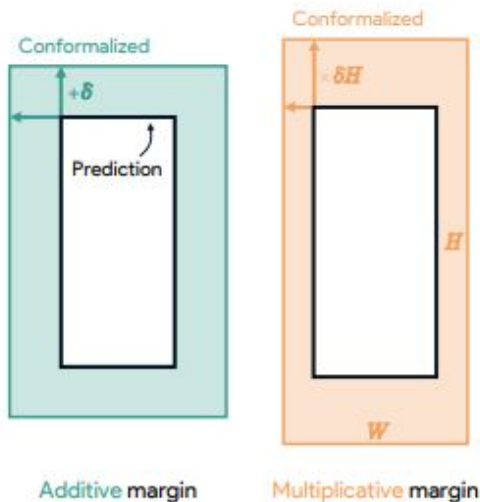
- Données de test i.i.d. par rapport à la calibration
- **Indépendance** : chaque donnée ne dépend pas des autres
- **Distribution identique** : même loi pour test & calibration

Étapes clés de la détection d'objets conforme :

1. **Entraîner un prédicteur de base** → YOLO
2. **Constituer un ensemble de calibration** → indépendant de l'ensemble d'entraînement
3. **Conserver uniquement les vrais positifs** (basés sur l'IoU)
4. **Calculer 4 scores de non-conformité** → écarts entre les prédictions et la vérité terrain, pour les 4 coordonnées de la boîte (xmin, ymin, xmax, ymax)
5. **Élargir la boîte prédite**

Élément de Calibration				
Vrai positif ?				





Pour des objets de
taille similaires

Pour des objets de
différentes
tailles

Par exemple, pour la pénalité additive:

1. Calcul des scores de non-conformités

$$\mathbf{r}_k^a = (\hat{x}_{\min}^k - x_{\min}^k, \hat{y}_{\min}^k - y_{\min}^k, x_{\max}^k - \hat{x}_{\max}^k, y_{\max}^k - \hat{y}_{\max}^k)$$

4 distributions distinctes ... mais est-ce que le risque α est respecté ?



2. SOLUTION = Quantiles plus strictes

$$\hat{q}(j) = q_{\lceil (1-\frac{\alpha}{4})(n+1) \rceil / n}(\{r_k(j) : k \in \{1, \dots, n\}\})$$

$$\mathbb{P}(\mathbf{b}_{n+1}^{gt} \subseteq \hat{\mathbf{b}}_{n+1}^{\text{conf}}) \geq 1 - \alpha$$

$$x_{\min} \geq \hat{x}_{\min}^{\text{conf}}, \quad y_{\min} \geq \hat{y}_{\min}^{\text{conf}}, \quad x_{\max} \leq \hat{x}_{\max}^{\text{conf}}, \quad y_{\max} \leq \hat{y}_{\max}^{\text{conf}}$$

Gauche Droite

3. Modification des coordonnées

$$\hat{\mathbf{b}}_k^{\text{conf}} = (\hat{x}_{\min}^k - \hat{q}_1, \hat{y}_{\min}^k - \hat{q}_2, \hat{x}_{\max}^k + \hat{q}_3, \hat{y}_{\max}^k + \hat{q}_4)$$

L'IoU et IoA sont-elles des métriques robustes ?

Comment garantir une
détection robuste ?

avec $\text{IoA} = 1$!

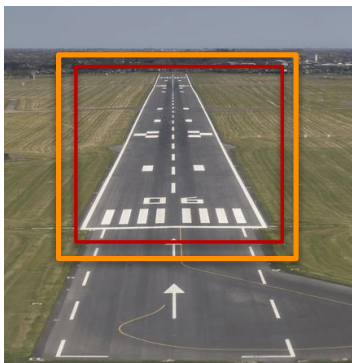


$\text{IoA} = 1$
&
 $\text{IoU} = 0.4$



Seulement ?

Non aussi
avec un seuil
d'IoU
minimum ...



$\text{IoA} = 1$
&
 $\text{IoU} = 0.9$



	$\text{IoU} > t$	$\text{IoA} = 1$	Robust ?
YOLO	✓	✗	✗
C-YOLO	?	✓	?

