

Rapport de STAGE



Esteban Carrasco
Paris, Septembre 2025



Rapport de stage

Ingénierie d'étude sur le projet *Dance*
précisement sur la GeoDanceHive

Esteban Carrasco

Etudiant No. 725542

Maître de stage : Guillaume TOUYA
Tuteur entreprise

Responsable pédagogique : Cécile FRANKART
Tutrice école

Co-Directrice projet : Freddie-Jeanne RICHARD
Co-Directrice projet Dance

École supérieure d'ingénieur Léonard de Vinci (ESILV)
Majeur objets connectés & cybersécurité (OCC)
Institut national de l'information géographique et forestière (IGN)
Laboratoire en sciences et technologies de l'information géographique (LASTIG)

Stage de Première année de Master (A4)

Paris, Septembre 2025

Rapport de stage

Copyright © 2025 - Esteban Carrasco, École supérieure d'ingénieur Léonard de Vinci (ESILV).

Ce mémoire est un travail original et toutes les personnes dont les études et publications ont contribué y sont dûment citées. La reproduction partielle est autorisée avec mention de l'auteur.



Je reconnais avoir utilisé [Chat GPT](#) pour affiner le ton et améliorer la précision linguistique de ce travail, notamment sur la grammaire, la ponctuation, le vocabulaire ou les traductions.

La réalisation graphique de ce rapport a été inspirée du template [IPLeiria-Thesis](#).

Tous les programmes, scripts et comptes rendus utilisés pour la rédaction de ce rapport sont disponibles sur ce dépôt [GitHub](#).

Remerciements

C'est ainsi que ces quatre mois s'achèvent, la tête remplie de souvenirs et le cœur lourd. Non, je n'exagère pas ni ne dramatise lorsque j'écris ces mots. Cette expérience a, pour moi, été le marqueur d'une nouvelle lumière. Ce stage, bien qu'il le soit, n'est pas qu'une simple étape académique dans un parcours où quatre mois représentent si peu. Il est la lueur d'une nouvelle possibilité, d'un nouveau chemin derrière la porte de mon avenir. Je pensais être sûr, voire convaincu, de ce que je comptais faire de mon futur, mais c'était sans compter sur ce stage qui m'a humblement remis les pieds sur terre, ajoutant à la longue liste de mes futurs aspirés celui d'un chercheur passionné.

Cela, je leur dois. Ainsi, je leur adresse ces remerciements. Je ne peux évidemment pas tou·te·s les citer tant iels sont nombreu·ses ; c'est pourquoi je me permettrai de rester non exhaustif.

Merci aux collègues, amis, du LASTIG : Laura Wenclik, Bérénice Le Mao, Alexane Nghien, Solenn Tual, Paul Bourcier, Justin Berli, Florent Geniet, Chahine-Nicolas Zede, Emmanuel Cledat, Ines Lebrun Moro, Youdas Bedhouche.

Merci à Freddi-Jeanne Richard sans qui ce stage n'aurait jamais pu avoir lieu. Merci à Guillaume Touya de m'avoir fait confiance pendant ces quatre mois. Merci à l'IGN et à l'ENSG pour leur accueil.

Merci à l'ESILV de m'avoir permis d'effectuer ce stage et mon année sans jamais avoir à interrompre mon cursus artistique au Conservatoire de Paris.

Merci à Emmanuelle Oltra pour sa relecture attentive.

Merci à mes ami·e·s, ma famille et mes deux colocataires pour m'avoir soutenu et accompagné durant toute cette période.

Table des matières

<i>Glossaire</i>	ii
<i>Acronymes</i>	v
1 Contexte Professionnel	5
1.1 État de l'art	5
1.1.1 Étude du secteur économique	5
1.1.2 Analyse de la compétition et orientation du secteur	6
1.1.3 Engagement DD-RSE	7
1.2 Positionnement de l'IGN au sein de son secteur d'activité	8
1.2.1 Caractéristiques différenciantes	8
1.2.2 Segments de marché	8
1.2.3 Stratégie de développement	8
2 Présentation du stage	11
2.1 Contextualisation	11
2.2 Problématique	12
2.3 Objectif du stage	13
2.4 Plan de route	13
3 Déroulement du stage	17
3.1 Arrivée dans l'entreprise	17
3.2 Méthodologie de travail et collaboration	18
3.3 YOLOv10	19
3.4 Deep SORT	26
3.5 Byte Track	26
4 Bilans	31
4.1 Compétences	31
4.2 Difficultés	32
4.3 Réussites majeures	32
5 Ouverture	35
5.1 Analyse critique du stage	35
5.2 Plan de carrière	35
5.3 Conclusion	36

Annexes

A Cisor2000	39
B Script ROI	43
C YOLO Extractor	45
D Benchmark YOLO	47
E Matrice de corrélation	56
F Script de tracking	59
<i>Bibliographie</i>	66

Glossaire

anthropocène	L'Anthropocène est une nouvelle époque géologique qui se caractérise par l'avènement des hommes comme principale force de changement sur Terre, surpassant les forces géophysiques. C'est l'âge des humains ! Celui d'un désordre planétaire inédit.
bounding box	Une bounding box est un rectangle tracé autour d'un objet dans une image pour en délimiter précisément la position et la taille. Ce rectangle est défini par des coordonnées, généralement celles des coins supérieur gauche et inférieur droit, et sert principalement à localiser des objets en vision par ordinateur et apprentissage automatique.
datascience	La Data Science est un domaine interdisciplinaire qui combine des compétences en mathématiques, en statistiques, en informatique et en communication pour extraire des informations à partir de données brutes.
deep SORT	Il s'agit d'un algorithme de suivi d'objets en temps réel qui améliore le système SORT (S imple O nline and R ealtime T

French Tech	La mission French Tech, ou simplement French Tech, est une administration publique française chargée de soutenir la structuration et la croissance de l'écosystème des start-ups françaises, en France et à l'étranger.
géo-commun	Données et ressources géographiques produites collectivement par une communauté d'acteurs hétérogènes et gouvernée par des règles qui assurent leur survie et garantissent leurs caractères ouverts et partagés.
géodésique	Relatif à la géodésie, la science qui mesure et étudie la forme, les dimensions et le champ gravitationnel de la Terre.
géomatique	La géomatique regroupe l'ensemble des outils et méthodes permettant d'acquérir, de représenter, d'analyser et d'intégrer des données géographiques. La géomatique consiste donc en au moins trois activités distinctes : collecte, traitement et diffusion des données géographiques.
géoplatforme	Plateforme numérique facilitant la collecte, le traitement, l'analyse et la visualisation de données géospatiales.
IGNfab	L'IGNfab est l'accélérateur de start-up mis en place par l'IGN et qui propose aux entrepreneurs d'accélérer le développement de projets innovants mobilisant la description du territoire et la géolocalisation à travers un appui apporté en expertise et en mise en réseau institutionnel et partenarial.
LiDAR	Technique de télédétection utilisant un laser pour mesurer des distances précises, souvent utilisée pour la cartographie 3D et l'analyse topographique.
machine learning	Le machine learning est une technique d'IA qui permet à un programme d'apprendre à partir de données sans être explicitement programmé. Il analyse des exemples pour en extraire des modèles, puis utilise ces modèles pour prédire ou classer de nouvelles données. Il existe trois grands types : supervisé, non supervisé et par renforcement.
open source	Modèle de développement logiciel où le code source est librement accessible, modifiable et redistribuable par tous.
pipeline	Un pipeline désigne une suite structurée et organisée d'étapes ou de processus permettant de transformer une idée en un livrable ou un résultat exploitable.

- éco-responsabilité** Engagement à adopter des pratiques respectueuses de l'environnement pour réduire son impact écologique.
- écophyto** Le plan Écophyto est une stratégie gouvernementale française de réduction d'usage, agricole et non agricole, des pesticides.

Acronymes

CLAHE	contrast limited adaptive histogram equalization.
CNN	réseau neuronal convolutif – Convolutional Neural Network.
DD	développement durable.
ENSG-Géomatique	école Nationale des Sciences Géographiques et de la Géomatique.
ESILV	École supérieure d'ingénieur Léonard de Vinci.
FFRP	fédération française de la randonnée pédestre.
GDH	GeoDanceHive.
IA	intelligence artificielle.
IGN	Institut national de l'information géographique et forestière.
INRAE	institut national de la recherche pour l'agriculture, l'alimentation et l'environnement.
LASTIG	Laboratoire en sciences et technologies de l'information géographique.
OCC	objets connectés & cybersécurité.
OFB	office française de la biodiversité.
RSE	responsabilité sociétale des entreprises.
RSU	rapport social unique.
SE	systèmes embarqués.
WSL	windows subsystem for linux.

Abstract

À l'origine, rien ne me prédestinait à travailler dans le domaine de l'intelligence artificielle. Passionné par les systèmes embarqués et l'informatique de terrain, j'avais une aversion totale pour l'IA et tout ce qui s'en rapprochait. Je voyais dans l'apprentissage automatisé une boîte noire souvent déconnectée du réel, qui manquait de concret pour me susciter de l'intérêt. Lors de ma recherche de stage, après plus de 120 candidatures restées muettes, seules deux propositions me sont parvenues : l'une, en start-up dans l'aéronautique, pour un stage en systèmes embarqués, complètement alignée avec mes centres d'intérêt et qui plus est mieux rémunérée ; l'autre, à l'Institut national de l'information géographique et forestière (IGN), dans un cadre de recherche appliquée au développement d'un modèle d'IA.

Le choix ici paraissait évident pour quiconque avait eu l'occasion d'être à proximité de moi en cours de *machine learning*. Néanmoins, c'est bien l'IGN que j'ai choisi. Pourquoi ? Parce que ce stage faisait le lien avec mes valeurs, bien plus fortes que n'importe quelle technologie. Effectivement, sa connexion intrinsèque avec les enjeux environnementaux et la préservation des pollinisateurs résonnait profondément avec mes convictions écologiques. Sans compter le fait que ce stage m'offrait la possibilité de conserver la dualité de mon parcours en m'accordant la possibilité d'aménager mon emploi du temps pour continuer mes cours au conservatoire.

Ce rapport retrace ainsi l'histoire d'un virage inattendu : celui d'un étudiant amoureux des systèmes embarqués, qui, au service de l'écologie, a plongé dans ses tourments, avec ses doutes et ses découvertes. Une chose est sûre, cette rencontre, aussi brutale soit-elle, aura laissé des traces qui marqueront peut-être le début d'une improbable histoire d'amitié.

Mots-clés : Intelligence artificielle, Écologie, Remise en question

1.

Contexte professionnel du stage

1.1 État de l'art

1.1.1 Étude du secteur économique

L'institut national de l'information géographique et forestière (IGN) occupe une place centrale dans le secteur de la géomatique en France, avec une mission de service public axée sur l'acquisition, la production, l'utilisation, l'enseignement et la diffusion de données géographiques et forestières souveraines. Son positionnement, ses acteurs, ses produits et services ainsi que la dynamique concurrentielle du secteur témoignent d'une transformation profonde à l'ère du numérique et de l'anthropocène [14, 28].

Description des acteurs du secteur

L'IGN est l'acteur public de référence de la donnée géographique en France. Placé sous la tutelle du ministère chargé de l'environnement et de la transition écologique, il travaille en étroite collaboration avec de nombreux ministères, opérateurs de l'État, collectivités territoriales et locales, mais aussi avec des organismes scientifiques, des entreprises et des acteurs de la société civile. Ces dernières années, son écosystème s'est élargi, intégrant des partenaires publics et privés dans le cadre de programmes collaboratifs, d'innovation ouverte (IGNfab) [19] et de la démarche des « géo-communs » qui vise à co-construire des solutions avec l'ensemble des parties prenantes du territoire (citoyens, publics, privés, associatifs).

À côté de l'IGN, plusieurs entreprises privées et éditeurs de cartes agissent sur des segments spécifiques du marché, notamment Michelin, Blay-Flodex, Didier Richard, ainsi

que des associations comme la fédération française de la randonnée pédestre (FFRP). Enfin, les géants du numérique (GAFAM¹), avec Google Maps en tête, constituent une concurrence majeure sur le volet des applications cartographiques et de la géolocalisation grand public.

Produits et services proposés

L'IGN propose une gamme étendue de produits et services qui s'articulent autour de plusieurs axes :

- **Production et diffusion de données géographiques** : cartographie topographique, bases de données géolocalisées, modèles 3D du territoire (notamment grâce à la technologie LiDAR), données forestières, référentiels géodésique et géographiques nationaux.
- **Services numériques** : applications cartographiques (comme l'application *Cartes IGN*), géoportails, plateformes de partage de données (Géoplatforme), services d'observation du territoire (ex : SONEL²).
- **Appui aux politiques publiques** : accompagnement des collectivités et de l'État pour l'aménagement du territoire, la gestion des risques, la transition écologique, la biodiversité, l'agriculture, la forêt, etc.
- **Innovation et partenariats** : soutien à la French Tech, appui aux startups via IGNfab, développement de solutions alternatives aux applications des GAFAM, animation de la communauté des « géo-communs ».

L'IGN a fait évoluer son modèle économique vers la gratuité de la plupart de ses données, renforçant ainsi son rôle d'acteur de l'écosystème national de l'information géographique.

1.1.2 Analyse de la compétition et orientation du secteur

Le secteur de la géomatique est marqué par une concurrence à plusieurs niveaux. Sur le marché des cartes papier et des guides, Michelin reste le leader historique, notamment sur les cartes routières et les atlas (avec plus de 80% de parts de marché sur certains segments³), tandis que l'IGN détient une position dominante sur les cartes de randonnée (96 % de parts de marché en 2002 pour les cartes de randonnée⁴).

Sur le marché des applications numériques, la concurrence est mondiale et dominée par les plateformes des GAFAM, en particulier Google Maps, qui impose ses standards en matière de géolocalisation et de cartographie interactive.

Face à cette concurrence, l'IGN a choisi de se positionner comme un opérateur public garant de la qualité et de la souveraineté des données géographiques, misant sur la gratuité, l'ouverture et la collaboration. Sa stratégie repose sur trois axes principaux : l'ob-

1. Google, Apple, Facebook, Amazon et Microsoft

2. Système d'observation du niveau des eaux littorales

3. d'après *lemag-ic.fr*

4. d'après la Décision n° 06-D-23 du 21 juillet 2006 relative à la situation de la concurrence dans les secteurs de l'édition cartographique et de l'information touristique[8]

servation continue du territoire, l'animation de la communauté des « géo-communs » et la valorisation de la cartographie comme outil de médiation et d'émancipation citoyenne. L'IGN investit fortement dans l'innovation (intelligence artificielle, datascience, modélisation 3D) et dans la transformation numérique de ses services pour répondre aux nouveaux besoins des politiques publiques et de la société.

Le secteur est donc en pleine évolution, avec une orientation marquée vers l'ouverture des données (l'open source), la collaboration public-privé, l'innovation technologique et la réponse aux enjeux écologiques et sociétaux. L'institut, par sa mission de service public et sa capacité à fédérer un écosystème, reste un acteur clé de cette transformation, tout en devant composer avec la concurrence des acteurs privés et des plateformes internationales.

1.1.3 Engagement DD-RSE

L'entreprise affiche un engagement fort en matière de développement durable (DD) et de responsabilité sociétale des entreprises (RSE), qui s'inscrit au cœur de sa stratégie et de ses missions. Cet engagement se traduit par plusieurs axes d'action concrets, aussi bien en interne qu'au service de la société.

L'IGN a intégré la RSE dans sa gouvernance, en menant une politique ambitieuse en matière de qualité de vie au travail et d'éco-responsabilité. Cela se manifeste notamment par une attention particulière portée à l'amélioration des conditions de travail, à la santé et à la sécurité des agents, ainsi qu'à la promotion du dialogue social et de l'égalité professionnelle. L'établissement publie régulièrement un rapport social unique (RSU) qui recense et analyse les données relatives à l'emploi, la formation, la rémunération, la santé au travail et l'action sociale, afin d'accompagner les transformations et de garantir le bien-être de ses personnels.

