# Détection d'écocups sans réseaux de neurones

Massyle Oumessaoud - Yazan El Mahmoud
massyle.oumessaoud@etu.utc.fr, yazan.elmahmoud@etu.utc.fr
Contribution équitable - Juin 2025

***Abstract*—La détection d'écocups sans réseaux de neurones est le défi de cette étude dans le cadre de l'unité de valeur SY32 enseigné par Julien Moreau [1]. Après diverses tentatives, il s'avère que l'approche histogram orientation gradient (HOG) avec support vector machine (SVM) et la recherche sélective nous apportent les meilleurs résultats, malgré d'autres pistes très prometteuses.**

## I. Introduction

Ce rapport présente la conception et l'implémentation de notre pipeline de détection d'écocups. Nous y détaillons nos choix méthodologiques ainsi que les différentes approches que nous avons mises en place. Dans un premier temps, nous exposons notre stratégie de prétraitement des données. Nous abordons ensuite deux approches d'extraction de caractéristiques et de classification, avant de décrire les différentes techniques de détection que nous avons développées. Enfin, nous présenterons nos stratégies d'optimisation des résultats ainsi que des pistes potentielles pour des améliorations futures.

Pour chaque partie, nous expliquons la démarche adoptée, nous présentons les résultats obtenus et nous en proposons une interprétation. Les grandes sections suivent un ordre chronologique, mais leur contenu se plonge dans les détails des solutions apportées et des analyses menées durant toute la période du projet.

De l'intelligence artificielle générative fut utilisée comme soutien pour l'implémentation du code uniquement.

## II. Prétraitement des données

### A. Découpage des annotations

La première étape a consisté à récupérer nos écocups annotés et à les visualiser afin de vérifier la qualité des annotations. Nous avons repéré des erreurs d'annotation dans le jeu de données, qui ont été supprimées pour éviter de biaiser notre modèle. Cette étape nous a également permis de réaliser une étude statistique exploratoire pour mieux comprendre nos annotations. Par exemple, la médiane des dimensions des bounding boxes était de $313 \times 186$ pixels.

Nous avons donc testé un premier pipeline avec une taille d'entrée fixée à $313 \times 186$ pixels. Chaque bounding box était redimensionnée à cette taille, en préservant le ratio et en ajoutant éventuellement du padding.

Cependant, les résultats obtenus avec cette taille se sont révélés nettement inférieurs à ceux obtenus avec une taille plus compacte, notamment $64 \times 128$ pixels. Plusieurs facteurs peuvent expliquer cette baisse de performance :

Le redimensionnement à une taille aussi grande dilue le signal utile et introduit du bruit.

Le modèle n'est pas optimisé pour gérer une telle résolution.

L'augmentation de la taille d'entrée complique l'apprentissage sans apporter de gain significatif.

### B. Augmentation du jeu de données

Avec ces annotations nettoyées, nous avons décidé de recourir à l'augmentation des données en appliquant des transformations à nos images, telles que de petites rotations, l'ajout de bruit ou la modification de la luminosité (gamma). Nous avons également testé l'entraînement sans augmentation des données, mais les métriques (accuracy, F1-score) de nos classifications se sont avérées moins bonnes, malgré l'utilisation de la validation croisée. Cela confirme l'intérêt de l'augmentation pour améliorer la robustesse du modèle.

### C. Génération initiale de patchs négatifs par fenêtre glissante

Dans un premier temps, nous avions appliqué nos mêmes techniques d'augmentation (petites rotations, ajout de bruit, correction gamma) aux images destinées à constituer le jeu de patchs négatifs. Malgré ces efforts, l'entraînement du SVM sur ces exemples négatifs augmentés s'est traduit par un trop grand nombre de faux positifs et une généralisation insuffisante.

Pour y remédier, nous avons repensé la création des négatifs dès la phase initiale :

1) Nous avons extrait des patchs négatifs par fenêtre glissante sur l'ensemble des images (négatives globales), en appliquant à chaque patch les mêmes transformations d'augmentation qu'aux positifs.
2) Nous avons également extrait, dans les images positives, des patchs hors de la zone de la bounding-box (donc garantis négatifs), eux aussi augmentés de la même manière.

Cette stratégie permet au classifieur de rencontrer dès l'entraînement des exemples négatifs beaucoup plus représentatifs des cas réels rencontrés lors de la détection. Nous observons ainsi une nette diminution du taux de faux positifs, tout en conservant le niveau de rappel atteint précédemment.

