



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE

Département de génie informatique et de génie logiciel

INF8970 – INF8980 – INF8985
Projet final en génie informatique

Rapport final

Estimation de la taille d'un poisson à partir d'une photo

Équipe No 10

Emile Dumas

Félix Duguay

Félix Gauthier

Guillaume Rochon-Vear

Steven Duchêne

Wissale Achkanou

20 avril 2023

Table des matières

1. Changement dans le contexte du projet	3
1.1 Description des changements imposés par le client depuis la mi-session	3
1.2 Difficultés techniques nouvellement apparues	4
2. Description de l'architecture finale du système (Q5.2 – Q5.3 – Q5.4)	5
2.1 Résumé de ce qu'était le système à la mi-session et ce qu'il est maintenant (Q4.5 et Q4.6)	5
2.2 Quelques diagrammes de classes ou de modules	8
2.3 Diagramme d'états et/ou d'interface usager	10
2.4 Interfaces logicielles et/ou matérielles importantes	13
2.5 Description d'outils, librairies ou cadre de travail (framework) utilisés	14
2.6 Quelques méthodes de tests appliquées pour valider la solution	16
3. Gestion du projet	16
3.1 Identification des tâches principales du projet	16
3.2 Répartition des tâches et responsabilités dans l'équipe	18
3.3 Principales difficultés de gestion rencontrées durant la deuxième partie du projet	20
4. Conclusion	22
4.1 Retour sommaire sur le travail complété et ce qui n'a pu être réalisé (Q3.5)	22
4.2 Causes des succès et difficultés rencontrées (Q3.6)	25
5. Apprentissage en continu (Q12)	29
5.1 Lacunes identifiées dans ses savoirs et savoir-faire durant le projet	29
5.2 Méthodes prises pour y remédier	31
5.3 Identifier comment cet aspect aurait pu être amélioré	34
6. Références utilisées	35

1. Changement dans le contexte du projet

Suite à notre présentation de mi-session, nous avons reçu plusieurs retours positifs de la part de nos clients qui mettaient en évidence leur satisfaction par rapport à l'avancement du projet et la complétion des tâches. Aucun changement n'a été imposé par le client depuis la mi-session. Toutefois, une suggestion de tests reliés à la prise de photo nous a été suggérée. Une fonctionnalité a toutefois été retirée, puisqu'elle n'était tout simplement pas réalisable. En effet, nous avons cru possible de pouvoir estimer la pose du poisson à l'aide de la caméra stéréo du téléphone, mais l'algorithme de l'article « Estimating 3D Pose Estimation And Length Measurement of Severely Deformed Fish From Monocular Videos in Longline Fishing » ne convergeait presque jamais ou encore prenait un temps beaucoup trop long. Ce n'était donc pas suffisamment intéressant pour notre client, car le gain était minimal et le temps de manipulation était beaucoup plus long qu'estimé. De plus, c'était très imprécis et nous avons préféré nous concentrer sur des parties plus critiques de l'application comme l'intégration du LiDAR qui avait été problématique durant la première moitié du projet.

1.1 Description des changements imposés par le client depuis la mi-session

Les résultats présentés lors de la présentation ont grandement satisfait les attentes du client, malgré le fait que l'estimation automatique de la profondeur du poisson n'était pas encore en vigueur. En effet, dans le cas parfait où le poisson était bien identifié, nos estimations avaient une erreur d'au plus 5 centimètres. Ces résultats ont donc lancé un questionnement lors d'une de nos réunions quotidiennes: comment notre prototype réagit-il lorsque le poisson sort de l'eau et réfléchit la lumière? Après cette discussion nous avons déterminé qu'il fallait vérifier que la taille du poisson est toujours précise même avec une grande réflexion. Il a donc fallu vérifier cela en ajoutant des surfaces réfléchissantes à nos maquettes. Ensuite, du côté de l'estimation de pose, nous avons simplement laissé tomber ce module et modifié le pipeline de l'application en conséquence comme il est possible de le voir à la figure 1.1, 1.2 et 1.3.