Sur le plan environnemental, l'IGN joue un rôle clé dans la transition écologique grâce à ses projets en développement utilisant notamment l'intelligence artificielle. Par exemple, l'établissement participe à la cartographie des habitats naturels et au suivi de l'artificialisation des sols, contribuant ainsi à la connaissance et à la préservation de la biodiversité. Ces initiatives permettent d'appuyer le pilotage des politiques publiques et d'accompagner les acteurs des territoires dans la mise en place de pratiques plus durables.

L'ouverture de toutes les bases de données de l'IGN depuis 2021, ainsi que le développement de nouveaux référentiels géographiques, illustrent aussi la volonté de l'établissement de favoriser la transparence, la collaboration et l'innovation au service de l'intérêt général. L'ensemble de ces actions témoigne d'une démarche globale et structurée, qui place le développement durable et la responsabilité sociétale au centre de la mission de service public de l'IGN.

1.2 Positionnement de l'IGN au sein de son secteur d'activité

L'IGN occupe une position centrale et différenciée dans le secteur de l'information géographique et forestière en France. Il se distingue par son statut d'opérateur public de référence, placé sous la double tutelle du ministère de l'environnement et du ministère de l'agriculture, ce qui lui confère une légitimité unique auprès des pouvoirs publics et des collectivités territoriales.

1.2.1 Caractéristiques différenciantes

Ses caractéristiques différenciantes résident dans la production de référentiels géographiques de haute qualité, la maîtrise de technologies avancées (satellites, avions, drones, LiDAR, intelligence artificielle), et une expertise reconnue dans la gestion et l'analyse de données multithématisques. L'IGN s'appuie sur une longue expérience, une école interne dédiée à la formation et à la recherche (école Nationale des Sciences Géographiques et de la Géomatique (ENSG-Géomatique)), ainsi qu'un réseau de partenariats avec des entreprises innovantes et des institutions scientifiques.

1.2.2 Segments de marché

L'IGN intervient sur plusieurs segments de marché :

- L'appui aux politiques publiques dans des domaines variés comme la prévention des risques, l'agriculture, la forêt, l'urbanisme, la défense, la sécurité et la gestion des ressources naturelles.
- La fourniture de données et de services géographiques à l'État, aux collectivités, aux entreprises, au secteur de la défense, mais aussi à la communauté scientifique et au grand public.
- Le développement de nouveaux usages autour de la donnée géographique, notamment pour répondre aux enjeux du changement climatique, de la biodiversité et de l'urbanisation.

1.2.3 Stratégie de développement

Sa stratégie actuelle vise à s'adapter aux grands bouleversements écologiques et numériques en :

- Accélérant l'observation continue du territoire et la production de cartes thématiques sur des enjeux majeurs (forêts, artificialisation des sols, érosion, etc.).
- Valorisant les « géo-communs » et en animant une communauté d'utilisateurs autour de la donnée ouverte et de la médiation cartographique.
- Investissant massivement dans l'intelligence artificielle, la data science, et la formation de nouveaux talents pour accompagner la transformation numérique et la création d'un « jumeau numérique » du territoire français.

- Renforçant la confiance et la fiabilité de ses produits, grâce à une expertise historique et une adaptation constante aux besoins des utilisateurs, en particulier les porteurs de politiques publiques.

Face à la concurrence des grands acteurs privés internationaux, l'IGN mise sur la souveraineté de la donnée, la qualité scientifique et la mission de service public pour conserver son rôle de référence nationale et européenne dans l'information géographique.[25, 3, 16]

2.

Présentation du stage

2.1 Contextualisation

S'il y a bien une chose que je ne soupçonnais pas avant ce stage, c'est à quel point les abeilles sont bavardes. Enfin, à leur manière. Car oui, ces petits insectes sont capables de communiquer entre elles de façon étonnamment sophistiquée, en indiquant à leurs congénères où se trouvent les bonnes ressources à butiner. Leur outil ? Une sorte de chorégraphie millimétrée qu'on appelle la *danse frétillante*, ou *waggle dance* en anglais. Leur interprétation représente une aide précieuse pour les biologistes, les écologues, les apiculteurs et, plus largement, tous ceux qui s'inquiètent (à juste titre) du déclin des polliniseurs. C'est justement dans cette optique qu'est née la GeoDanceHive (GDH), un dispositif d'étude aussi fascinant que complet.

Ce projet, porté par Sylvain Galopin, est issu d'une thèse financée par le Ministère de l'Agriculture, suivie d'un projet financé l'office française de la biodiversité (OFB) dans le cadre du plan Écophyto, le projet DANSE¹. La thèse est réalisée sous la direction de Guillaume Touya (IGN) et Freddie-Jeanne Richard (INRAE). Ensemble, ils ont imaginé une ruche un peu particulière : la GeoDanceHive, une ruche expérimentale pensée pour observer sans déranger [13, 12].

Concrètement, il s'agit d'un assemblage de deux ruches : l'une classique, et l'autre vide servant de "studio", où l'on peut filmer les abeilles sur le premier cadre, sans les perturber. Ce "studio" embarque des caméras haute définition, un éclairage LED UV, et un mini-ordinateur équipé d'un algorithme de vision par ordinateur, capable de détecter et d'analyser ces fameuses danses².

Tout ça, pour quoi faire ? Pour mieux comprendre les dynamiques de butinage, identi-

1. C'est un prolongement et une aide financière pour aller au-delà de la GéoDanceHive monitorée par Freddie-Jeanne Richard et Guillaume Touya

2. les waggle dances

fier les fleurs réellement visitées, et obtenir des données fines sur la manière dont les abeilles interagissent avec leur environnement. En somme, c'est un outil au service du suivi écologique, de la protection des polliniseurs, et de l'amélioration des pratiques agricoles.

Mais pour bien saisir l'utilité de tout ça, il faut plonger un instant dans le monde de la communication chez l'abeille. Au-delà de leur petite taille, ces insectes ont mis en place un système de transmission d'informations très élaboré.

En effet, quand une butineuse trouve une source de nourriture intéressante, elle revient à la ruche et exécute, dans l'obscurité et au contact de ses consœurs, une danse qui varie en fonction de la distance de la ressource. Ici, nous nous focaliserons sur celle en forme de 8, au centre de laquelle elle frétille en ligne droite. Ce frétinement encode deux informations clés :

- la direction de la ressource (par rapport au soleil),
- la distance à parcourir (estimée par la durée du frétinement).

Oui, c'est un langage symbolique³, chez un animal non humain. Ce n'est pas rien (!).

Mais ce n'est pas tout : elles utilisent aussi des phéromones, des vibrations, des contacts physiques, voire des échanges de nourriture pour se transmettre des signaux d'alerte, d'organisation ou d'identification. C'est une organisation eusociale⁴ incroyablement bien rodée, où tout semble pensé pour assurer la survie collective.

C'est donc dans ce contexte (entre biologie comportementale, écologie, et nouvelles technologies) qu'intervient la GeoDanceHive. Et c'est dans ce cadre, aussi surprenant que stimulant, que j'ai réalisé mon stage.

2.2 Problématique

Il suffit d'un simple coup d'œil sur une carte de ruchers pour comprendre une chose évidente : les abeilles ne vivent pas à côté de centres de calcul. En général, elles préfèrent les champs, les lisières, les zones isolées... Là où ni fibre optique, ni prises électriques, ni GPU⁵ dernier cri ne sont à portée de main.

Et pourtant, la GeoDanceHive repose sur une analyse vidéo complexe, qui, dans sa version actuelle, nécessite environ quatre minutes de calcul pour dix minutes de vidéo, le tout sur un ordinateur classique. Autant dire que, dans l'état, aucune chance de l'embarquer au fond d'un champ sans un câble Ethernet de deux kilomètres. Le défi est donc vite posé.

3. système de communication utilisant des symboles pour exprimer des idées, concepts ou émotions, dépassant le langage conventionnel

4. il s'agit du niveau d'organisation le plus complexe du règne animal

5. élément de calcul assurant les fonctions de calcul d'image dans les ordinateurs

Problématique

Comment faire tenir cette intelligence dans une ruche autonome ?

Comment améliorer la qualité de nos détections, tout en ouvrant la possibilité de détecter d'autres comportements que la danse frétilante ?

2.3 Objectif du stage

C'est donc dans le sillage direct de cette problématique que s'est dessiné le sujet de mon stage. Face à un algorithme trop lourd pour être embarqué tel quel, il devenait indispensable de réfléchir à des pistes d'allègement, d'optimisation, et même d'amélioration. Mon rôle ? Explorer ces pistes, les tester, et voir jusqu'où je pouvais aller.

Ainsi, ma mission lors de ce stage était de développer un modèle d'apprentissage profond capable de détecter certains comportements spécifiques des abeilles comme les danses frétilantes. Du moins, c'est l'intention que j'avais en amorçant mes premiers jours de travail. Les objectifs ont néanmoins été très rapidement réajustés par Guillaume⁶, qui m'a expliqué que patience et longueur de temps étaient de rigueur dans la recherche. Il était effectivement utopique d'espérer aboutir à un modèle complètement fonctionnel en seulement quatre mois. C'est pourquoi l'une de mes premières tâches fut d'entamer des recherches préliminaires, après un état de l'art sur le projet, dans le but de dresser un plan de route et des objectifs réalistes.

Néanmoins, à plus large échelle, ce projet permettrait aussi et surtout le développement de véritables laboratoires d'étude portables, nous accordant la possibilité d'étudier divers comportements et réactions des abeilles, dans l'objectif de leur préservation, notamment face aux pesticides.

2.4 Plan de route

Le 7 mai 2025, après deux semaines de recherche et d'assimilation du projet, deux directions s'offraient à moi. Reprendre et adapter un modèle existant ou développer un modèle spécifique. C'est alors que je fus d'abord intéressé par DeepLabCut. Cependant, je finis par le mettre de côté à la suite d'échanges avec Guillaume car ce modèle montrait son efficacité pour la recherche lorsqu'il était utilisé sur des populations de faibles densités. Or nos cadres possèdent une population importante d'individus par image. Notre intérêt c'est alors porté sur le travail d'un laboratoire japonais. Le modèle se prénomme YOWO [7]. Il consiste en l'agrégation d'un YOLO et d'un réseau de neurones permettant le suivi temporel. Néanmoins, leur travail étant toujours à l'état de préliminaires peu documentées, nous décidions de poursuivre vers un développement spécifique. C'est ainsi que nous choisissions d'entamer les travaux par l'adaptation de YOLOv10⁷.

6. Guillaume Touya – mon maître de stage

7. cf. chapitre 3

Détaillons brièvement l'anatomie de ce pipeline. Dans cette forme, il s'agit de la version initiale. Nous verrons par la suite que celui-ci a légèrement évolué au fil des semaines.

Étape 1 – YOLO Abeilles

Ici, nous travaillons image par image, chaque image étant indépendante des autres. L'idée de cette première étape est de détecter le plus d'abeilles possible, individuellement, sans tenir compte de leur position ni de leur comportement. Le but est de réaliser une détection de masse, qui sera ensuite discrétisée à travers les prochaines étapes.

Pour cela, je prévois d'utiliser un modèle d'apprentissage profond déjà entraîné sur une grande quantité d'images, à savoir YOLOv10. L'enjeu est ici de parvenir à ré-entraîner (fine-tuner) le modèle pour qu'il reconnaisse les abeilles de la Geo-DanceHive.⁸

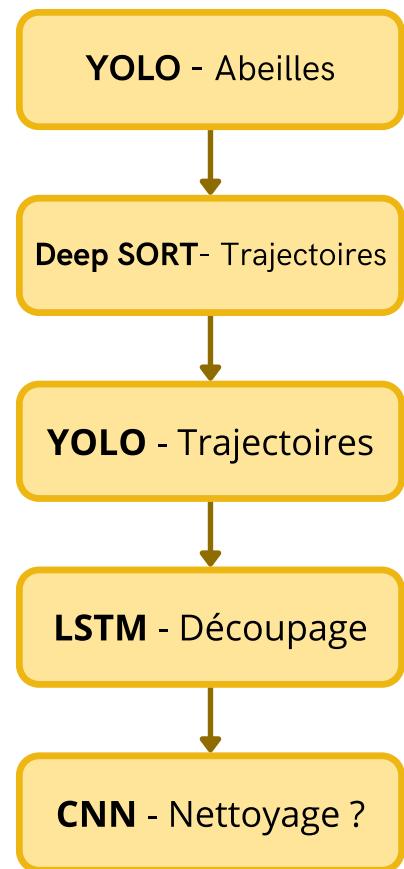


FIGURE 2.1 – Pipeline retenu

Étape 2 – Deep SORT

Une fois les individus discrétisés, l'objectif est d'utiliser Deep SORT⁹ afin de suivre leur trajectoire. Ainsi, chaque pollinisatrice aurait un identifiant unique, ce qui nous permettrait d'installer une logique de suivi à travers les frames¹⁰ d'une vidéo, donc de tracer/récupérer les coordonnées de leurs déplacements.

Étape 3 – YOLO Trajectoires

À partir des coordonnées extraites via le suivi Deep SORT, il devient possible de reconstruire visuellement les trajectoires des abeilles sous forme d'images binaires (noir et blanc), où les tracés représentent leurs déplacements au fil du temps. Ce type de représentation simplifiée permettrait de repérer certaines formes caractéristiques, notamment la fameuse trajectoire en « 8 » des danses frétilantes. L'orientation de ces boucles pourrait être exploitée pour déterminer l'angle de la danse, tandis que la longueur du

8. sources : [31, 30, 33, 34, 17, 24, 20, 32]

9. cf. glossaire

10. « image » en anglais

segment central, la phase de frétillement, donnerait une indication sur la distance annoncée, à condition de connaître le nombre d'images sur lesquelles elle s'étale.

Plusieurs stratégies sont alors envisageables : entraîner un modèle comme YOLOv10 pour la reconnaissance des formes de trajectoire, ou recourir à un modèle de segmentation comme U-Net, mieux adapté à la nature binaire des images. Le choix du modèle dépendra de la précision, de la consommation de ressources et du niveau de généralisation attendu.

Quoiqu'il en soit, cette étape vise à produire une première reconnaissance automatique des danses, en extrayant les motifs correspondant aux comportements recherchés. Les résultats obtenus resteront probablement imparfaits à ce stade : bruit, faux positifs, trajectoires incomplètes sont à prévoir.

Étape 4 – LSTM Découpage

Une danse frétillante est répétitive, elle répète la même séquence de nombreuses fois (plus de 20 phases parfois), d'où l'intérêt de découper chacune de ces phases dans une danse. C'est à cette étape que l'usage du LSTM¹¹ (Long Short-Term Memory) prend tout son sens. Ce type de réseau neuronal récurrent est particulièrement adapté au traitement de séquences temporelles, comme celles issues des danses des abeilles. Son rôle est de découper les trajectoires en segments pertinents, en identifiant précisément le début et la fin des communications.

Contrairement aux approches purement visuelles, le LSTM tient compte de la dynamique : accélérations, virages, rythmes, etc. Il apprend à reconnaître des motifs temporels qui caractérisent une danse, même si celle-ci est partiellement masquée ou perturbée par du bruit. On peut lui fournir une séquence de positions (ou leur représentation visuelle) et lui demander de produire un signal binaire indiquant la présence ou non d'une danse à chaque intervalle temporel.

Ce découpage affinerait la détection brute précédente, en filtrant les faux positifs et en localisant précisément les danses effectives dans la vidéo.

Étape 5 – CNN Nettoyage ?

Enfin, le nettoyage. Cette étape reste incertaine : la performance du LSTM pourrait rendre inutile un réseau neuronal convolutif – Convolutional Neural Network (CNN) final. Mais dans l'idée, une fois les séquences candidates extraites et découpées, il resterait à les valider, nettoyer et affiner. Un réseau convolutif¹² pourrait servir de filtre final ou de classeur binaire, confirmant ou non la présence d'une danse frétillante à partir de séquences annotées. L'objectif est surtout de garantir que seules les trajectoires vraiment caractéristiques soient conservées pour l'analyse finale.