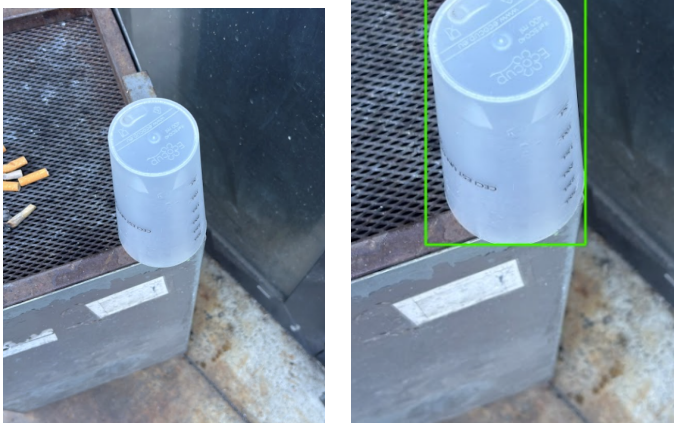
### D. Augmentation ciblée : rotation des écocups

L'une des erreurs les plus fréquentes de notre premier modèle concernait les écocups présentés *à l'envers* (rotation de $180°$) ou posés horizontalement (rotation de $90°$). Le descripteur HOG étant sensible à l'orientation globale des gradients, ces cas de figure ne ressemblaient pas suffisamment aux exemples vus à l'entraînement ; la probabilité prédite tombait souvent sous le seuil de validation.

*a) Stratégie mise en place:*

1) Pour chaque patch positif, deux versions supplémentaires ont été générées :
   - une rotation de $180°$ (écocup totalement inversé) ;
   - deux rotations de $90°$ et $-90°$ pour simuler un écocup couché.
2) Le classifieur SVM a été ré-entraîné avec ce corpus enrichi, en conservant exactement le même pipeline de feature extraction (HOG + HSV + LBP) et la même validation croisée.

*b) Résultat qualitatif:* La Figure 1 illustre l'impact direct de cette augmentation : l'écocup inversé qui n'était pas détecté auparavant est désormais correctement localisé après ré-entraînement.




(a) Avant augmentation (b) Après augmentation

Fig. 1: Impact de l'ajout des rotations : l'écocup inversé est désormais détecté.

*c) Analyse:*

- **Taux de rappel amélioré** : les écocups inversés ou couchés sont désormais intégrés à la distribution de référence, ce qui réduit significativement les faux négatifs observés dans cette configuration.
- **Faux positifs stables** : aucune hausse notable n'a été observée sur les faux positifs, car les nouvelles vues restent proches des motifs d'intérêt (contours / texture) déjà appris ; le SVM généralise sans confusion supplémentaire.
- **Coût de calcul inchangé** : l'extraction des descripteurs et l'inférence n'augmentent pas, car la modification se situe au niveau du corpus, non du pipeline.

Cette simple rotation ciblée illustre l'intérêt d'un *data augmentation* guidé par l'erreur : plutôt que de multiplier les transformations au hasard , il s'agit d'ajouter précisément les vues que le modèle échoue à reconnaître, puis de ré-entraîner pour combler la lacune. Cette démarche pourra être reproduite sur d'autres variantes géométriques (perspective, léger tilt) à mesure que de nouvelles erreurs apparaîtront.

### *E. Création des patchs négatifs*

Une fois les patchs positifs récupérés et augmentés, nous avons créé des patchs négatifs à l'aide d'une fenêtre fixe et glissante. Nous pensons qu'une amélioration est possible en générant ces patchs négatifs à partir des régions proposées par la recherche sélective. Nous avons veillé à respecter des effectifs équivalents entre les patchs positifs et négatifs afin de limiter les biais dans l'apprentissage du classificateur.

## III. Extraction des caractéristiques et classification

### *A. Paseudo-Haar et Adaboost*

Nous avons créé quatre filtres de type Haar, comme illustré par la Figure 2. Ces filtres ont été appliqués à différentes échelles afin d'extraire des caractéristiques en parcourant les patchs positifs. Leur rôle consiste uniquement à générer des descripteurs simples (contrastes, motifs locaux), qui sont ensuite exploités par l'algorithme AdaBoost pour sélectionner les caractéristiques les plus discriminantes et construire le classifieur final. Ce dernier a atteint un score d'accuracy de 0,77.

Cette approche a été poussée jusqu'à la détection avec une fenêtre glissante, mais elle a été abandonnée, car le développement en parallèle d'une méthode basée sur HOG et SVM a montré un meilleur potentiel en termes de performance.