1.2 Difficultés techniques nouvellement apparues

Malgré le fait que notre équipe avait travaillé avec rigueur afin d'établir un plan solide pour la deuxième partie du projet, cela ne s'est toutefois pas déroulé sans embûches. Tout d'abord, comme mentionné plus tôt, l'estimation de pose s'est révélée être infructueuse autant du point de vue des résultats que d'un point de vue de temps d'exécution. Effectivement, les résultats que nous obtenions en utilisant cette partie du pipeline amenaient plus d'incertitudes et d'erreurs que lorsque nous ne l'utilisions pas. Nous n'avons toutefois pas pris de mesures quantitatives de ces erreurs, car elles étaient tellement flagrantes que nous n'avons pas cru nécessaire de les enregistrer. De plus, cette opération semblait prendre beaucoup trop de temps et le but principal de notre application était de limiter le temps de manipulation du poisson. Nous avons donc décidé de supposer que les poissons seraient plats lors de la prise ce qui est le cas dans la majorité des cas de toute façon. En nous inspirant de la règle de Pareto, nous avons décidé de nous concentrer plus sur les cas les plus classiques afin d'établir une solution la plus robuste sur ces cas précis. Le client était d'ailleurs d'accord avec cela. Le cas considéré était donc les poissons à plat, dans une rotation X dans le plan de la caméra avec un niveau d'occlusion faible à moyen.

Une autre difficulté technique rencontrée par notre équipe a été la classification de l'espèce des poissons. En effet, vers la toute fin du projet, nous avons tenté de rajouter par nous même un module de classification des poissons en tentant d'ajouter des classes à notre modèle et de ne pas prédire seulement réaliser que l'instance de segmentation. Avant de décrire le problème technique, il est nécessaire de savoir que nous ne disposions que de très peu d'images annotées de poissons. Nous avons dû le faire nous-mêmes à la main avec un logiciel nommé Labelme. Cela prend toutefois un temps considérable, car nous devons faire un contour parfait de chacun des poissons. C'est un processus laborieux et nous avons décidé d'annoter environ une vingtaine de poissons, et ce pour chacune des maquettes dont nous disposions. Lorsque nous ne considérions qu'une seule espèce, ces images étaient suffisantes pour obtenir de très bons résultats au niveau de la segmentation d'instances. Cependant, lorsqu'il a fallu que le réseau apprenne à différencier les différents poissons, ce fut beaucoup plus difficile. Notre

hypothèse était que nous n'avions pas suffisamment une grande base de données pour les raisons énumérées précédemment et notre réseau de neurones avait donc une grande difficulté à trouver des caractéristiques de plus bas niveau. En effet, en ayant si peu d'images, il est fort probable que le réseau n'ait pas pu abstraire les caractéristiques spécifiques à chacune des espèces, car plusieurs des maquettes dont nous disposions étaient relativement similaires. Par exemple, le maskinongé et l'omble de fontaine étaient assez semblables dans leur forme ainsi que dans leur palette de couleurs.

2. Description de l'architecture finale du système (Q5.2 – Q5.3 – Q5.4)

2.1 Résumé de ce qu'était le système à la mi-session et ce qu'il est maintenant (Q4.5 et Q4.6)

À la mi-session, le système était une application iOS de base permettant d'envoyer une image à un serveur Flask. L'application mobile nous permettait simplement de prendre le poisson en photo. Par la suite, le traitement de l'image se faisait entièrement sur un ordinateur. Comme nous l'avions mentionné dans le rapport de mi-session, nous étions assez limités par les appareils dont nous disposions, car une seule personne dans notre équipe avait un iPhone avec un LiDAR. Nous avons conclu avec le client qu'il était préférable de faire le traitement sur l'ordinateur afin de pouvoir maximiser la contribution de chacun des membres de l'équipe et de ne pas surcharger certains membres. À ce moment, l'application ne servait qu'à envoyer des photos et ne pouvait pas encore recevoir de réponses. La taille du poisson ainsi que le résultat de la segmentation étaient affichés sur le terminal du serveur en exécution. Aussi, à la mi-session nous avons eu des problèmes avec le format de données du LiDAR alors il fallait se positionner à une distance connue à l'avance lors de la prise de photo.

La première étape de ce processus était de prendre la photo du poisson en s'assurant que celui-ci était à une distance précalculée du téléphone. Ensuite, il fallait encoder l'image dans le format Base64 afin de pouvoir l'envoyer sous forme de requête HTTP/REST à notre serveur. Une fois l'encodage réalisé, l'image était envoyée au serveur.

Lorsque la requête était reçue, une étape de segmentation était effectuée en utilisant un modèle entraîné avec Detectron2. Le modèle que nous avons décidé d'entraîner était Mask-RCNN afin de faire la segmentation d'instances. À ce moment, notre client nous avait conseillé de faire le modèle avec une seule espèce de poisson. Notre modèle avait donc seulement été entraîné avec des images de l'omble de fontaine. Pour notre réseau de base, nous avons utilisé resnet50 qui avait été préalablement entraîné sur MS-COCO, une base de données d'image réalisée par Microsoft dans le but de faire avancer la détection d'objets. En utilisant des poids préentraînés, cela nous a fait sauver un temps considérable en entraînement, car resnet50 était déjà très bon pour encoder les caractéristiques des images. Également, cela nous a permis d'annoter beaucoup moins d'images, car certaines «connaissances» du réseau ont pu être transmises ce qui faisait qu'il était beaucoup plus facile d'apprendre à détecter les poissons.