11. type de réseau de neurones récurrent conçu pour apprendre et mémoriser des dépendances à long terme dans des séquences de données grâce à des mécanismes de portes qui régulent le flux d'information.

12. type de réseau de neurones conçu pour extraire automatiquement des caractéristiques locales dans des données structurées, comme des images.



En somme

Nous avons désormais en main les bases scientifiques, techniques et méthodologiques du projet. En recontextualisant les enjeux des danses frétilantes, en présentant la GeoDanceHive et en exposant la problématique d'embarquabilité, nous en avons cerné les contours.

Le pipeline théorique présenté ici illustre l'ambition du projet, mais aussi la vision que j'en avais avant même de démarrer. Il constitue une feuille de route précieuse, sur laquelle s'est appuyée la totalité de mon travail.

Mais, conçu dans l'enthousiasme des premiers jours, il a été confronté à la réalité du temps, des moyens et des imprévus inhérents à toute recherche.

Dans le chapitre suivant, nous avons l'occasion d'approfondir le déroulement du stage : choix techniques, expérimentations, obstacles, ajustements... et peut-être les surprises qui l'ont rythmé.

3.

Déroulement et avancées techniques

3.1 Chapitre 0

Nous étions le lundi 28 avril, j'étais à la fois stressé et impatient. On m'avait dit d'arriver à 9h, mais moi, toujours en retard et soucieux de faire bonne impression, j'avais pris de l'avance. Résultat : je suis arrivé avec presque 1h d'avance. Le bâtiment s'éveillait lentement au rythme des premiers cafés, tandis que le couloir où j'avais rendez-vous somnolait encore dans la pénombre. J'ai alors attendu, assis, devant cette pièce dont les murs laissaient tout transparaître de l'intérieur, sans même savoir que cela deviendrait mon bureau. C'est Laura¹, ingénierie d'étude au laboratoire depuis quelque temps déjà, qui avait été missionnée par Guillaume (en télétravail), pour m'accueillir en ce premier jour de stage.

Si pour elle cela devait être une intégration parmi tant d'autres, pour moi, c'était une journée charnière, à la fois importante et décisive. Importante car c'est durant cette matinée que j'ai eu le droit à la visite guidée de ce qui devint mon quotidien pour les quatre prochains mois. Et décisive car dès les premières minutes j'étais convaincu d'être intégré dans une équipe souriante, agréable et dynamique. Que rêver de plus pour un premier stage ?

J'ai alors très rapidement pris place dans mon bureau tout de verre vêtu avec Paul² à ma gauche et Justin³ à ma droite ; enfin cela à l'heure où j'écris ce rapport. En réalité, lorsque je suis arrivé, j'étais dos à eux, face à un couloir où les regards ne font que se croiser. Cependant, j'ai très vite été invité à me joindre plus encore à eux. Ils m'ont alors

1. Laura Wenclik

2. Paul Bourcier – ingénieur d'étude

3. Justin Berli – ingénieur d'étude

aidé à m'installer, à commander le matériel et à faire les démarches logistiques nécessaires à ma mobilité. Même si j'en étais un, je ne me sentais pas stagiaire. À l'image de Paul et Justin, tout le monde dans cette équipe m'a dès le début fait sentir comme un collègue à part entière.

Je passerai rapidement sur les détails rocambolesques de mon installation informatique... quoique. Ce fut presque romanesque : d'abord aucun ordinateur disponible, puis une machine qui, une fois en main, écrivait toute seule (!). Finalement, après plusieurs ajustements, il fut décidé que je travaillerais sur une machine Linux dédiée, encore en transit depuis un autre site. En attendant, on m'attribua un PC bureautique temporaire, inadapté aux calculs nécessaires à mes tâches, et que je devais bientôt restituer. Résultat : je me suis vite replié sur mon ordinateur personnel, le temps que ma station de travail arrive. Ce qui devait prendre deux semaines mais qui s'est transformé en un mois et demi.

Bien que peu facilitant, cet épisode kafkaïen aura au moins eu le mérite d'ajouter une touche de suspense à mes débuts de stage. Malgré ces aléas matériels, j'ai pu commencer à travailler sans attendre, puisque ma première mission consistait à effectuer un état de l'art approfondi. Armé de Python et de mon fidèle moteur de recherche, j'ai consacré une bonne semaine et demie à explorer la littérature sur le projet *Dance*, le fonctionnement physique et algorithmique de la GeoDanceHive, ainsi que les travaux de Sylvain Galopin. Cette recherche m'a même permis d'identifier quelques éléments complémentaires que j'ai pu transmettre pour enrichir l'état de l'art de la thèse de Sylvain. C'est aussi lors de ces premiers jours que j'ai mis en place une rigueur d'organisation qui me manquait jusque-là. Étudiant souvent épargillé, j'ai pris le parti de structurer strictement mon travail, conscient que c'était une condition sine qua non pour progresser efficacement dans ce contexte de recherche exigeant.

3.2 Méthodologie de travail et collaboration

Cette organisation a reposé sur un outil que je n'oublierai pas de mentionner : **Obsidian**. Déjà familier avec ce logiciel utilisé comme simple éditeur de texte, j'ai découvert à travers ce stage toute sa puissance. Sa vue graphique, précieuse pour structurer le projet, et son espace de toile, idéal pour articuler mes idées, m'ont été indispensables. Les notes quotidiennes automatisées ont instauré une rigueur nouvelle, et son intégration à Git⁴ a facilité la portabilité et la sauvegarde. Gratuit et open source, ce logiciel est pour moi incontournable.

À cette organisation personnelle s'est ajoutée la cadence structurante des points hebdomadaires avec Guillaume. Ces échanges réguliers, renforcés à mi-parcours par des réunions d'équipe⁵ plus élargies, ont rythmé mon stage avec efficacité.

Ils m'ont permis d'ajuster ma trajectoire au fil de l'avancement, de partager mes réussites, mais aussi d'exprimer mes doutes ou blocages. Ce cadre d'échange, à la fois bienveillant et stimulant, a grandement facilité mon immersion dans l'univers de la

4. Git est un logiciel de gestion de versions décentralisé

5. réunion avec toute l'équipe LostInZoom de Guillaume Touya

recherche, pourtant nouveau pour moi.

Grâce à cette organisation bien en place, j'ai pu aborder sans encombre la première étape concrète du pipeline : la détection d'abeilles à l'aide de YOLOv10.

Ce fut le début du travail technique à proprement parler, où il ne s'agissait plus de lire ou de structurer, mais de manipuler des scripts, des images, des modèles... Bref, de rentrer dans le cœur de mon stage.

3.3 Détection des abeilles avec YOLOv10

Si le cadre de travail était clair, le chemin pour y parvenir, lui, l'était beaucoup moins. L'objectif était de faire « simple » : il fallait « juste » détecter les abeilles dans une vidéo. Facile à dire, moins à faire, surtout lorsqu'on sait que ces vidéos sont tournées dans une ruche, dans des conditions lumineuses somme toute peu fidèles et où les abeilles elles-mêmes s'empilent, se fondent dans le décor, disparaissent et réapparaissent comme par magie.



C'est donc dans ce contexte que j'ai entamé mes travaux sur YOLOv10. Le plan initial ? Le fine-tuner sur un petit jeu de données annoté manuellement pour vérifier s'il pouvait repérer nos abeilles sur les cadres de la GeoDanceHive (*cf.* Figure 3.1).

(Spoiler alert : ça n'a pas suffi)

FIGURE 3.1 – Image brute de la GDH

Les débuts chaotiques

L'installation du modèle, bien qu'aidée par une documentation dense, ne fut pas de tout repos. Il fallut d'abord préparer l'environnement de travail : configuration de l'environnement virtuel, installation des bibliothèques, ajustement de la version de Python, abandon de WSL⁶ en route pour cause de sécurité informatique... Une série d'obstacles techniques classiques, mais chronophages, qui m'ont parfois fait regretter le confort d'une simple Arduino. D'autant plus que, bien que cela soit prévu, du fait que j'eusse commencé sans ordinateur technique, je n'avais pas accès au calculateur. J'ai donc lancé mes premiers entraînements de modèle sur le GPU de Google Colab⁷.

Une fois l'environnement installé, les premiers tests ont pu commencer. Lors de mes

6. couche de compatibilité permettant d'exécuter des exécutables binaires Linux de manière native sur Windows

7. Google Colab est un environnement de développement en ligne qui permet d'écrire et d'exécuter du code Python directement dans un navigateur, sans installation locale. Il fournit un accès gratuit à des ressources informatiques puissantes, comme des GPU et TPU

recherches, j'avais trouvé un dataset de plus de 900 images d'abeilles sur RobotFlow⁸. Je me suis alors servi initialement de ce dernier, enrichi de deux images de la GDH annotées à la main. Deux modèles ont été entraînés (sur Google Colab), avec et sans les ajouts personnels, mais les résultats furent décevants : les abeilles n'étaient pas du tout détectées. Les raisons pouvaient être multiples : trop peu de données, trop de bruit, de mauvais hyperparamètres, trop d'abeilles ?

Améliorer le dataset, affiner le modèle

À ce stade-là, pour moi, la suite logique était l'amélioration des images du dataset GDH. Il fallait réussir à améliorer la qualité des frames des vidéos afin d'augmenter la lisibilité et donc le potentiel discernement des modèles. Pour ce faire, j'ai d'abord pensé à de simples modifications de contraste et de luminosité superposées. Toujours dans la même logique, après des recherches et discussions avec Guillaume, j'avais une idée d'un workflow⁹ à suivre pour augmenter la résolution de mes images. J'ai donc une nouvelle fois abouti à un pipeline (*cf.* Figure 3.2). [11, 21]

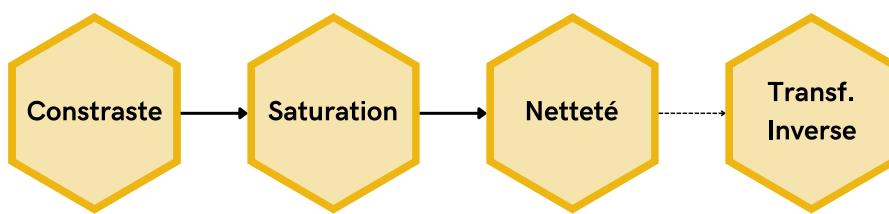


FIGURE 3.2 – Pipeline initial de traitement d'image

J'ai alors commencé par créer un script prenant des images au hasard dans un dossier contenant toutes les frames de la vidéo de travail (obtenue aussi à partir d'un autre script) pour pouvoir leur appliquer les mêmes modifications. [26]

Néanmoins, pour aboutir à une combinaison optimale d'étapes améliorantes, il a fallu que je procède à de nombreux tests et explorations afin de déterminer ce qui méritait mon attention ou pas et surtout afin de fixer les bonnes valeurs des différents paramètres.

À ce moment-ci, je pensais avoir une amélioration optimale de mes images (*cf.* Figure 3.3).

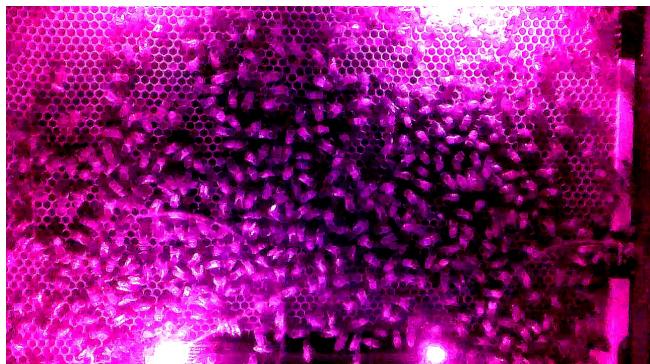


FIGURE 3.3 – Amélioration initiale

8. Plateforme répertoriant des modèles et des jeux de données destiné à des réseaux de neurones

9. synonyme de pipeline

Il ne fallait donc plus qu'à la développer sur un plus grand nombre d'images de la GDH. J'avais alors besoin d'un grand nombre d'images annotées, or seulement deux m'avaient pris presque une après-midi entière. C'est ici que je saisis l'intérêt d'avoir intégré une équipe de recherche. J'ai donc mobilisé tout le potentiel du laboratoire et ai demandé l'aide de toute l'équipe.

Grâce à Roboflow et sa plateforme en ligne, nous avons pu procéder à une annotation collaborative des images de la GDH. Initialement plein de bonne volonté, j'avais prévu un set de 100 images à annoter à la main. Malheureusement, une seule image prenait presque 30 min à une seule personne pour être entièrement complétée. C'est d'ailleurs pourquoi, de ces 100 images, seules 9 furent complètement annotées. Il fallait trouver une solution pour multiplier le dataset sans multiplier nos efforts. Deux possibilités s'offraient à moi : une augmentation virtuelle ou physique du jeu de données.

C'est ainsi qu'est né le *Cisor2000* (*cf. Annexe A*), un script que j'ai développé capable de découper les images et leur fichier label. De ce fait, une image annotée en donnait neuf. Cela était possible du fait de la densité de population et de la qualité des images. J'avais donc désormais une augmentation physique significative de mon dataset. Pour 9 images, je pouvais en générer 89. [27, 29]

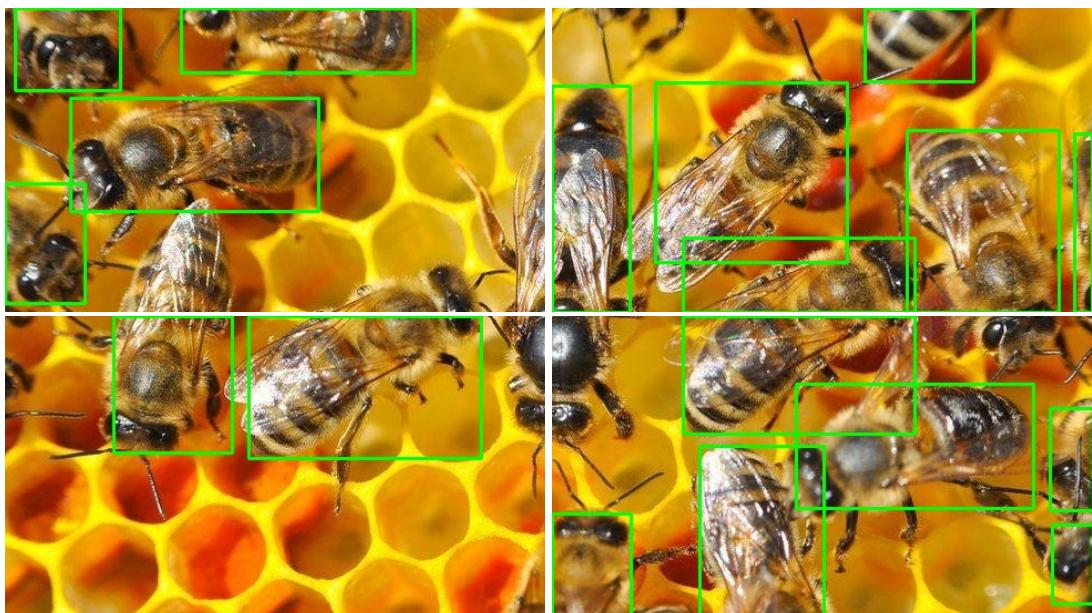


FIGURE 3.4 – Exemple du travail généré par le script *Cisor2000*

Dans les mêmes temps, Guillaume me proposa de copier le dataset original et d'en faire une copie « violetisée¹⁰ ». En effet, nous supposions que cela pouvait aider le modèle que de s'entraîner sur des images ayant la même colorimétrie que les images de la GDH.