$$\sum_i \mathbb{1}_{Y_i \in \mathcal{C}_{\hat{\lambda}}(X_i)}$$

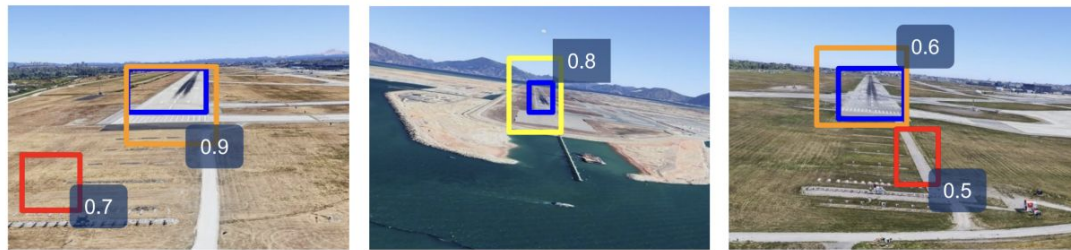
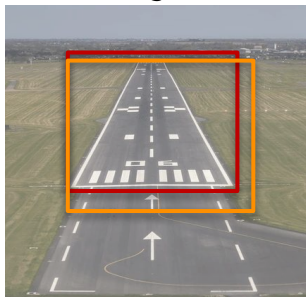
1 si la **GT** Y_i pour un input X_i est contenu dans la bounded box conforme $\mathcal{C}_{\hat{\lambda}}(X_i)$

0 si la **GT** Y_i pour un input X_i est **pas** contenu dans la bounded box conforme $\mathcal{C}_{\hat{\lambda}}(X_i)$

Coverage = 1



Coverage = 0



Confidence	0.9	0.8	0.7	0.6	0.5
b_{pred}					
$\text{IoU}(b_{\text{pred}}, b_{\text{gt}}) \geq \tau$	1	0	0	1	0
$\text{IoA}(b_{\text{pred}}, b_{\text{gt}}) = 1$	1	1	0	1	0
TP	1	1	1	2	2
FP	0	1	2	2	3
FN	2	2	2	1	1
Precision = $\frac{TP}{TP+FP}$	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{5}$
Recall = $\frac{TP}{TP+FN}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$

Model	mAP	C-mAP	C-mAP@50@80:100
YOLOv5	96.88	0.77	46.92
c-YOLOv5-a	92.67	56.86	80.73
c-YOLOv5-m	96.17	55.84	82.18
YOLOv6	98.13	1.31	51.94
c-YOLOv6-a	95.09	55.75	81.86
c-YOLOv6-m	96.71	52.71	81.93

⇒ Conformalisation : +51 à 55 pts de C-mAP (52,7 % → 56,9 %)

⇒ **mAP reste élevé : > 92 %**

⇒ Pénalisation **multiplicative** = meilleur compromis mAP / C-mAP

Model	Coverage
c-YOLOv5-a	77.06
c-YOLOv5-m	75.88
c-YOLOv6-a	75.73
c-YOLOv6-m	73.93

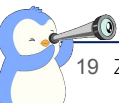
⇒ Pénalisation **multiplicative** = meilleur équilibre

⇒ Particulièrement efficace avec **c-YOLOv5-m**

Perspectives :

⇒ limitations à l'évaluation d'une seule piste

⇒ intégration de contraintes robustes dans l'entraînement ?



Projet 3 :

Vers une pipeline End-to-End intégrant l'Estimation de position avec YOLO-NAS-POSE

But : Etudier la faisabilité d'un pipeline "towards E2E" et intégrer la *pose* de la caméra dans l'apprentissage



Approche Préliminaire

Mes Missions :

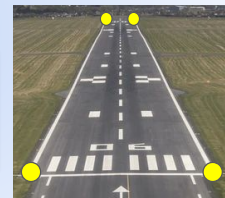
- Etudier le fonctionnement de YOLO-NAS-POSE
- Comprendre le terme *pose*
- Comprendre le P3P et l'implémenter en Pytorch
- Fine-Tuner YOLO-NAS-POSE sur LARD
- Incorporer la pose de la caméra dans l'apprentissage

FUSION

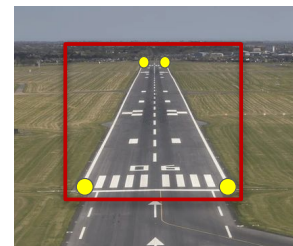
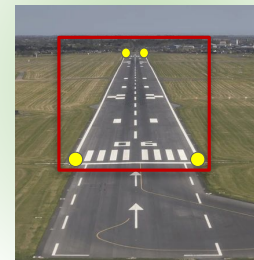
1. Détection de la Piste



2. Extraction des coins 2D de la piste



Avec YOLO-NAS-POSE



De la pose 2D humaine à la pose d'une piste d'atterrissage ...

YOLO-NAS Pose (Top-Down):

- Détection de la piste (bounding box).
- Prédiction des keypoints relatifs à la boîte → conversion en absolu.