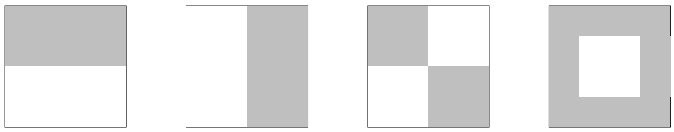



Fig. 2: Exemple schématisant les filtres Haar.

### *B. HOG et SVM*

Plutôt qu'un simple HOG en niveaux de gris, nous avons choisi une classe HOG améliorée avec un vecteur composite plus puissant pour chaque patch ($64 \times 128$).

Les descripteurs HOG sont calculés indépendamment sur les canaux bleu, vert et rouge et ils sont ensuite concaténés. Le patch est converti dans l'espace de couleur HSV [2] et les histogrammes sont ensuite calculés.

Avec cette méthode, on ajoute aussi un Local Binary Pattern (LBP) qui permet de capturer des motifs de texture.

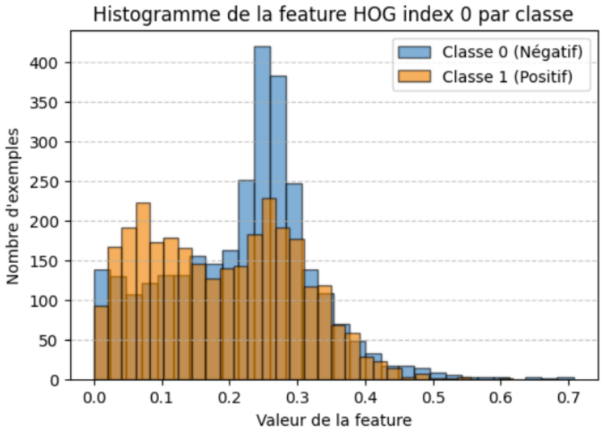
Notre vecteur final combine forme (HOG), couleur (HSV) et texture (LBP) en un seul vecteur descriptif très complet (plus de 3700 dimensions).

### *C. Réduction dimensionnelle avec une analyse des composantes principales (ACP)*

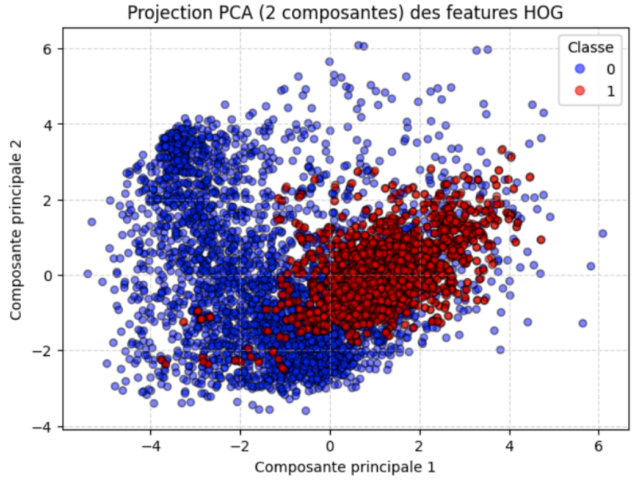
Un vecteur aussi riche est coûteux en calcul et peut entraîner un surapprentissage (fléau de la dimension).

L'ACP projette les données dans un espace de plus faible dimension en maximisant la variance expliquée. La Figure 3

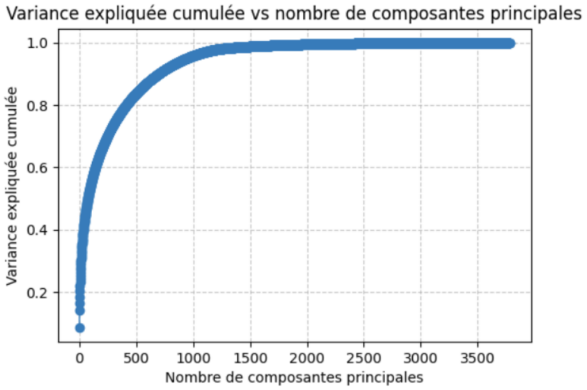
présente, d'une part, l'histogramme d'une des composantes HOG par classe, la projection 2D des deux premières composantes, et la courbe de variance cumulée. D'autre part, le Tableau I récapitule le nombre de composantes nécessaire pour différents seuils de variance expliquée.




(a) Histogramme feature HOG




(b) Projection PCA (2 comp.)