J'ai donc appliqué un filtre de couleur, que je jugeais ressemblant aux images de la

10. une copie violette



FIGURE 3.5 – Avant coloration



FIGURE 3.6 – Après coloration

GDH, aux 900 images du dataset original.

Désormais, j'avais un jeu de données de 89 images dont la qualité avait été améliorée (*cf.* Figure 3.3). Cependant, malgré l'amélioration, il restait des zones illisibles. Effectivement, les régions sous les projecteurs UV étaient cramées, de ce fait toutes les abeilles dans ces espaces étaient invisibles. Pour régler ce problème, j'ai d'abord pensé et mis en place une stratégie de ROI¹¹.

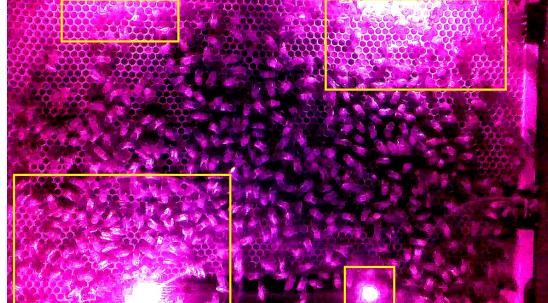


FIGURE 3.7 – Repérage des ROI

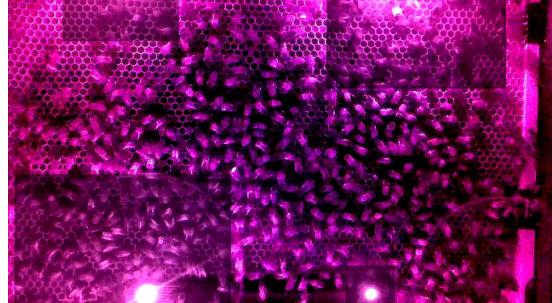


FIGURE 3.8 – Modifications des ROI

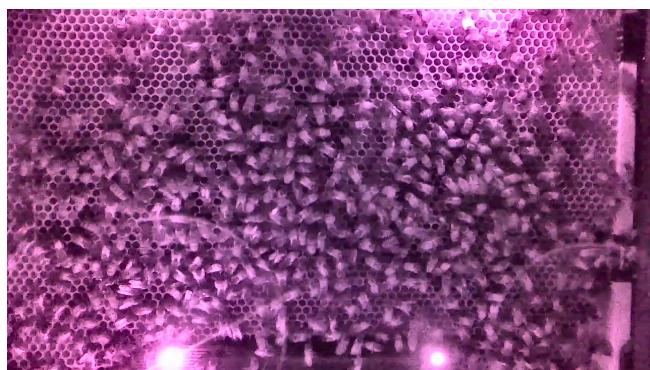


FIGURE 3.9 – Résultat du filtre CLAHE

L'idée était la suivante. Après avoir déterminé les coordonnées précises des régions d'intérêts (cramées) sur Paint (*cf.* Figure 3.7), je n'avais qu'à appliquer des modifications différentes dans ces périmètres dans mon pipeline de traitement (*cf.* Annexe B). Pourtant fonctionnel, les résultats obtenus laissaient apparaître des frontières incisives aux

11. détermination de Region Of Interest ou région d'intérêt

abords de ces ROI.

Lors d'une réunion d'équipe où je présentais ces résultats, Sylvain me fit part de ses doutes quant aux produits obtenus et m'encouragea à me pencher du côté des filtres dynamiques. C'est ainsi que le filtre CLAHE¹² est arrivé dans mon pipeline de traitement d'image. Les résultats étaient sans appel (*cf. Figure 3.9*).¹³

Le pipeline définitif de traitement d'image (*cf. Figure 3.10*) venait d'être re-designé avec l'aide d'un sondage sur Zulip¹⁴ pour m'aider à trancher sur le meilleur résultat.

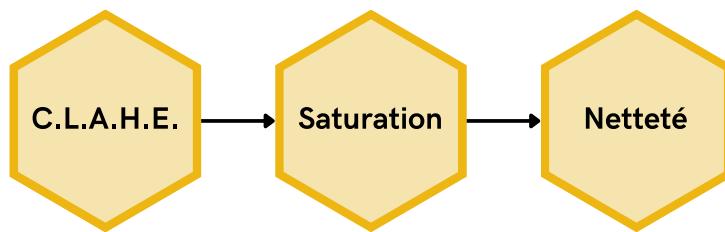


FIGURE 3.10 – Pipeline final de traitement d'image

Maintenant, il fallait trouver une manière de récupérer les images déjà annotées sur Roboflow de façon à ne pas perdre le travail déjà amorcé et surtout de façon à pouvoir le compléter. C'est ainsi qu'est apparu le *YOLOExtractor* (*cf. Annexe C*), un script de scraping, capable d'extraire du code source du site Roboflow, le json des coordonnées des boîtes et de le transformer en fichier label au format Yolo.

Finalement, je n'avais plus qu'à réinjecter mes images découpées, traitées et complétées dans la BDD. Ainsi, d'initialement neuf images annotées, plus de soixante sont apparues. Comme je n'ai pas relancé de phase d'annotation, certaines images peu annotées n'ont pas été intégrées au dataset. [22]



FIGURE 3.11 – Rendu final en sortie de pipeline

12. filtre d'égalisation du contraste dynamique

13. sources : [6, 10]

14. plateforme de communication d'équipe (comme *Teams* mais open source)

Désormais j'ai 1800 images dans ma base de données, il ne reste plus qu'à les tester !

Une première lumière au bout du tunnel

Hourra !! * danse de la joie *

Les premières inférences réussies ont eu lieu début juillet. Mon modèle détectait les abeilles, imparfairement certes, mais de façon cohérente. Quelle satisfaction de voir ces petites boîtes bleues apparaître autour de ces minuscules tâches, dans des images pourtant si denses ! Un vrai tournant dans le stage, confirmé par une série de réunions avec Guillaume et Freddie-Jeanne Richard, qui ont aidé à fixer la marche à suivre : continuer à fiabiliser ce pipeline, tout en gardant en tête qu'il ne s'agit que d'un maillon de la chaîne plus ambitieuse du projet.

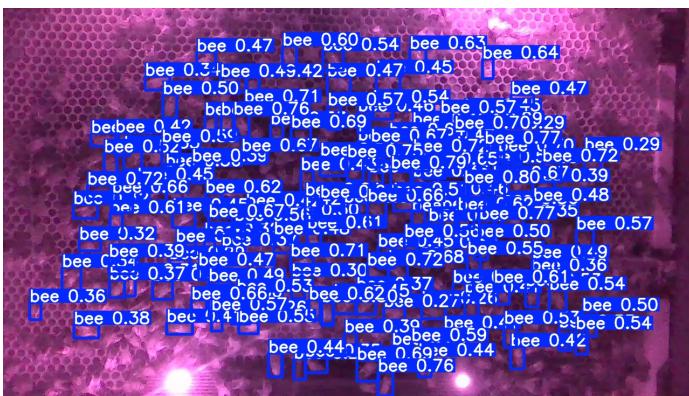


FIGURE 3.12 – Premier résultat d'inférence

Comme mes premiers entraînements (avec le calculateur de l'IGN à travers une connexion ssh¹⁵) étaient encourageants (cf. Figure 3.12), il me fallait désormais déterminer quelle combinaison d'hyperparamètres était la plus optimale. Pour ce faire, j'ai développé un script de benchmark (cf. Annexe D) me permettant de procéder à la même évaluation sur mes différents modèles [2, 4, 15, 35].

J'ai alors pu évaluer 21 modèles entraînés différemment. Grâce à une matrice de corrélation (cf. Figure E) j'ai pu déterminer quels paramètres étaient importants et quel modèle était le plus efficace. Notre grand vainqueur est ainsi le modèle 15 qui présente le meilleur taux de détections confirmées.[1, 36, 18].

	Score total	Precision	Recall	IoU mean	Nb. Detec.	Matched	FP	FP rate	Epochs	Batch	BDD version
Score total	1										
Precision	0.881	1									
Recall	0.915	0.647	1								
IoU mean	0.974	0.954	0.841	1							
Nb. Detec.	0.899	0.630	0.997	0.827	1						
Matched	0.915	0.647	0.971	0.841	0.997	1					
FP	0.846	0.578	0.971	0.779	0.986	0.971	1				
FP rate	0.907	0.650	0.989	0.848	0.988	0.989	0.967	1			
Epochs	0.174	0.187	0.230	0.182	0.225	0.229	0.053	0.185	1		
Batch	0.059	-0.025	0.074	0.015	0.185	0.074	-0.001	0.058	0.716	1	
BDD version	0.745	0.514	0.869	0.676	0.868	0.869	0.850	0.808	0.129	0.093	1

TABLE 3.1 – Matrice de corrélation des métriques des modèles (détails Annexe E)

15. connexion à distance par le réseau IGN



FIGURE 3.13 – Résultat de l'inférence du modèle 15

Et après ?

Cette étape YOLO, bien que semée d'embûches, m'a beaucoup appris. À la fois sur les modèles de détection en temps réel, sur les limites d'un dataset peu fournit, mais surtout sur la patience nécessaire quand on travaille avec des objets vivants, dans un cadre réel, avec tout ce que cela implique de bruit, d'imprévus et d'incertitudes. Et c'est peut-être ça, le plus important.

YOLOv10 m'a permis d'installer la première brique d'un système de détection complexe même s'il ne s'agit encore que du début. Car détecter les abeilles sur une image, c'est une chose. Mais les suivre dans le temps, image après image, pour reconstituer leurs trajectoires et, à terme, tenter d'interpréter leur comportement... c'en est une autre !



FIGURE 3.14

A ce moment-là, il ne s'agissait plus seulement de dire où se trouve une abeille, mais où elle va, d'où elle vient, et combien de temps elle reste. Ce changement de perspective m'a naturellement conduit vers l'étape suivante du pipeline : Deep SORT. Néanmoins, avant de changer de bloc, permettez-moi de vous partager les premiers résultats. Fruits de toutes les étapes précédentes, vous pourrez trouver une courte vidéo issue de l'inférence du modèle 15 sur un extrait (15s) de la vidéo de travail (en scannant ce qr-code : figure 3.14) [23].

3.4 Implémentation du suivi multi-objets

Comme on peut le remarquer, la grande majorité des abeilles sont ici détectées cependant elles ne sont pas identifiées. C'est-à-dire que d'une image (frame) à l'autre, le modèle ne fait pas les liens entre les individus détectés. Il n'est pas conscient que la même détection au même endroit, mais sur la frame d'après, correspond à la même abeille.

De ce fait, il était nécessaire d'intégrer un algorithme de suivi multi-objets. Mes recherches en début de stage laissaient sous-entendre que Deep SORT était la meilleure solution pour cette étape du workflow.

Me voilà alors lancé dans le décorticage de Deep SORT. Sur la base de projets déjà existants. J'ai alors tenté de procéder à des implémentations simples, en copiant simplement des dépôts GitHub.

Mais malheureusement, le modèle était très peu performant. Malgré la tentative de modification de certaines méthodes dans les classes ou même la modification des hyperparamètres ou du filtre Kalman, les résultats n'étaient pas satisfaisants. Comme en témoignent les vidéos générées (*cf.* Figure 3.15), les suivis sont faux, les bbox¹⁶ de détection attachées aux centres malgré les tentatives de correction ou de remplacement par une croix ne sont pas bien placés, et le nombre d'individus détecté est trop peu important par rapport à la réalité (facteur 10). De plus, le modèle mettait plus de 490 s pour générer une vidéo de 15 s (ce qui représente plus de 8 min). Cette inefficience s'explique peut-être dans la définition de la classe objet du tracker¹⁷ ou par la complexité des images, mais je n'ai entamé aucune recherche approfondie pour confirmer ou pas ces hypothèses [9, 37, 39].



FIGURE 3.15

Il était alors nécessaire de se réorienter vers un autre algorithme de suivi pour tenter de parvenir à des résultats plus satisfaisants.

3.5 Abandon de Deep SORT et appréhension de Byte Track

C'est une nouvelle salve de recherches qui m'ont mené à Byte Track. Avant tout, faisons un rapide point sur les différences entre Deep SORT et Byte Track.

Si Deep SORT excelle dans le maintien de l'identité sur de longues séquences grâce aux embeddings visuels, il peut être plus lourd côté calcul et nécessite un détecteur et un extracteur de caractéristiques puissants. Or, dans notre cas de figure, le détecteur (YOLO) ne donne pas que des résultats à forte confiance. ByteTrack, en revanche, est

16. bounding box

17. façon dont est codé un tracker dans les programmes utilisant DeepSort

plus léger et plus rapide, tout en obtenant de meilleurs scores MOTA et IDF1¹⁸ que Deep SORT dans plusieurs études, ce qui défend sa fiabilité malgré l'absence de modèle d'apparence.

Pour faire simple, Deep SORT privilégie une association basée sur l'apparence (en tentant de re-déetecter de son côté) pour garantir la continuité d'identité dans des scènes encombrées, tandis que ByteTrack mise sur une stratégie duale d'association (haute et basse confiance) pour maximiser la récupération des objets en mouvement rapide ou partiellement occultés [38, 5].



FIGURE 3.16
C'est pourquoi ByteTrack offre un excellent compromis entre vitesse et performance.

Alors, de la même façon que pour Deep SORT, j'ai d'abord procédé à une simple implémentation par copie de dépôt, qui s'est avérée encourageante. J'ai alors commencé de plus amples modifications impliquant notamment l'ajustement des hyperparamètres.

Après quelques essais pour ajuster les hyperparamètres, les résultats étaient bien meilleurs que Deep SORT pour un temps d'exécution nettement inférieur (72s).

En relançant les tests avec la vidéo originale dont seule la qualité a été touchée, les résultats étaient très satisfaisants (*cf.* Figure 3.16). ByteTrack parvient à détecter et suivre presque toutes les détections de YOLO. Je me suis alors complètement approprié le script originel (*cf.* Annexe F) en y intégrant plusieurs modes, notamment la possibilité d'afficher ou pas les labels, de générer une couleur différente pour chaque individu ou de modifier le format de la vidéo de sortie (le fond et les infos à afficher).

Néanmoins, malgré la modulation de ces différents modes, les vidéos sont très chargées en information et donc presque illisibles pour l'œil humain. C'est pourquoi j'ai également implémenté un mode capable de générer un fichier json contenant toutes les trajectoires, nous laissant ainsi la possibilité de tracer, isoler, visualiser celles que l'on désire. De ce fait, j'ai extrait les différents id¹⁹ des deux *waggle dance* de la vidéo (à l'aide de l'affichage des labels) et je les ai tracés à l'aide d'un script python.

Ici, on remarque que la figure 3.18 nous donne un résultat très satisfaisant. Effectivement, on voit que le tracé de la courbe bleue dessine un motif remarquable, celui de la fameuse danse en « 8 » (Figure 3.19.C²⁰). C'est sur cette portion de « zig-zag » qu'elle frétille.