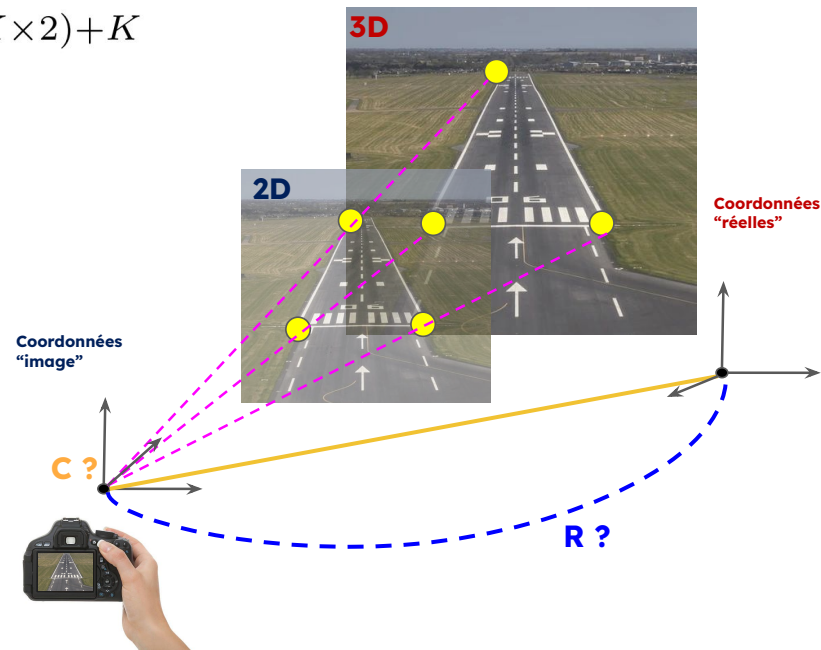
⇒ Résultat : pose 2D (squelette / repères projetés sur l'image).

Passage en 3D (PnP / P3P)

- Association : points 2D détectés ↔ points 3D connus.
- Avec les paramètres intrinsèques de la caméra (matrice A).

⇒ Résolution du problème Perspective-n-Point (PnP).

$$\mathbf{y}_{pose} \in \mathbb{R}^{1+4+(K \times 2)+K}$$



Objectif : estimer la pose de la caméra (R, C) par rapport à la piste d'atterrissage.

⚠ Jusqu'à 4 solutions possibles → garder la plus cohérente (erreur de reprojection minimale)



⚠ **Problème difficile** : plus complexe que la simple détection

From scratch :

- AP/AR comparables aux poids pré-entraînés (AP \approx 0.30, AR \approx 0.52)
- Mais **scores de confiance très faibles** → résultats peu exploitables

Avec poids COCO-POSE :

- **AP/AR similaires**, mais **confiance beaucoup plus élevée** (0.73–0.79 keypoints, 0.265 global)
- Calibration meilleure → convergence plus rapide & modèle exploitable?



(a) Prédiction avec des poids aléatoires après 200 epochs : faible confiance



(b) Prédiction après fine-tuning avec des poids pré-entraînés après seulement 80 epochs : confiance élevée

Setup	Epoch	Train Loss	Val Loss	AP	AP@0.50	AR
COCO Weights	50	2.2037	2.1310	0.2728	0.3052	0.5131
	98	1.7963	1.8772	0.2822	0.3168	0.5224
No Weights	50	2.7741	2.2378	0.2775	0.3085	0.5083
	100	2.1083	1.9026	0.2905	0.3217	0.5139
	198	1.6147	1.6510	0.2987	0.3355	0.5249

Setup	Keypoint 1	Keypoint 2	Keypoint 3	Keypoint 4	Keypoints globaux	Détection
COCO Weights	0.793	0.795	0.741	0.734	0.766	0.265
No Weights	0.300	0.300	0.271	0.269	0.285	0.077

- Estimation de position simulée & premiers tests réalisés
- Pose-aware loss en cours de développement (non encore finalisée)
- Pas encore intégrée à l'entraînement de YOLO-NAS-POSE
- **Prochaine étape** : test sur le jeu de données LARD v2 (2D-3D complet)
- **Transfert de connaissances au prochain alternant**

Conclusions et Perspectives

- Compétences **renforcées** : Deep Learning, PyTorch, recherche, anglais
- Expérience en **cohérence** avec mon Master MIAHS
- Envie de poursuivre en **recherche appliquée** (thèse ou ingénieure de recherche)
- **Ouverture internationale**, spécialisation à affiner (CV / NLP)





Merci !

© Copyright Airbus Operations 2025 / Soutenance Master 2

This document and all information contained herein is the sole property of Airbus. No intellectual property rights are granted by the delivery of this document or the disclosure of its content. This document shall not be reproduced or disclosed to a third party without the expressed written consent of Airbus. This document and its content shall not be used for any purpose other than that for which it is supplied.

Airbus, its logo and product names are registered trademarks.

Découpage : grille (13×13, 19×19...)

Responsabilité : cellule = centre de l'objet

Prédiction :

- Coordonnées relatives (centre, largeur, hauteur → anchors)
- Score de confiance
- Classe

Taille des boîtes : pas limitée à la cellule

- 👉 grâce aux anchors (formes/gabarits prédéfinis)
- et aux offsets relatifs appris (transformations → coordonnées absolues)

Multiples boîtes : plusieurs anchors par cellule

Nettoyage : seuil de confiance + NMS