(c) Variance expliquée cumulée

Fig. 3: Analyse visuelle des composantes principales pour les descripteurs HOG.

TABLE I: Nombre de composantes pour différents niveaux de variance expliquée

| Variance expliquée (%) | Nombre de composantes |
|---|---|
| 40 | 36 |
| 45 | 53 |
| 50 | 75 |
| 55 | 105 |
| 60 | 142 |
| 64 | 189 |
| 70 | 247 |
| 75 | 320 |
| 80 | 413 |
| 84 | 534 |
| 89 | 698 |
| 94 | 944 |
| 99 | 3781 |

### *D. Entraînement du SVM*

Nous utilisons toujours la validation croisée pour entraîner et tester notre modèle. De plus, un Grid Search est utilisé pour optimiser les hyperparamètres $C$ et $\gamma$.

## IV. Détection

Une fois le classificateur SVM entraîné, il est utilisé pour détecter la présence des écocups dans les images de test. Nous avons essayé plusieurs stratégies :

- **Fenêtre glissante avec pyramide d'images :** Une fenêtre de taille fixe (64 $\times$ 128 pixels) se déplace sur l'image à différentes échelles grâce à une pyramide d'images. Le paramètre à ajuster est le nombre de niveaux dans la pyramide. Les limites de cette méthode sont qu'elle est coûteuse, car il faut parcourir l'image de manière exhaustive, et qu'il est difficile de balayer l'image de façon optimale. En effet, la fenêtre fixe ne s'adapte pas correctement aux différentes positions et tailles des écocups recherchés. Nous avons donc testé une nouvelle méthode...
- **Recherche sélective :** En regroupant des segments homogènes en termes de couleur, de texture et de forme. Cette méthode permet de réduire le nombre de fenêtres analysées tout en augmentant la pertinence des régions sélectionnées. Nous avons ainsi légèrement réduit notre temps de calcul; cet avantage reste toutefois relatif, car la recherche sélective génère souvent un grand nombre de régions candidates, dont beaucoup sont mal adaptées. Malgré cela, nous avons optimisé cette méthode au maximum afin d'obtenir nos meilleurs résultats.
- **EdgeBoxes et Simple Linear Iterative Clustering (SLIC) superpixels :** En fin de projet, nous avons essayé une nouvelle approche qui combine des EdgeBoxes, génèrant des propositions de boîtes basées sur les contours détectés dans l'image (via un modèle de détection structurée des bords). Et SLIC [3] qui segmente l'image en superpixels compacts. La Figure 4 illustre les images obtenu après ce traitement, par contrainte de temps nous n'avons pas optimisé cette méthode mais nous avons beaucoup d'espoir avec cette dernière car elle est non seulement plus mais aussi plus efficace.

Fig. 4: Exemple de régions proposés après EdgeBoxes et SLIC superpixels

À l'issue de ce processus, nous appliquons toujours une suppression des non-maximums (NMS) afin de fusionner les détections redondantes et d'affiner nos résultats. Le seuil d'IoU et la prise en compte du meilleur score peuvent également être optimisés pour améliorer encore la précision des détections.

## V. Optimisation

### A. Extraction des faux positifs difficiles

Un cap important a été franchi lorsque nous avons décidé d'intégrer le hard negative mining : nous récupérons les faux positifs générés par notre modèle et réentraînons ce dernier en l'exposant à ces mauvaises détections. Nous avons appliqué cette technique juste après un premier entraînement, en fournissant de nouveau au modèle toutes les images identifiées à tort comme des objets positifs dans notre jeu de données négatif. Cela a permis d'améliorer considérablement nos résultats, même si nous n'avons pas encore atteint d'excellentes performances. Une piste prometteuse serait d'effectuer ce hard negative mining de manière plus ciblée, en l'appliquant après la phase de détection en comparant directement avec les annotations de vérité terrain. Cette approche, un peu plus complexe à mettre en œuvre, devrait toutefois se révéler plus efficace.

### B. Parallélisme

Pour gagner du temps, nous avons utilisé tardivement de la parallélisation grâce à la librairie RAPIDS [4], conçue pour une utilisation avec Google Colab que nous avons utilisée pour travailler en binôme. Cette dernière stratégie permet de tester et d'optimiser bien plus rapidement.

### C. Repartir de zéro

Nous sommes repartis de zéro à un moment du projet, en ré-annotant l'ensemble des images et en poursuivant avec les méthodes décrites précédemment. Cela nous a permis de mieux comprendre chaque étape du pipeline et de prendre du recul sur chacune d'elles.