18. des mesures d'évaluation en suivi objet

19. numéro d'identification

20. schéma de Sylvain Galopin

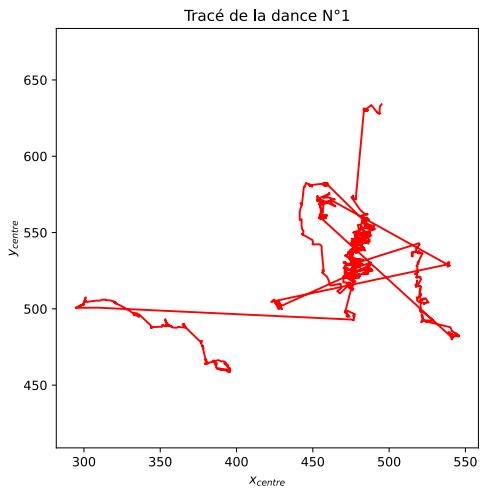


FIGURE 3.17 – Danse n°1

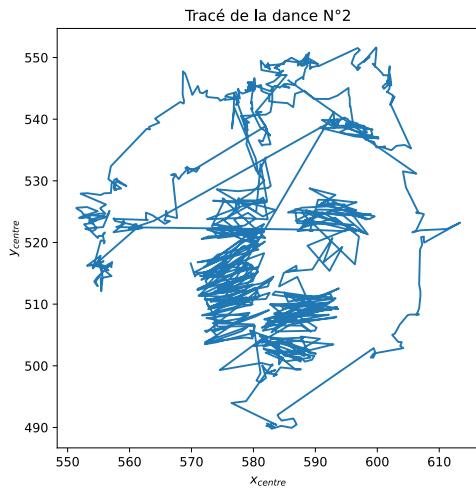


FIGURE 3.18 – Danse n°2

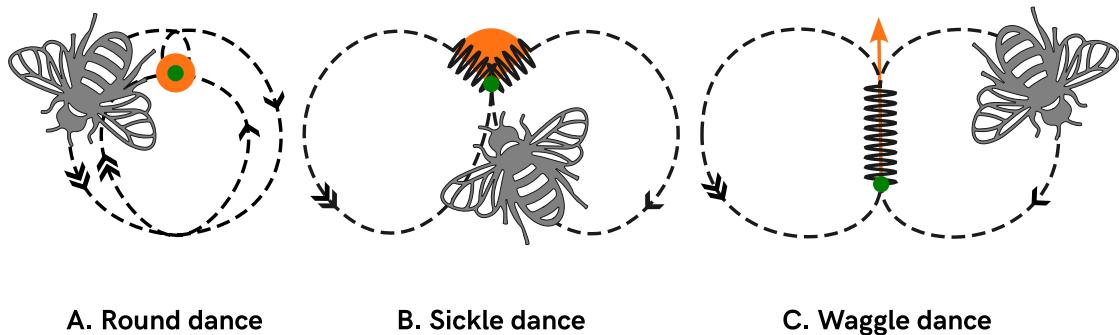


FIGURE 3.19 – Différents motifs théoriques de danse

Comme on peut le remarquer, on retrouve sur le graph (Figure 3.20) associé à la danse deux, le motif de « 8 » et le frétinement. Alors, nous pouvons conclure quant au fait qu'il s'agisse bien d'une *waggle dance*.

Bien qu'imparfait, ce résultat, généré avec la meilleure combinaison d'hyperparamètres, m'a permis de générer une réelle preuve de concept que voici.

POC

Il est possible de détecter, suivre des abeilles et tracer leurs trajectoires bien qu'elles soient sur des images à population très dense à l'aide du Deep Learning.

Tracé de la dance N°2

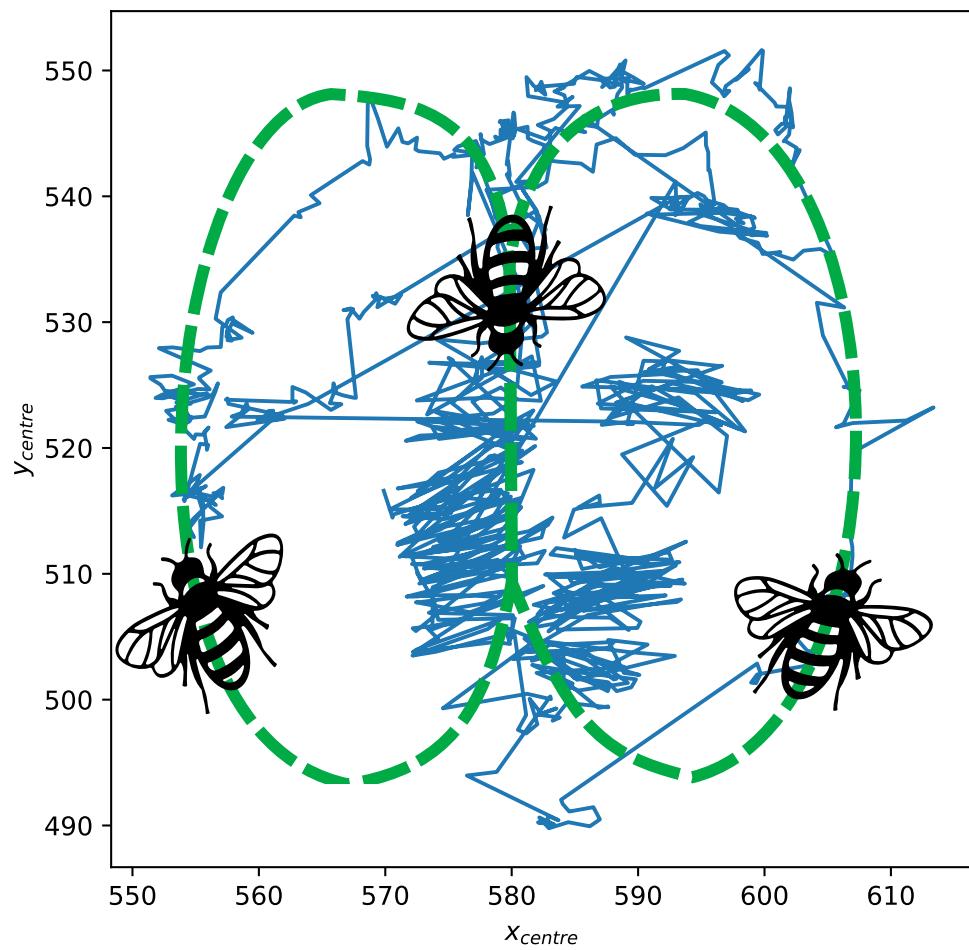


FIGURE 3.20 – Comparaison des motifs

4.

Bilans du stage

4.1 Bilan des compétences

Compétences acquises

Tout au long de ce stage, j'ai pu développer de nouvelles compétences techniques (hard skills), notamment en manipulation et annotation d'images associées à la reconnaissance automatique. J'ai appris à augmenter, découper et nettoyer des datasets tout en faisant garde des différents formats des fichiers labels. L'entraînement et le fine-tuning des modèles YOLOv10 m'ont permis de comprendre en profondeur les mécanismes de détection d'objets. Par ailleurs, j'ai pu implémenter des algorithmes complexes de suivi multi-objets tels que Deep SORT et ByteTrack, tout en intégrant des techniques avancées de traitement d'image comme le filtrage dynamique CLAHE ou la gestion de régions d'intérêt pour améliorer la qualité des données visuelles.

Sur le plan méthodologique et organisationnel (soft skill), ce stage m'a conduit à adopter une gestion rigoureuse de mon travail, grâce notamment à l'utilisation d'outils tels qu'Obsidian qui m'ont aidé à structurer mes prises de notes et à suivre efficacement mon avancée. J'ai aussi pratiqué une progression itérative basée sur des essais-erreurs et des benchmarks réguliers, ce qui m'a permis de comparer différentes stratégies, d'analyser les résultats de manière critique et d'optimiser progressivement mes modèles et réflexions tout en intégrant les retours de mes encadrants.

Enfin, ce projet m'a permis de renforcer des compétences transversales essentielles en recherche. J'ai été contraint d'apprendre à gérer efficacement mon temps et à prioriser mes tâches en dépit des contraintes matérielles et techniques. Je crois aussi avoir développé une capacité d'adaptation et d'apprentissage rapide face à des technologies

et concepts nouveaux. Ce stage m'a confirmé l'importance de conserver une approche persévérente et proactive dans le monde professionnel.

Compétences développées

Néanmoins, grâce à ma formation, j'ai pu mettre en œuvre des compétences d'ores et déjà développées, au service de mes missions. Je pense notamment à l'utilisation d'outils et d'environnements de calcul sur GPU, au développement en Python à travers la création de scripts d'automation et de traitement d'images, ou encore la gestion de version grâce à Git.

Bien sûr, je n'oublierai pas ce qui fait la force de ma formation à l'ESILV, les compétences transversales et les soft skills que j'ai pu mobiliser au travers du travail et de la communication en équipe ou même dans la résolution des problèmes (parfois techniques) complexes.

4.2 Difficultés

Nonobstant, lors de ce stage, j'ai pu recenser deux difficultés majeures. L'impatience. J'avoue ne pas être de nature patiente, mais quand il s'agit de travail, j'arrive, en temps normal, à prendre du recul quant aux objectifs et à adapter mon organisation. Cependant, je n'avais jamais côtoyé le monde de la recherche, alors il a été difficile pour moi d'avancer ou, du moins, d'essayer alors qu'il pouvait m'arriver de rester plusieurs jours à faire la même recherche ou même de rester plusieurs jours bloqué au même endroit en dépit de mes efforts, sans même savoir si je parviendrait à surmonter l'obstacle en question. Ce qui m'amène à vous évoquer ma seconde difficulté.

L'incertitude. Oui, cela paraît peut-être évident, mais on ne s'en rend réellement compte que lorsqu'on la traverse. L'incertitude nous drape d'un voile nous réclamant d'avancer alors que l'on ne voit pas ce qu'il y a à nos pieds. C'est pour moi la définition la plus juste et fidèle de la recherche. Aller dans une direction, convaincu que l'on y trouvera ce que l'on cherche, mais ne pas voir où l'on va et finalement trouver autre chose que ce que l'on cherchait. Engendrant parfois de petites baisses de motivation ou d'intérêt, j'ai pu compter sur mes collègues ou même Guillaume pour me rappeler pourquoi nous faisions tout cela.

4.3 Réussites majeures

En rédigeant ce rapport, et en tentant de prendre tout le recul nécessaire, je suis heureux d'écrire que je suis fier du travail accompli durant ces quatre mois. Je suis content d'avoir réussi à faire preuve d'autonomie lorsque cela était nécessaire et d'être parvenu aussi loin dans la progression de mon pipeline initial. Car, même si mon résultat final n'est pas parfait, il est tout de même encourageant quant au potentiel de cette direction

dans la satisfaction de nos objectifs primaires. De ce fait, l'équipe du projet DANSE, satisfaite des résultats, prévoit de poursuivre dans cette direction et prévoit même d'en produire une publication scientifique.

5.

Ouverture

5.1 Analyse critique du stage

Enfin nous arrivons à la fin de ce long rapport (qui pour ma défense fait bien 30 pages si l'on enlève les couvertures, pages blanches, le glossaire, les acronymes, les copyrights, la bibliographie et les illustrations) (oups).

En tentant de rester sérieux pour conclure, j'écrirai juste qu'il a pour moi été très difficile de condenser, en si peu, 4 mois dont j'aurais voulu vous raconter chaque instant tant je les ai appréciés. J'ai également eu la chance de participer à des projets extra-stage en assistant aux réunions d'équipe LostInZoom et en pouvant réaliser et organiser des expériences pour le projet.

Je profiterais de cette section pour faire une mention spéciale relative à l'ambiance générale de l'IGN où j'ai pu rencontrer des personnes intéressantes, inspirantes, militantes et engagées dans la défense de différentes causes, notamment environnementales. Même le restaurant professionnel proposait quotidiennement et sans exception une alternative végétarienne encourageant aux petits efforts personnels pour l'éologie.

5.2 Plan de carrière

Je dirais que ce stage a chamboulé la vision que j'avais de mon futur tous termes confondus, car il a replacé au centre de mes aspirations, celle de poursuivre dans la recherche.

En effet, je précise tout de même que je souhaitais initialement faire de mon appétence pour les systèmes embarqués mon cheval de combat principal, visant alors de célèbres

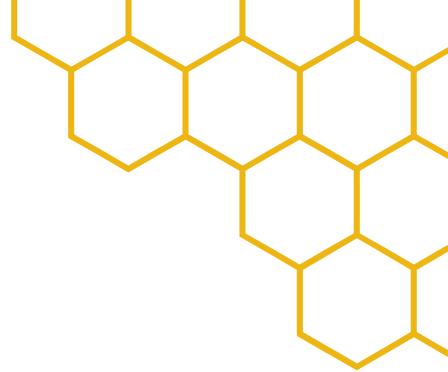
entreprises dans le secteur et me permettant d'annexer tout ce qui s'apparente à de l'apprentissage automatique (IA). Or, oui c'est donc là le dénouement tant attendu (je n'en doute pas), ai-je depuis noué une amitié avec l'intelligence artificielle ?

Et bien, c'est rempli de surprises que j'ai le plaisir de vous écrire que oui, je n'éprouve désormais plus de dégoût ni même de rejet pour cette technologie majeure !

5.3 Conclusion

Pour conclure rapidement, je pense que vous aurez compris que ce stage fut pour moi une expérience positive, marquante et plus que jamais d'actualité. En effet, sans m'étaler dans les détails, au moment où je conclus ce rapport la Loi Duplomb, visant l'autorisation de pesticides évalués dangereux pour les abeilles, vient seulement d'être jugée inconstitutionnelle.

Ce stage fut pour moi une révélation, renforçant ma conviction de m'investir pleinement dans la recherche et les enjeux cruciaux qui l'accompagnent.



ANNEXES



Cisor2000

Script de découpage d'images annotées

L'objectif de ce script est de diviser des images déjà annotées au format YOLO en plusieurs sous-images, tout en recalculant les coordonnées des *bounding boxes* en fonction des zones découpées. Il permet également de filtrer les zones annotées ne respectant pas un seuil minimal de recouvrement (`ratio_min`) (fixé à 0.3 après essais) et de générer de nouvelles étiquettes ainsi que des images avec possibilité de visualisation des détections.

Listing A.1 – Configurations initiales

```
1 import os
2 import cv2
3 import cv2
4
5 INPUT_IMAGE_DIR = 'Test/images'
6 INPUT_LABEL_DIR = 'Test/labels'
7 OUTPUT_IMAGE_DIR = 'Output/images'
8 OUTPUT_LABEL_DIR = 'Output/labels'
9
10 PRINT_BBOX_DIR = 'BBOX30'
11
12 ROWS = 2 # Découpage vertical
13 COLS = 2 # Découpage horizontal
14
15 # Préparation des dossiers de sortie
16 os.makedirs(OUTPUT_IMAGE_DIR, exist_ok=True)
17 os.makedirs(OUTPUT_LABEL_DIR, exist_ok=True)
18 os.makedirs(PRINT_BBOX_DIR, exist_ok=True)
```

On sait que le format des *bounding boxes* est le suivant :

class, x_center, y_center, width, height	(normalisés entre 0 et 1)
--	---------------------------

J'ai alors rédigé une classe qui convertit les bbox originelles en bbox relatives aux découpes.

Listing A.2 – Classe de conversion des bbox

```
1 def convert_bbox(bbox, orig_w, orig_h, x_offset, y_offset, crop_w,
2     crop_h):
3     """
4     Convertit les coordonnées bbox d'origine en bbox relative au crop
5     .
6
7     Args:
8         bbox (tuple[float, float, float, float]): données initiales
9             de la bbox (x_center, y_center, width, height)
10        orig_w (int): largeur de l'image d'origine en px
11        orig_h (int): hauteur de l'image d'origine en px
12        x_offset (int): décalage sur x du crop par rapport à l'image
13            d'origine en px
14        y_offset (int): décalage sur y du crop par rapport à l'image
15            d'origine en px
16        crop_w (int): largeur du crop en px
17        crop_h (int): hauteur du crop en px
18
19    Returns:
20        tuple[float, float, float, float]: données convertit de la
21            bbox relativement au crop
22    """
23
24    cx, cy, w, h = bbox
25
26    # Conversion des coordonnées normalisées en cordonnées absolues (
27    # px)
28    abs_cx = cx * orig_w
29    abs_cy = cy * orig_h
30    abs_w = w * orig_w
31    abs_h = h * orig_h
32
33    bbox_area = abs_w * abs_h      # Aire de la bbox
34    ratio_min = 0.3              # Ratio min pour une bbox valide
35
36
37    x1 = abs_cx - abs_w / 2    # Coin Gauche
38    y1 = abs_cy - abs_h / 2    # Coin Haut
39    x2 = abs_cx + abs_w / 2    # Coin Droit
40    y2 = abs_cy + abs_h / 2    # Coin Bas
```

```

35
36     # Détermination des coupes
37     crop_x1, crop_y1 = x_offset, y_offset
38     crop_x2, crop_y2 = x_offset + crop_w, y_offset + crop_h
39
40     # On détecte si une bbox est partiellement ou entièrement dans le
41     # crop
42     inter_x1 = max(x1, crop_x1)
43     inter_y1 = max(y1, crop_y1)
44     inter_x2 = min(x2, crop_x2)
45     inter_y2 = min(y2, crop_y2)
46
47     if inter_x1 >= inter_x2 or inter_y1 >= inter_y2:
48         return None # Le rectangle d'intersection est vide (la bbox
49         # n'est pas dans le crop)
50
51     # Nouvelle box dans le crop
52     new_cx = (inter_x1 + inter_x2) / 2 - crop_x1 # on fait moins
53     # crop_x1 pour repositionner dans le repère local du crop
54     new_cy = (inter_y1 + inter_y2) / 2 - crop_y1
55     new_w = inter_x2 - inter_x1
56     new_h = inter_y2 - inter_y1
57
58     #print(new_w, new_h, bbox_area)
59
60     if (new_h * new_w) / bbox_area < ratio_min:
61         #print('je suis rentré dans la condition')
62         return None # La bbox est trop coupée
63
64     # Normalisation pour le format YOLO
65     return [
66         new_cx / crop_w,
67         new_cy / crop_h,
68         new_w / crop_w,
69         new_h / crop_h
70     ]

```

J'utilise ensuite cette classe dans un programme simple, qui découpe, puis modifie les fichiers labels en conséquence relativement à la proportion restante de la bbox sur la nouvelle image. Suite à des tests, cette proportion à était fixée à 30%.

Listing A.3 – Corp du programme

```

1 for img_file in os.listdir(INPUT_IMAGE_DIR):
2     if not img_file.lower().endswith('.jpg', '.jpeg', '.png'):
3         continue
4
5     img_path = os.path.join(INPUT_IMAGE_DIR, img_file)
6     label_path = os.path.join(INPUT_LABEL_DIR, os.path.splitext(
7         img_file)[0] + '.txt')

```

```

7
8     img = cv2.imread(img_path)
9     orig_h, orig_w = img.shape[:2] # On récupère la taille de l'image
10    en px
11    crop_h = orig_h // ROWS
12    crop_w = orig_w // COLS
13
14    # Chargement des labels YOLO
15    bboxes = []
16    if os.path.exists(label_path):
17        with open(label_path, 'r') as f:
18            for line in f:
19                parts = line.strip().split()
20                if len(parts) != 5:
21                    continue
22                cls, x, y, w, h = map(float, parts)
23                bboxes.append((int(cls), [x, y, w, h]))
24
25    # Découpe de l'image
26    for i in range(ROWS):
27        for j in range(COLS):
28            x_offset = j * crop_w
29            y_offset = i * crop_h
30
31            crop = img[y_offset:y_offset + crop_h, x_offset:x_offset
32                        + crop_w]
33            crop_name = f"{os.path.splitext(img_file)[0]}_{i}_{j}.jpg"
34
35            cv2.imwrite(os.path.join(OUTPUT_IMAGE_DIR, crop_name),
36                        crop)
37
38            # On construit le fichier label
39            new_labels = []
40            for cls, bbox in bboxes:
41                new_bbox = convert_bbox(bbox, orig_w, orig_h,
42                                       x_offset, y_offset, crop_w, crop_h)
43                if new_bbox:
44                    new_labels.append(f'{cls} {new_bbox[0].join(f'{v:.6f}' for v in new_bbox)})')
45
46            # On écrit le fichier label
47            with open(os.path.join(OUTPUT_LABEL_DIR, crop_name), 'w'
48                      ) as f:
49                f.write('\n'.join(new_labels) + '\n' if new_labels
50                           else '')

```

B.

Script ROI

Définition des régions d'intérêts (ROI)

Si visuellement les régions concernées étaient évidentes, il a fallu les transformer en coordonnées afin de pouvoir les retranscrire dans un programme. Pour ce faire, j'ai simplement utilisé Paint pour dessiner (et ainsi les visualiser) et pour récupérer les pixels les délimitant. J'ai donc abouti à cela :

Listing B.1 – Coordonnées des ROI

```
1 rois = [  
2     (128, 0, 401, 99, -70, 1.3, 1.1),      # Zone 1  
3     (748, 0, 1170, 210, -100, 1.4, 1.1),    # Zone 2  
4     (793, 625, 908, 720, -80, 1.4, 1),       # Zone 3  
5     (15, 411, 524, 720, -80, 1.4, 1)         # Zone 4  
6 ]
```

Script de modification locale

La logique du script est la suivante. Je pars d'une image déjà améliorée et lui colle aux coordonnées précises des bouts de l'image originale auxquelles je vais appliquer des modifications personnalisées. J'obtiens donc au final une image avec les ROI traitées.

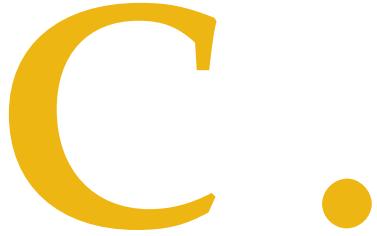
Listing B.2 – Configurations initiales

```
1 image = cv2.imread('modified_image2.jpg')  
2 image_ori = cv2.imread('originalimage.jpg')  
3  
4 # transformations à appliquer aux ROI
```

```

5 transformations = [
6 # (brightness, contrast, saturation)
7     (-100, 1.4, 1.1),
8     (-100, 1.4, 1.1),
9     (-100, 1.4, 1.1),
10    (-100, 1.4, 1.1)
11]
12
13 rois = [
14     (128, 0, 401, 99, -70, 1.3, 1.1),      # Zone 1
15     (748, 0, 1170, 210, -100, 1.4, 1.1),    # Zone 2
16     (793, 625, 908, 720, -80, 1.4, 1),       # Zone 3
17     (15, 411, 524, 720, -80, 1.4, 1)         # Zone 4
18]
19
20 # On ajoute les bouts de l'image originale
21 for (x1, y1, x2, y2, b, c, s) in rois:
22     image[y1:y2, x1:x2] = image_ori[y1:y2, x1:x2]
23
24 # On applique les modifications
25 for (x1, y1, x2, y2, brightness, contrast, saturation) in rois:
26     roi = image[y1:y2, x1:x2]
27     roi_mod = cv2.addWeighted(roi, contrast, np.zeros(roi.shape, roi.
28                               dtype), 0, brightness)
29
30     roi_hsv = cv2.cvtColor(roi_mod, cv2.COLOR_BGR2HSV).astype(np.
31                           float32)
32     roi_hsv[:, :, 1] *= saturation
33     roi_hsv[:, :, 1] = np.clip(roi_hsv[:, :, 1], 0, 255)
34     roi_hsv = roi_hsv.astype(np.uint8)
35     roi_bgr = cv2.cvtColor(roi_hsv, cv2.COLOR_HSV2BGR)
36
37     image[y1:y2, x1:x2] = roi_bgr
38
# Sauvegarder l'image
cv2.imwrite('modified_image_roi1.jpg', image)

```



C.

YOLO Extractor

Logique du script

La logique du script repose sur la conversion d'annotations d'objets présentes dans un fichier JSON en un format texte spécifique attendu par le modèle YOLO. Essentiellement, il s'agit de filtrer les annotations pour ne garder qu'une certaine classe ("bee"), puis de normaliser les coordonnées de chaque boîte englobante par rapport aux dimensions de l'image avant de les sauvegarder dans un fichier texte.

Plus précisément, le script :

- Charge d'abord les annotations au format JSON qui contiennent les coordonnées absolues des boîtes ainsi que la taille de l'image.
- Filtre les boîtes pour sélectionner uniquement celles ayant le label "bee".
- Pour chaque boîte sélectionnée, il calcule la position du centre et la taille en divisant les coordonnées et dimensions par la largeur et la hauteur de l'image, ce qui ramène les valeurs entre 0 et 1 (format normalisé attendu par YOLO).
- Écrit ensuite ces données formatées dans un fichier texte, chaque ligne correspondant à une boîte annotée, conformément aux exigences de format d'entrée de YOLO.

Cela permet de transformer des annotations au format JSON, souvent utilisé pour divers systèmes d'annotation, en un format texte simple et normalisé utilisable directement pour l'entraînement de modèles de détection d'objets tels que YOLO.

Listing C.1 – Script transformation json en label

```
1 import json
2
3
4 json_files = "ImageUpgrader/annotateImages/json/truthImage.json"
5 output_path = "ImageUpgrader/annotateImages/label/truthImage.txt"
6
7
8
9 with open(json_files, 'r') as f:
10     data = json.load(f)
11
12 boxes = data.get("boxes", [])
13 img_width = float(data.get("width", 1))
14 img_height = float(data.get("height", 1))
15
16 with open(output_path, 'w') as f_out:
17     for box in boxes:
18         if box.get("label") != "bee":
19             continue
20
21         classe = 0
22         x = float(box.get("x"))
23         y = float(box.get("y"))
24         width = float(box.get("width"))
25         height = float(box.get("height"))
26
27         # Normalisation par taille image
28         x_norm = x / img_width
29         y_norm = y / img_height
30         width_norm = width / img_width
31         height_norm = height / img_height
32
33         f_out.write(f"{classe} {x_norm:.6f} {y_norm:.6f} {width_norm}
34             :.6f} {height_norm:.6f}\n")
35
36 #print(json_files)
```

D.

Benchmark YOLO

Script d'évaluation des performances de détection

Ce script est conçu pour évaluer quantitativement les performances d'un modèle YOLO de détection d'objets (dans mon cas, des abeilles) en comparant ses prédictions sur une image test à la vérité terrain (annotations réelles). L'objectif est de calculer un score global sur 100 points, basé sur plusieurs critères (précision, rappel, IoU, etc.), et de sauvegarder les résultats pour archivage.

Listing D.1 – Script du benchmark

```
1 from ultralytics import YOLO
2 import pandas as pd
3 import numpy as np
4 import os
5 import shutil
6
7 empty = False
8
9
10 # Charge les prédictions du modèle YOLO sur une image et renvoie le
11 # dataframe
12 def load_PRED_labels(model_path, image_path, train_num, nb_epochs,
13 nb_batch):
14     model = YOLO(model_path, task='detect')
15     results = model(image_path)
16
17     empty_test = results[0].to_df()
```

```

18     global empty
19     empty = True
20     results[0].save(f"ResultImage_e{nb_epochs}_b{nb_batch}_t{train_num}.jpg")
21     data = []
22     df = pd.DataFrame(data)
23
24 else:
25     results[0].save(f"ResultImage_e{nb_epochs}_b{nb_batch}_t{train_num}.jpg")
26     results[0].save_txt(f"ResultLabels_e{nb_epochs}_b{nb_batch}_t{train_num}.txt")
27     df = pd.read_csv(f"ResultLabels_e{nb_epochs}_b{nb_batch}_t{train_num}.txt", sep=' ', header=None, names=['class', 'x_center', 'y_center', 'width', 'height'])
28
29 return df
30
31
32
33 def load_GT_labels(txt_path):
34     df = pd.read_csv(txt_path, sep=' ', header=None, names=['class', 'x_center', 'y_center', 'width', 'height'])
35     return df
36
37
38
39 # Convertit les dataframes en liste de dict (pour match_boxes)
40 def df_to_label_list(df):
41     labels = []
42     if not empty:
43         for _, row in df.iterrows():
44             bbox = [row['x_center'], row['y_center'], row['width'], row['height']]
45             labels.append({
46                 'class': int(row['class']),
47                 'bbox': bbox
48             })
49     return labels
50
51
52
53
54 def bbox_iou(box1, box2):
55     def to_corners(box):
56         x, y, w, h = box
57         return [x - w/2, y - h/2, x + w/2, y + h/2]

```

```

59     b1 = to_corners(box1)
60     b2 = to_corners(box2)
61
62     xi1, yi1 = max(b1[0], b2[0]), max(b1[1], b2[1])
63     xi2, yi2 = min(b1[2], b2[2]), min(b1[3], b2[3])
64     inter_area = max(0, xi2 - xi1) * max(0, yi2 - yi1)
65
66     b1_area = (b1[2] - b1[0]) * (b1[3] - b1[1])
67     b2_area = (b2[2] - b2[0]) * (b2[3] - b2[1])
68     union_area = b1_area + b2_area - inter_area
69     return inter_area / union_area if union_area > 0 else 0
70
71
72
73
74 def match_boxes(gt_labels, pred_labels, iou_thresh=0.5):
75
76     matches = []
77     gt_matched = set()
78     pred_matched = set()
79
80     for i, gt in enumerate(gt_labels):
81         for j, pred in enumerate(pred_labels):
82             if gt['class'] == pred['class']:
83                 iou = bbox_iou(gt['bbox'], pred['bbox'])
84                 if iou >= iou_thresh:
85                     matches.append((i, j, iou))
86                     gt_matched.add(i)
87                     pred_matched.add(j)
88
89     return matches, gt_matched, pred_matched
90
91
92
93
94 def compute_metrics(gt_labels, pred_labels, inference_time_ms):
95     n_gt = len(gt_labels)
96     n_pred = len(pred_labels)
97     matches, gt_matched, pred_matched = match_boxes(gt_labels,
98         pred_labels)
99     n_matched = len(matches)
100    ious = [iou for _, _, iou in matches]
101
102    # 1. Nombre d'individus détectés (25 points)
103    rel_error = abs(n_pred - n_gt) / max(1, n_gt)
104    score_count = 25 * (1 - min(rel_error / 0.5, 1))
105
106    # 2. IoU moyen (25 points)

```

```

106     iou_mean = np.mean(ious) if ious else 0
107     score_iou = 25 * (iou_mean - 0.5) / 0.35 if iou_mean >= 0.5 else
108         0
109     score_iou = min(max(score_iou, 0), 25)
110
111     # 3. Précision (10 points)
112     precision = n_matched / n_pred if n_pred > 0 else 0
113     score_precision = 10 if precision >= 0.95 else 10 * precision if
114         precision >= 0.1 else 0
115     score_precision = min(max(score_precision, 0), 10)
116
117     # 4. Rappel (10 points)
118     recall = n_matched / n_gt if n_gt > 0 else 0
119     score_recall = 10 if recall >= 0.95 else 10 * recall if recall >=
120         0.1 else 0
121     score_recall = min(max(score_recall, 0), 10)
122
123     # 5. mAP@0.5 (approximation) (10 points)
124     ap = n_matched / (n_pred + n_gt - n_matched) if (n_pred + n_gt -
125         n_matched) > 0 else 0
126     score_map = 10 * ap if ap >= 0.1 else 0
127     score_map = min(max(score_map, 0), 10)
128
129     # 6. Temps d'inférence (10 points)
130     if inference_time_ms < 15:
131         score_time = 10
132     elif inference_time_ms > 1000:
133         score_time = 0
134     else:
135         score_time = 10 * (1 - (inference_time_ms - 15) / 900)
136
137     # 7. Faux positifs (10 points)
138     false_positives = n_pred - len(pred_matched)
139     fpr = false_positives / max(1, n_pred)
140     if fpr <= 0.05:
141         score_fpr = 10
142     elif fpr >= 0.3:
143         score_fpr = 0
144     else:
145         score_fpr = 10 * (1 - (fpr - 0.05) / 0.25)
146
147     # Score F1 (Affichage seulement)
148     if recall + precision > 0:
149         score_f1 = 2 * ((recall * precision) / (recall + precision))
150     else:
151         score_f1 = 0