## VI. Évaluation des performances des méthodes

Nos différentes expérimentations ont permis d'identifier les situations dans lesquelles nos approches sont les plus performantes, ainsi que leurs principales limites.

L'association HOG + SVM, enrichie par les descripteurs couleur et texture et combinée à la recherche sélective, donne de bons résultats dans les cas suivants :

- Les écocups sont clairement visibles, bien séparés de l'arrière-plan, avec un contraste suffisant pour que les descripteurs capturent efficacement leurs contours et leurs motifs.
- Les écocups apparaissent à des échelles proches de celles rencontrées à l'entraînement, ce qui facilite leur détection par la fenêtre glissante ou la pyramide d'images.
- Les conditions d'éclairage sont homogènes et ne génèrent pas d'ombres parasites perturbant les caractéristiques extraites.

En revanche, les performances se dégradent dans plusieurs cas :

- Les écocups sont partiellement occultés, fortement déformés, ou mal segmentés par la recherche sélective, ce qui réduit la qualité des propositions de régions.
- L'image comporte des objets ou motifs présentant des caractéristiques similaires aux écocups (textures, couleurs ou formes cylindriques), ce qui augmente le taux de faux positifs.
- Les scènes sont complexes, avec des arrière-plans chargés, ce qui génère un grand nombre de régions candidates et complique la classification.
- La fenêtre glissante et la pyramide d'images, bien qu'exhaustives, sont coûteuses en calcul et peu efficaces sur des images de grande taille ou avec des écocups de dimensions très variables.

Comme nous l'avons déjà constaté, l'intégration des faux positifs difficiles (hard negative mining) permet de renforcer la robustesse du classificateur, bien que les gains deviennent rapidement marginaux sans un raffinement plus poussé de la stratégie de génération des faux positifs ou de la méthode de détection.

Nous rappellons notre plus récente piste testée (EdgeBoxes et SLIC superpixels) qui présente un potentiel intéressant, mais nécessite des optimisations supplémentaires pour être exploitées pleinement dans des contextes variés.

## VII. Discussion

### A. Comparaison globale des pipelines

Nous avons successivement évalué quatre familles d'approches :

- **Haar + AdaBoost**
- **HOG + SVM** (fenêtre glissante)
- **HOG + SVM** (couplé à la recherche sélective)
- **EdgeBoxes + SLIC + HOG + SVM**

Chaque pipeline présente un compromis distinct entre exactitude et coût de calcul ; les paragraphes suivants détaillent leurs atouts et limites observés empiriquement.

*Haar + AdaBoost:* Cette solution est la moins efficace : La représentation Haar manque de robustesse : dès qu'un écocup est partiellement occulté ou montré sous un angle peu familier, le classifieur produit de nombreux faux positifs. Elle convient donc plutôt à des scénarios très contrôlés.

*HOG + SVM + fenêtre glissante:* En l'absence de mécanisme de propositions de régions, la fenêtre glissante doit balayer l'image de façon exhaustive, ce qui devient rapidement prohibitif sur des images haute définition. Lorsque l'échelle de l'objet correspond bien à la taille de la fenêtre, la précision est satisfaisante, mais le rappel s'effondre dès que l'écocup devient trop petit ou trop incliné.

*HOG + SVM + recherche sélective:* La recherche sélective filtre la majorité des fenêtres inutiles ; le pipeline tient alors une cadence raisonnable tout en maintenant un bon taux de détection. Son principal inconvénient est la génération de nombreuses régions quasi redondantes : une suppression des non-maximums (NMS) agressive est indispensable, sous peine d'avoir un trop fort chevauchement de boîtes (crowding) dans les résultats.

*EdgeBoxes + SLIC:* Bien que nous n'ayons pas eu le temps de pousser l'optimisation, ce prototype révèle un potentiel intéressant : il parvient à proposer des boîtes correctes pour des écocups de tailles très variées. Cependant, il introduit aussi un grand nombre de faux positifs. Une calibration plus fine des seuils (score de contours, granularité SLIC) ou un reclassement des boîtes candidates serait nécessaire avant un usage opérationnel.

### B. Limites actuelles

- **Déséquilibre positif/négatif.** Les écocups représenteraient une proportion relativement faible des patchs disponibles ; il est donc difficile d'entraîner un SVM sans sur-représenter artificiellement les positifs.
- **Sensibilité à la résolution.** Les très petits écocups (quelques dizaines de pixels de hauteur) échappent encore aux descripteurs HOG, qui reposent sur une maille spatiale fixe.
- **Variabilité lumineuse.** Les scènes à fort contre-jour ou exposition inégale déstabilisent les histogrammes de couleur et les gradients.