```

```

150     # Score final
151     score_total = score_count + score_iou + score_precision +
152         score_recall + score_map + score_time + score_fpr
153
154     details = {
155         'score_total': round(score_total, 2),
156         'score_count': round(score_count, 2),
157         'score_iou': round(score_iou, 2),
158         'score_precision': round(score_precision, 2),
159         'score_recall': round(score_recall, 2),
160         'score_f1': round(score_f1, 2),
161         'score_map': round(score_map, 2),
162         'score_time': round(score_time, 2),
163         'score_fpr': round(score_fpr, 2),
164         'precision': round(precision, 3),
165         'recall': round(recall, 3),
166         'iou_mean': round(iou_mean, 3),
167         'n_gt': n_gt,
168         'n_pred': n_pred,
169         'n_matched': n_matched,
170         'inference_time_ms': inference_time_ms,
171         'false_positives': false_positives,
172         'fpr': round(fpr, 3)
173     }
174
175
176
177     def save_metrics_to_txt(score_details, model_path, nb_epochs,
178                             nb_batch, id_num):
179         file_name = f"Results/Result_e{nb_epochs}_b{nb_batch}_t{id_num}/
180                     ResultsResum_e{nb_epochs}_b{nb_batch}_t{id_num}.txt"
181         os.makedirs(f"Results/Result_e{nb_epochs}_b{nb_batch}_t{id_num}",
182                     exist_ok=True)
183
184         shutil.copyfile(model_path, f"Results/Result_e{nb_epochs}_b{
185             nb_batch}_t{id_num}/Best0{id_num}.pt")
186
187         with open(file_name, 'w') as f:
188             f.write(f"*** RESULTATS ***\n\nepoch={
189                 nb_epochs}, batch={nb_batch}, BDD version={bdd_version},
190                 train ID={id_num}\n\n")
191             f.write(f"Score global: {score_details['score_total']}/100\n")
192             f.write("---- Détails des scores ----\n")
193             f.write(f"Nombre individus: {score_details['score_count']}
194                 []/25\n")
195             f.write(f"IoU moyen: {score_details['score_iou']}/25 (valeur

```

```

        ={score_details['iou_mean']})\n")
189 f.write(f"Précision: {score_details['score_precision']}/10 (\n"
          valeur={score_details['precision']})\n")
190 f.write(f"Rappel: {score_details['score_recall']}/10 (valeur
          ={score_details['recall']})\n")
191 f.write(f"F1: {score_details['score_f1']}\n")
192 f.write(f"mAP@0.5: {score_details['score_map']}/10\n")
193 f.write(f"Temps d'inférence: {score_details['score_time']}/10
          ({score_details['inference_time_ms']} ms)\n")
194 f.write(f"Faux positifs: {score_details['score_fpr']}/10 (fpr
          ={score_details['fpr']})\n\n")
195 f.write("---- Statistiques brutes ----\n")
196 f.write(f"Nombre total réels d'individus: {score_details['
          n_gt']}\n")
197 f.write(f"Nombre total détectés d'individus: {score_details['
          n_pred']}\n")
198 f.write(f"Invidus réels détectés: {score_details['n_matched
          ']}\n")
199 f.write(f"Faux positifs: {score_details['false_positives']}\n
          ")
200
201
202 #----
203
204 image_path = 'Truth/truthImage.jpg'
205 gt_txt = 'Truth/truthImage.txt'
206
207
208 nb_epochs = 100
209 nb_batch = 32
210 bdd_version = 2
211 train_num = 10
212
213 inference_time_ms = 65
214
215 model_path = f'runs/detect/train{train_num}/weights/best.pt'
216
217 gt_labels = df_to_label_list(load_GT_labels(gt_txt))
218 pred_labels = df_to_label_list(load_PRED_labels(model_path,
          image_path, train_num, nb_epochs, nb_batch))
219
220 if empty == False:
221     score_details = compute_metrics(gt_labels, pred_labels,
          inference_time_ms)
222     save_metrics_to_txt(score_details, model_path, nb_epochs,
          nb_batch, train_num)
223     shutil.move(f"ResultImage_e{nb_epochs}_b{nb_batch}_t{train_num}.
          jpg", f"Results/Result_e{nb_epochs}_b{nb_batch}_t{train_num}/"

```

```

        )
shutil.move(f"ResultLabels_e{nb_epochs}_b{nb_batch}_t{train_num}.txt",
            f"Results/Result_e{nb_epochs}_b{nb_batch}_t{train_num}/")
)

225
226 print("\n#####")
227 print("Score global : ", score_details['score_total'],"/100")
228 print("#####\n")
229 print("Détail du score :")
230 print("-----")
231 for score in score_details:
232     print(score, " = ", score_details[score])
233
234 else:
235     file_name = f"Results/Result_e{nb_epochs}_b{nb_batch}_t{train_num}_ResultsResum_e{nb_epochs}_b{nb_batch}_t{train_num}.txt"
236     os.makedirs(f"Results/Result_e{nb_epochs}_b{nb_batch}_t{train_num}", exist_ok=True)
237
238     shutil.copyfile(model_path, f"Results/Result_e{nb_epochs}_b{nb_batch}_t{train_num}/Best0{train_num}.pt")
239
240     with open(file_name, 'w') as f:
241         f.write(f" *** RESULTATS ***\n\nepochs={nb_epochs}, batch={nb_batch}, BDD version={bdd_version},\ntrain ID={train_num}\n\n")
242         f.write(f"Score global: 0/100\n\n")
243         f.write("---- Détails des scores ----\n")
244         f.write("NULL\n")
245         f.write("---- Statistiques brutes ----\n")
246         f.write("NULL\n")
247
248
249     print("\n#####")
250     print("Score global : 0/100")
251     print("AUCUNES ABEILLES N'A ETE DETECTE")
252     print("#####\n")

```

Logique du script

1. Chargement des données

- **Vérité terrain (GT)** : lecture d'un fichier .txt contenant les annotations réelles des objets (classe, coordonnées normalisées des boîtes englobantes).
- **Prédictions YOLO** : utilisation du modèle entraîné (best.pt) pour détecter les objets dans l'image test.
 - Si aucune détection n'est faite, un indicateur `empty` passe à True et le script enregistre une image annotée vide.
 - Si des détections existent, elles sont sauvegardées en image et fichier texte.

2. Conversion et préparation

Les données GT et prédictions sont converties en listes de dictionnaires (`class` + `bbox`) afin de faciliter la comparaison entre boîtes.

3. Calcul des correspondances

Une fonction `match_boxes` associe chaque boîte prédite à la boîte réelle correspondante si :

- Les classes sont identiques.
- L'Intersection over Union (IoU) est supérieure ou égale à un seuil de 0,5.

4. Métriques et notation

Le script calcule :

- **Exactitude du comptage** (25 pts) : comparaison entre nombre d'objets détectés et réels.
- **IoU moyen** (25 pts) : qualité moyenne d'ajustement des boîtes.
- **Précision** (10 pts) : proportion de détections correctes parmi toutes les détections.
- **Rappel** (10 pts) : proportion d'objets réels correctement détectés.
- **mAP@0.5 approximatif** (10 pts) : métrique standard en vision par ordinateur.
- **Temps d'inférence** (10 pts) : rapidité du modèle (score maximal si < 15 ms).
- **Taux de faux positifs** (10 pts) : pénalités si trop de détections incorrectes.

Un `score F1` est aussi calculé à titre informatif (non intégré à la note finale). Le score total correspond à la somme de tous les points attribués.

5. Sauvegarde des résultats

Création automatique d'un dossier `Results/` pour stocker :

- L'image annotée.
- Le fichier des prédictions YOLO.

- Un résumé texte détaillant le score global, le détail par critère, et les statistiques brutes.
- Une copie du modèle utilisé pour conserver la traçabilité.

6. Gestion des cas sans détection

Si aucune abeille n'est détectée, le script enregistre un score nul et mentionne explicitement l'absence de détection.

En résumé : ce script permet d'évaluer objectivement un modèle de détection sur un échantillon fixe, en attribuant un score global normalisé et en enregistrant automatiquement tous les éléments nécessaires pour comparaison et archivage.

E.

Matrice de corrélation

Analyse qualitative de la matrice de corrélation

Pour rappel, le benchmark repose sur une série de 18 entraînements réalisés sur le même calculateur (celui de l'IGN), avec quatre versions d'un dataset d'images d'abeilles (909 images, trois classes : bee, queen, drone). Les versions se différencient par des ajouts ou transformations d'images, allant du dataset original (v1) à des versions enrichies et/ou filtrées (v2, v3, v4). Chaque entraînement a été paramétré avec un nombre d'epochs et un batch size définis, puis évalué via un script de benchmark (YOLOBenchmark.py) appliquant les mêmes conditions de test à tous les modèles. Les métriques de performance ont ensuite été compilées dans un tableau, permettant de calculer la matrice de corrélation ci-dessous.

Lecture de la matrice

	Score total	Precision	Recall	IoU mean	Nb. Detec.	Matched	FP	FP rate	Epochs	Batch	BDD version
Score total	1										
Precision	0.881	1									
Recall	0.915	0.647	1								
IoU mean	0.974	0.954	0.841	1							
Nb. Detec.	0.899	0.630	0.997	0.827	1						
Matched	0.915	0.647	0.971	0.841	0.997	1					
FP	0.846	0.578	0.971	0.779	0.986	0.971	1				
FP rate	0.907	0.650	0.989	0.848	0.988	0.989	0.967	1			
Epochs	0.174	0.187	0.230	0.182	0.225	0.229	0.053	0.185	1		
Batch	0.059	-0.025	0.074	0.015	0.185	0.074	-0.001	0.058	0.716	1	
BDD version	0.745	0.514	0.869	0.676	0.868	0.869	0.850	0.808	0.129	0.093	1

TABLE E.1 – Matrice de corrélation des métriques

Cette matrice de corrélation de Pearson met en évidence la force et le sens des relations entre les différentes métriques et paramètres suivis.

Bloc principal : interdépendance des métriques de détection

On observe un ensemble de corrélations très fortes entre :

- Score total et IoU moyen (0,97), recall (0,91), matched detections (0,91), et nombre total de détections (0,90).
- Recall et Nb. detections (0,997), Matched et Nb. detections (0,997) : ces relations proches de 1 traduisent que le rappel et les détections sont quasiment interchangeables dans ce contexte.
- FP et FP rate (0,967) : plus un modèle détecte agressivement, plus le risque d'erreurs augmente.

Ces résultats confirment que le score total est largement construit sur la qualité des détections et leur volume.

Relation précision / localisation

La précision est fortement liée au IoU moyen (0,954) : de meilleures localisations spatiales tendent à améliorer la précision. Toutefois, sa corrélation avec le score total (0,88) reste légèrement inférieure à celle du recall (0,91), signe que dans mon indicateur global, le rappel a un poids un peu plus déterminant.

Impact des hyperparamètres

- Epochs : corrélation faible avec le score total (0,18). L'effet est positif mais limité, ce qui concorde avec les observations de plateau au-delà de 100–200 epochs.
- Batch : corrélation quasi nulle (0,059) et légèrement négative avec la précision (-0,025).

- BDD version : corrélation élevée avec le score total (0,745), ce qui souligne que le choix et l'enrichissement du dataset influencent bien plus la performance qu'un réglage d'hyperparamètres.

Concordance avec le benchmark

Les modèles les plus performants (train2, train10, train13, train18, train12) affichent tous un IoU > 0,76 et un FP rate < 0,3, ce qui correspond aux corrélations observées. La version 3 du dataset se démarque par ses résultats moyens élevés, tandis que la version 1, non enrichie, conduit à des performances quasi nulles.

Conclusion

Cette analyse statistique confirme que la performance des modèles est avant tout liée à la qualité des données d'entraînement (IoU, recall, matched), et que l'impact des hyperparamètres comme le batch size ou le nombre d'epochs reste marginal. En pratique, cela signifie que pour améliorer les résultats, il faut concentrer les efforts sur l'enrichissement, la diversité et la pertinence du dataset, plutôt que sur l'optimisation fine des paramètres d'entraînement.

F.

Script de tracking

Script de tracking adapté du dépôt de VuPhan

Pour être fonctionnel, il est nécessaire de télécharger la bibliothèque associée au tracker ByteTrack.

Listing F.1 – Script de tracking

```
1 from bytetrack.byte_track import ByteTrack
2 from dataclasses import dataclass
3
4 from supervision import ColorPalette, ColorLookup
5 from supervision import VideoInfo, VideoSink,
   get_video_frames_generator
6 from supervision import BoundingBoxAnnotator, LabelAnnotator,
   TraceAnnotator
7 from supervision import Detections
8
9
10 import numpy as np
11 from ultralytics import YOLO
12 import time
13 import json
14
15
16
17 INPUT_PATH = "UP_Extrait1_15s.mp4"
18 OUTPUT_PATH = "TrackingVideo.mp4"
19
20 label_mode = False
21 color_mode = True
```

```

22 palette_name = 'hsv'
23
24 # 1: default / 2: original / 3: white / 4: black / 5: json / 6:
25     trajectories
26 video_type = 3
27
28 MONO_PALETTE = ["#FF0000"]
29
30 GPT_PALETTE = [
31     "#FF0000", "#036B03", "#FFFF00", "#FF00FF", "#00FFFF",
32     "#FF6600", "#66FF00", "#00FF66", "#6600FF", "#FF0066",
33     "#CC0000", "#00CC00", "#CCCC00", "#CC00CC", "#00CCCC",
34     "#FF3300", "#33FF00", "#00FF33", "#FF0033",
35     "#990000", "#009900", "#999900", "#990099", "#009999",
36     "#FF9900", "#99FF00", "#00FF99", "#9900FF", "#FF0099",
37     "#FF1A1A", "#1AFF1A", "#FFFF1A", "#FF1AFF", "#1AFFF",
38     "#FF3300", "#195019", "#FFF33", "#FF33FF", "#BOC7C7",
39     "#CC3300", "#2A4122", "#00CC33", "#ACAOCF", "#CC0033",
40     "#CC6600", "#66CC00", "#00CC66", "#9042DF", "#CC0066",
41     "#FF0033", "#33FF00", "#FF3300", "#00FF33", "#FF00CC",
42     "#00FFCC", "#CC00FF", "#FFCC00", "#CCFF00", "#00CCFF",
43     "#FF0066", "#66FF00", "#FF6600", "#00FF66", "#FF3366",
44     "#6F8369", "#232427", "#FF6633", "#696969", "#FF3399",
45     "#99FF33", "#FF9933", "#33FF99", "#FF66CC", "#CCFF66",
46     "#FFCC66", "#66FFCC", "#FF6699", "#99FF66", "#FF9966",
47     "#66FF99", "#FF33CC", "#CCFF33", "#FFCC33", "#33FFCC",
48     "#FF0033", "#33FF33", "#FF3300", "#00FF33", "#FF00AA",
49     "#00FFAA", "#311D3B", "#FFAA00", "#AAFF00", "#00AAFF",
50     "#FFAA33", "#1B3027", "#AA33FF", "#FF33AA", "#33AAFF",
51     "#AAFF33", "#FF55FF", "#55FFFF", "#FFFF55", "#FF5555",
52     "#55FF55", "#FF7777", "#77FF77", "#FF77FF", "#77FFFF",
53     "#FFFF77", "#FF8800", "#00FF88", "#FF0088", "#88FF00",
54     "#FFBB00", "#00FFBB", "#FF00BB", "#BBFF00", "#00BBFF",
55     "#FFB3B3", "#B3FFB3", "#FFB3FF", "#B3FFFF", "#FFFFB3",
56 ]
57
58
59 class ObjectTracking:
60     def __init__(self, input_video_path, output_video_path) -> None:
61         self.model = YOLO("best015.pt")
62         self.model.fuse()
63
64         # Mapping classe (abeilles)
65         self.CLASS_NAMES_DICT = self.model.model.names
66         self.CLASS_ID = [0]
67
68         self.input_video_path = input_video_path

```

```

69     self.output_video_path = output_video_path
70
71     # Initialisation ByteTrack with updated parameter names
72     self.byte_tracker = ByteTrack(
73         track_activation_threshold=0.35,
74         lost_track_buffer=300,                               # 1s à 60 fps
75         minimum_matching_threshold=0.9,
76         frame_rate=60
77     )
78
79     # Infos vidéo & traceurs
80     self.video_info = VideoInfo.from_video_path(self.
81         input_video_path)
82     self.generator = get_video_frames_generator(self.
83         input_video_path)
84     self.trace_annotator = TraceAnnotator(thickness=4,
85         trace_length=1000)
86
87
88     self.seen_tracker_ids = 0
89
90
91     # key: tracker_id, value: list of (x1, y1, x2, y2)
92     self.trajectories = {}
93
94
95     def to_json(self, detections):
96         # Stockage des trajectoires
97         bboxes = detections.xyxy
98         track_ids = detections.tracker_id
99
100        for bbox, track_id in zip(bboxes, track_ids):
101            x1, y1, x2, y2 = bbox
102            if track_id not in self.trajectories:
103                self.trajectories[track_id] = []
104                self.trajectories[track_id].append((x1, y1, x2, y2))
105
106        json_trajectories = {str(track_id): [(int(x1), int(y1), int(
107            x2), int(y2)) for x1, y1, x2, y2 in points] for track_id,
108        points in self.trajectories.items()}
109
109        with open("TrackTrajectories.json", "w") as f:
110            json.dump(json_trajectories, f)

```

```

110
111     return
112
113
114     def callback(self, frame, index):
115         original_frame = frame.copy()
116
117         # Prédiction YOLO sur l'image
118         results = self.model(frame, verbose=False)[0]
119         detections = Detections.from_ultralytics(results)
120
121         # On garde uniquement les classes ciblées
122         detections = detections[np.isin(detections.class_id, self.
123             CLASS_ID)]
124
125         # Suivi avec ByteTrack
126         detections = self.byte_tracker.update_with_detections(
127             detections)
128
129         # Création des labels (ex: "#4 - 0.45")
130         labels = [
131             f"#{tracker_id} {confidence:0.2f}"
132             for confidence, tracker_id
133             in zip(detections.confidence, detections.tracker_id)
134         ]
135
136         match video_type:
137             case 1 | 'default':
138                 # ByteTrack bboxes and trajectories on YOLO basic out
139                 # video
140                 initial_frame = results.plot(conf=False, labels=False
141                     , boxes=True, masks=True, probs=False, kpt_line=
142                         False, line_width=2)
143                 bbox_flag = True
144
145             case 2 | 'original':
146                 # ByteTrack bboxes and trajectories on blank original
147                 # upgraded video
148                 initial_frame = original_frame
149                 bbox_flag = True
150
151             case 3 | 'white':
152                 # ByteTrack trajectories without bboxes on a white
153                 # background
154                 initial_frame = np.full((self.video_info.height, self.
155                     .video_info.width, 3), 255, dtype=np.uint8)
156                 bbox_flag = False

```

```

150
151     case 4 | 'black':
152         # ByteTrack trajectories without bboxes on a black
153         # background
154         initial_frame = np.zeros((self.video_info.height,
155                                     self.video_info.width, 3), dtype=np.uint8)
156         bbox_flag = False
157
158     case 5 | 'json':
159         # ByteTrack data on a json file and default video
160         # generation
161         initial_frame = frame.copy()
162         self.to_json(detections)
163         bbox_flag = True
164
165     case 6 | 'trajectories':
166         # ByteTrack trajectories on blank original upgraded
167         # video
168         initial_frame = original_frame
169         bbox_flag = False
170
171     # Ajout des traces + boîtes + labels
172     #Palette = ColorPalette.from_matplotlib(palette_name, 50)
173     if color_mode :
174         Palette = ColorPalette.from_hex(GPT_PALETTE)
175     else:
176         Palette = ColorPalette.from_hex(MONO_PALETTE)
177
178     self.trace_annotator.color = self.bbox_annotator.color = self
179     .label_annotator.color = Palette
180
181     frame_traced = self.trace_annotator.annotate(initial_frame,
182                                                 detections=detections, custom_color_lookup=ColorLookup.
183                                                 TRACK)
184
185     if bbox_flag :
186         frame_boxes = self.bbox_annotator.annotate(frame_traced,
187             detections=detections, custom_color_lookup=ColorLookup
188             .TRACK)
189     else:
190         frame_boxes = frame_traced
191
192     if label_mode:
193         frame_annotated = self.label_annotator.annotate(
194             frame_boxes, detections=detections, labels=labels,
195             custom_color_lookup=ColorLookup.TRACK)
196     else:

```

```

187         frame_annotated = frame_boxes
188
189     for tracker_id in detections.tracker_id:
190         if tracker_id > self.seen_tracker_ids:
191             self.seen_tracker_ids = tracker_id
192         else:
193             continue
194
195     return frame_annotated
196
197
198 def process(self):
199     with VideoSink(target_path=self.output_video_path, video_info=
200                     =self.video_info) as sink:
201         for index, frame in enumerate(self.generator):
202             result_frame = self.callback(frame, index)
203             sink.write_frame(result_frame)
204
205 if __name__ == "__main__":
206     begin = time.time()
207     obj = ObjectTracking(INPUT_PATH, OUTPUT_PATH)
208     obj.process()
209     end = time.time()
210
211     print(f'Le programme a mis {round(end-begin, 2)} secondes et a
212           détecté {obj.seen_tracker_ids} individus différents.')

```

Logique générale du script de tracking

Ce script met en œuvre un pipeline complet de détection et de suivi d'objets dans une vidéo, en l'occurrence des abeilles, à partir d'un modèle YOLO pré-entraîné.

La logique se déroule ainsi :

1. Initialisation

Le script charge le modèle YOLO optimisé (best015.pt) et configure ByteTrack pour le suivi multi-objets. Il définit aussi les paramètres d'affichage (palette de couleurs, mode vidéo, options de labels).

2. Lecture vidéo

Les informations de la vidéo d'entrée sont extraites et un générateur fournit les images une à une pour traitement.

3. Détection et filtrage

À chaque image, YOLO effectue la détection. Seules les classes visées (abeilles) sont conservées.

4. Suivi des individus

ByteTrack attribue un identifiant unique à chaque individu détecté et maintient ce lien au fil des images, même en cas de déplacements rapides ou d'occlusions partielles.

5. Annotation et sortie

En fonction du mode choisi, le script ajoute sur la vidéo les trajectoires, boîtes englobantes et/ou labels. Il peut aussi produire un fond blanc/noir, ou enregistrer les trajectoires dans un fichier JSON.

6. Export vidéo et statistiques

Les images annotées sont compilées en une vidéo de sortie. À la fin, le script indique le temps total d'exécution et le nombre d'individus uniques détectés.

En résumé, ce script automatise l'analyse d'une vidéo pour détecter, suivre et représenter visuellement les déplacements d'abeilles, avec la possibilité d'exporter les données brutes de trajectoire.

Bibliographie

- [1] *A Deep Dive Into Semantic Segmentation Evaluation Metrics* | HackerNoon. URL : <https://hackernoon.com/a-deep-dive-into-semantic-segmentation-evaluation-metrics> (visité le 03/06/2025).
- [2] ABONIA SOJASINGARAYAR. *Fine-Tuning YOLOv10 for Object Detection on a Custom Dataset #yolo #finetuning*. 5 juill. 2024. URL : <https://www.youtube.com/watch?v=QCXCbbi5w-k> (visité le 03/06/2025).
- [3] *Acteur de référence de la donnée géolocalisée souveraine, l'IGN avance sur un chemin à baliser*. Sénat. 7 avr. 2023. URL : <https://www.senat.fr/rap/r22-017/r22-017.html> (visité le 27/06/2025).
- [4] *Aperçu de l'évaluation et de la mise au point des modèles - Ultralytics YOLO Docs*. URL : <https://docs.ultralytics.com/fr/guides/model-evaluation-insights/#confidence-score> (visité le 03/06/2025).
- [5] *ByteTrack/exp/ example/mot at main · FoundationVision/ ByteTrack*. GitHub. URL : <https://github.com/FoundationVision/ByteTrack/tree/main/exp/example/mot> (visité le 29/07/2025).
- [6] *CLAHE Histogram Equalization - OpenCV*. GeeksforGeeks. Section : Python. URL : <https://www.geeksforgeeks.org/clahе-histogram-equalization-opencv/> (visité le 17/06/2025).
- [7] Duc Manh Nguyen DANG et al. *YOWOv3 : An Efficient and Generalized Framework for Human Action Detection and Recognition*. 9 août 2024. DOI : [10.48550/arXiv.2408.02623](https://doi.org/10.48550/arXiv.2408.02623). arXiv : [2408.02623\[cs\]](https://arxiv.org/abs/2408.02623). URL : <http://arxiv.org/abs/2408.02623> (visité le 06/06/2025).
- [8] “Décision n° 06-D-23 du 21 juillet 2006 relative à la situation de la concurrence dans les secteurs de l'édition cartographique et de l'information touristique”. In : ().
- [9] Rukesh DUWAL. *iamrukeshduwal/yolov11_real_time_object_detection_with_DeepSORT*. original-date : 2024-10-08T05:07:14Z. 8 juin 2025. URL : https://github.com/iamrukeshduwal/yolov11_real_time_object_detection_with_DeepSORT (visité le 25/07/2025).
- [10] *Enhance Local Contrast (CLAHE)*. ImageJ Wiki. Section : Filtering. URL : <https://imagej.github.io/plugins/clahе> (visité le 08/08/2025).

- [11] *FFmpeg*. URL : <https://ffmpeg.org/> (visité le 03/06/2025).
- [12] Sylvain GALOPIN. *GeoDanceHive : An Operational Hive for Honeybees Dances Recording*. URL : <https://www.mdpi.com/2076-2615/13/7/1182> (visité le 28/04/2025).
- [13] Sylvain GALOPIN. *Geolocating Bees by Translating the Waggle Dance Into Spatial Coordinates*. URL : https://www.researchgate.net/publication/381025459_Geolocating_Bees_by_Translating_the_Waggle_Dance_Into_Spatial_Coordinates (visité le 28/04/2025).
- [14] Géomatique. In : Wikipédia. Page Version ID : 223284952. 23 fév. 2025. URL : <https://fr.wikipedia.org/w/index.php?title=G%C3%A9omatique&oldid=223284952> (visité le 01/07/2025).
- [15] *How to Fine Tune YOLOv8 : Step-by-Step Guide | yolov8*. URL : https://yolov8.org/how-to-use-fine-tune-yolov8/#Fine-Tuning_Steps_How_to_Use (visité le 03/06/2025).
- [16] Cécile JARRY. « *La carte ne cherche pas à concurrencer le GPS, elle occupe toute sa place à côté des dispositifs de guidage* ». IC. 30 mai 2022. URL : <https://www.lemagic.fr/case-studies/la-carte-ne-cherche-pas-a-concurrencer-le-gps-elle-occupe-toute-sa-place-a-cote-des-dispositifs-de-guidage/> (visité le 01/07/2025).
- [17] Glenn JOCHER. *YOLOv5 by Ultralytics*. Version 7.0. Mai 2020. DOI : [10.5281/zenodo.3908559](https://zenodo.3908559). URL : <https://github.com/ultralytics/yolov5> (visité le 03/06/2025).
- [18] Tom KELDENICH. *Recall, Precision, F1 Score - Explication Simple Métrique en ML*. Section : Les Bases. 2 sept. 2021. URL : <https://inside-machinelearning.com/recall-precision-f1-score/> (visité le 08/07/2025).
- [19] LASTIG. URL : <https://www.umr-lastig.fr/> (visité le 29/06/2025).
- [20] Sudip LAUDARI. *YOLOv10 : A Step-by-Step Guide to Object Detection on a Custom Dataset*. Medium. 11 juill. 2024. URL : <https://medium.com/@sudiplaudari/yolov10-a-step-by-step-guide-to-object-detection-on-a-custom-dataset-9f3e3e56921c> (visité le 03/06/2025).
- [21] *OpenCV : Adding (blending) two images using OpenCV*. URL : https://docs.opencv.org/3.4/d5/dc4/tutorial_adding_images.html (visité le 03/06/2025).
- [22] *OpenCV : Drawing Functions in OpenCV*. URL : https://docs.opencv.org/4.x/dc/d45/tutorial_py_drawing_functions.html (visité le 28/07/2025).
- [23] *OpenCV : Getting Started with Videos*. URL : https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html (visité le 22/07/2025).
- [24] Slawomir Telega PhD. *Simple YOLO custom dataset automatic generation*. Medium. 16 août 2024. URL : <https://medium.com/@telega.slawomir.ai/simple-yolo-custom-dataset-automatic-generation-295f5c9ccfb2> (visité le 03/06/2025).

- [25] Présentation de l'IGN. URL : https://www.ign.fr/publications-de-l-ign/institut/carte_identite/presentation_ign.pdf (visité le 27/06/2025).
- [26] SNT Photo Filtrage. URL : <https://www.infoforall.fr/act/snt/filtrage-d-une-image/> (visité le 27/05/2025).
- [27] SUFFICIENT_CHAIN_431. How to do data augmentation on a YOLO annotated data-set ? r/computervision. 4 oct. 2024. URL : https://www.reddit.com/r/computervision/comments/1fvw0nn/how_to_do_data_augmentation_on_a_yolo_annotation/ (visité le 21/05/2025).
- [28] Terre, climat : qu'est-ce que l'Anthropocène, ère géologique | vie-publique.fr. 8 oct. 2019. URL : <https://www.vie-publique.fr/parole-dexpert/271086-terre-climat-quest-ce-que-lanthropocene-ere-geologique> (visité le 27/06/2025).
- [29] ULTRALYTICS. Augmentation des données YOLO. URL : <https://docs.ultralytics.com/fr/guides/yolo-data-augmentation> (visité le 21/05/2025).
- [30] ULTRALYTICS. Hyperparameter Tuning. URL : <https://docs.ultralytics.com/guides/hyperparameter-tuning> (visité le 03/06/2025).
- [31] ULTRALYTICS. Ultralytics - configuration. URL : <https://docs.ultralytics.com/usage/cfg> (visité le 23/05/2025).
- [32] ULTRALYTICS. Ultralytics - documentation. URL : <https://docs.ultralytics.com/reference/utils> (visité le 08/07/2025).
- [33] ULTRALYTICS. Ultralytics - Train. URL : <https://docs.ultralytics.com/modes/train> (visité le 03/06/2025).
- [34] Ultralytics YOLO Hyperparameter Tuning Guide - Ultralytics YOLO Docs. URL : <https://docs.ultralytics.com/guides/hyperparameter-tuning/#introduction> (visité le 03/06/2025).
- [35] Understanding Evaluation Metrics in Medical Image Segmentation | by Nghi Huynh | Medium. URL : https://medium.com/@nghihuynh_37300/understanding-evaluation-metrics-in-medical-image-segmentation-d289a373a3f (visité le 03/06/2025).
- [36] What are different evaluation metrics used to evaluate image segmentation models ? | GeeksforGeeks. URL : <https://www.geeksforgeeks.org/what-are-different-evaluation-metrics-used-to-evaluate-image-segmentation-models/> (visité le 03/06/2025).
- [37] Nicolai Wojke. nwojke/deep_sort. original-date : 2017-02-04T13:17:42Z. 29 juill. 2025. URL : https://github.com/nwojke/deep_sort (visité le 29/07/2025).
- [38] yolo-bytetrack-vehicle-tracking/object_tracking.py at main · VuBacktracking/yolo-bytetrack-vehicle-tracking. URL : https://github.com/VuBacktracking/yolo-bytetrack-vehicle-tracking/blob/main/object_tracking.py (visité le 28/07/2025).
- [39] YOLO-DeepSort-Tracking/main.py at main · seonghwan97/YOLO-DeepSort-Tracking. GitHub. URL : <https://github.com/seonghwan97/YOLO-DeepSort-Tracking/blob/main/main.py> (visité le 29/07/2025).