### C. Pistes d'amélioration

1) **Ensemble de classifieurs** : combiner plusieurs SVM (linéaire, RBF) ou ajouter un classifieur par arbre léger pour adoucir les cas limites et réduire les faux positifs récurrents.
2) **Apprentissage incrémental** : intégrer un cycle de hard-negative mining basé sur les nouveaux faux positifs rencontrés.
3) **Fusion bord + texture** : utiliser un détecteur de contours (par ex. Canny) pour restreindre l'espace de recherche avant passage au SVM, ce qui accélérerait l'inférence tout en améliorant la précision dans les arrière-plans complexes.

### D. Analyse qualitative des prédictions

Un examen manuel des résultats fait ressortir trois catégories d'erreurs :

1) **Faux négatifs** lorsque l'écocup est partiellement hors cadre ou caché par une main ; les contours typiques deviennent inexploitables.
2) **Faux positifs** sur des objets cylindriques (canettes, bouteilles) partageant une texture ou une couleur proche ; un descripteur plus discriminant serait nécessaire.
3) **Miss-detections** sous éclairage extrême (contre-jour, spot lumineux) où les gradients dominants sont écrasés ou saturés.

Dans l'ensemble, le couple HOG + SVM accompagné de la recherche sélective rest e le meilleur compromis entre exactitude et coût de calcul, tout en laissant une marge de progression notable via le hard-negative mining et la fusion de détecteurs de bords.

## VIII. Résultats de la détection

Les performances de notre modèle de détection sont présentées dans le Tableau II. Les colonnes sans astérisque correspondent aux résultats sur l'ensemble de test sans les images "difficiles", tandis que les colonnes marquées d'un * incluent aussi les images jugées difficiles dans l'ensemble de test.

TABLE II: Résultats de la détection

| | Précision | Rappel | F1 | AUC | Précision* | Rappel* | F1* | AUC* |
|---|---|---|---|---|---|---|---|---|
| Modèle | 20.00 | 16.92 | 18.33 | 12.58 | 20.00 | 14.86 | 17.05 | 12.27 |

On observe que l'inclusion des *tests jugés difficiles* (colonnes * ) conserve la même précision globale, tout en faisant légèrement baisser le rappel, ce qui se traduit par un F1 et un AUC très proches de ceux obtenus sur l'ensemble plus "facile".

## IX. Conclusion

Dans ce travail, nous avons démontré que, malgré l'absence de réseaux de neurones, il est possible de construire un pipeline de détection d'écocups opérationnel en combinant des descripteurs classiques (HOG, HSV, LBP) et un classifieur SVM.

Nous avons exploré plusieurs stratégies de génération de candidats (fenêtre glissante, recherche sélective, Edge-Boxes+SLIC) et montré que l'association HOG+SVM, enrichie par la recherche sélective, offrait le meilleur compromis entre précision et coût de calcul. L'ajout du hard negative mining a permis de stabiliser la précision tout en contrôlant le taux de rappel.

La section Discussion a mis en lumière les forces et les limites de chaque approche, et souligné l'importance d'un data augmentation ciblé: en intégrant dès l'entraînement des rotations (à 90° et 180°) et des patchs négatifs par fenêtre glissante issus des images positives et négatives, nous avons significativement réduit les faux positifs, sans alourdir le pipeline.

Pour aller plus loin, l'ajout d'un re-classement post-proposition, l'optimisation des seuils de filtrage, ou l'intégration de détecteurs de contours adaptatifs constituent des pistes prometteuses pour améliorer encore les performances et réduire le coût de calcul.

## REFERENCES


[1] J. Moreau, "Sy32 – vision et apprentissage," université de Technologie de Compiègne, Département Génie Informatique, Printemps 2025.

[2] F. Hamdaoui, S. Bougharriou, M. Gueddari, and A. Sakly, "Extraction of image features based on the hog method in the hsv color space," in *2022 IEEE 21st international Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Sousse, Tunisia, 2022, pp. 265–270.

[3] IVRL, EPFL, "Slic superpixels," https://www.epfl.ch/labs/ivrl/research/slic-superpixels/.

[4] R. D. Team, "RAPIDS: Collection of libraries for end-to-end gpu data science," https://rapids.ai, 2025, nVIDIA Corporation.