Ensuite, une fonction de la librairie OpenCV était utilisée pour trouver le plus petit rectangle entourant le masque binaire produit par le modèle de segmentation. Nous avons aussi, en parallèle, une estimation de pose qui était en développement afin de permettre d'estimer la courbure du poisson détectée dans la photo afin d'estimer sa taille complète en pixel.

Par la suite, la longueur de ce rectangle était ensuite utilisée avec d'autres données pour calculer la taille réelle des pixels, qui était ensuite multipliée par la taille en pixel pour obtenir la taille du poisson. Cependant, les données de profondeur devaient être entrées manuellement, car nous avons des difficultés techniques à les obtenir avec le LiDAR.

En date d'aujourd'hui, la base de notre système est globalement restée la même qu'elle était à la mi-session. En effet, nous avons toujours l'application iOS qui prend en photo le poisson, le modèle de reconnaissance du poisson ainsi que le serveur Flask qui permet la communication entre les deux systèmes. Toutefois, ces systèmes ont grandement été améliorés, malgré le fait que certaines idées ont dû être laissées de côté.

L'application iOS a tout d'abord été visuellement embellie et affiche maintenant les informations reliées à la capture du poisson. En plus de permettre à l'utilisateur d'obtenir

la taille du poisson pris en photo, l'application montre maintenant à l'utilisateur l'espèce du poisson, son poids et s'il est permis de le garder ou s'il est nécessaire de remettre le poisson à l'eau en fonction de sa taille. De plus, la technologie du LiDAR a été introduite afin de détecter automatiquement la distance entre le poisson et la caméra. Cet ajout améliore grandement l'estimation de la taille du poisson lors de la transformation de la taille en pixel à la taille en centimètre. En effet, la détection automatique de la distance réduit grandement le facteur de risque humain.

Du côté du modèle de reconnaissance, nous avons amélioré notre modèle de segmentation des poissons afin qu'il considère toutes les maquettes des poissons mises à notre disposition. Nous avons donc répété le processus de prise de photo et de découpage afin de les passer à notre modèle dans le but que celui-ci puisse détecter avec davantage de précision le poisson dans l'image. À l'opposé, l'estimation de la pose permettant de calculer la courbure du poisson a dû être laissée de côté. Étant donné que les deux masques utilisés, soit celui du poisson pris en photo et celui de l'espèce, étaient simplement des nuages de points en 2D, nous nous sommes aperçus qu'il était très difficile de déterminer la courbure actuelle du poisson. Nous avons remarqué une petite amélioration au niveau de la distance de Chamfer, qui mesurait la correspondance entre les deux nuages de points, mais la convergence était très lente et ne donnait pas de bons résultats. L'ajout du paramètre de courbure créait plus de bruit qu'il améliorerait notre modèle, nous avons donc considéré son utilité réelle comme faible et c'est pourquoi nous avons décidé de ne pas l'implémenter dans notre système final.

2.2 Quelques diagrammes de classes ou de modules

Maintenant que nous avons décrit le pipeline de notre application, voici un diagramme qui le décrit visuellement:



Figure 1.1: Partie initiale du pipeline se déroulant sur le téléphone

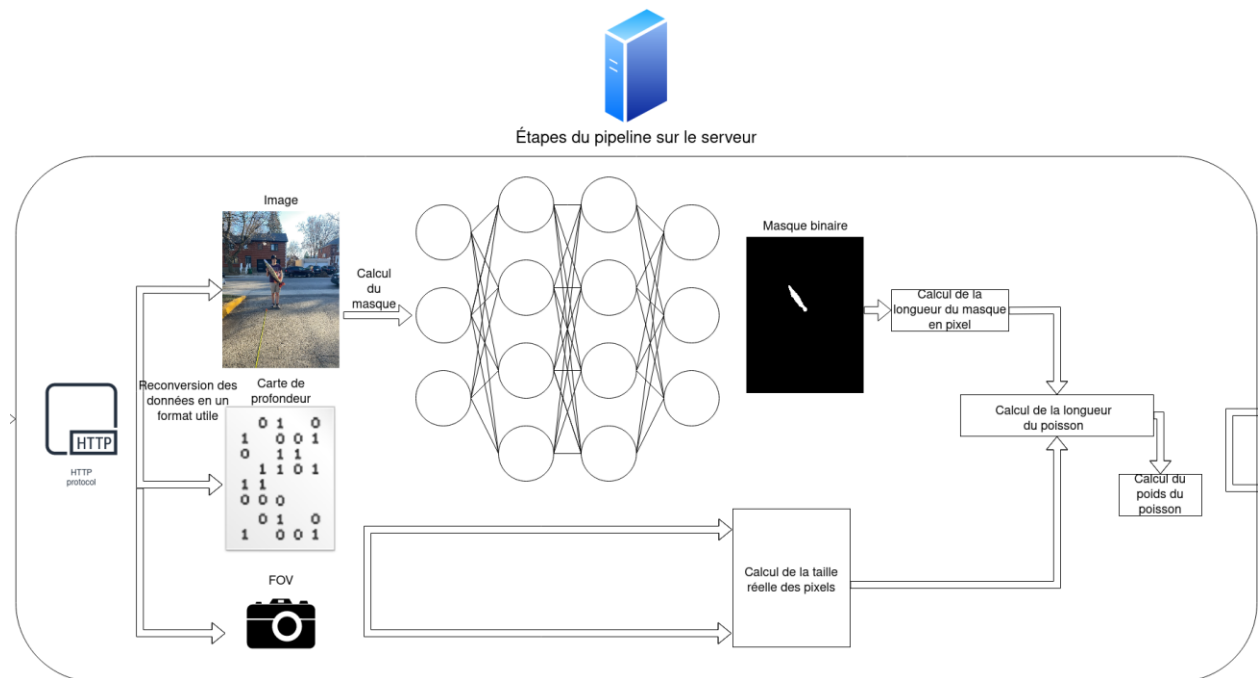


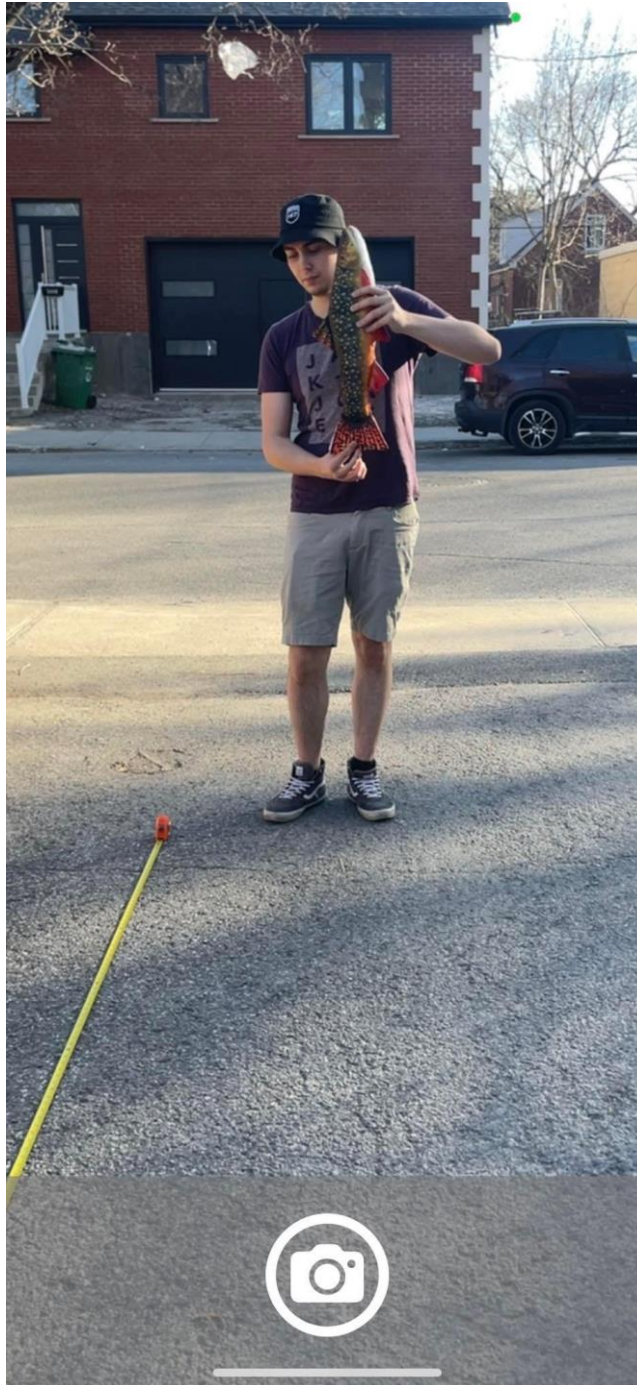
Figure 1.2: Partie du pipeline se déroulant sur le serveur



Figure 1.3: Partie finale et affichage des résultats du pipeline

2.3 Diagramme d'états et/ou d'interface usager

1. Voici un exemple de l'interface au tout début de l'ouverture de l'application:



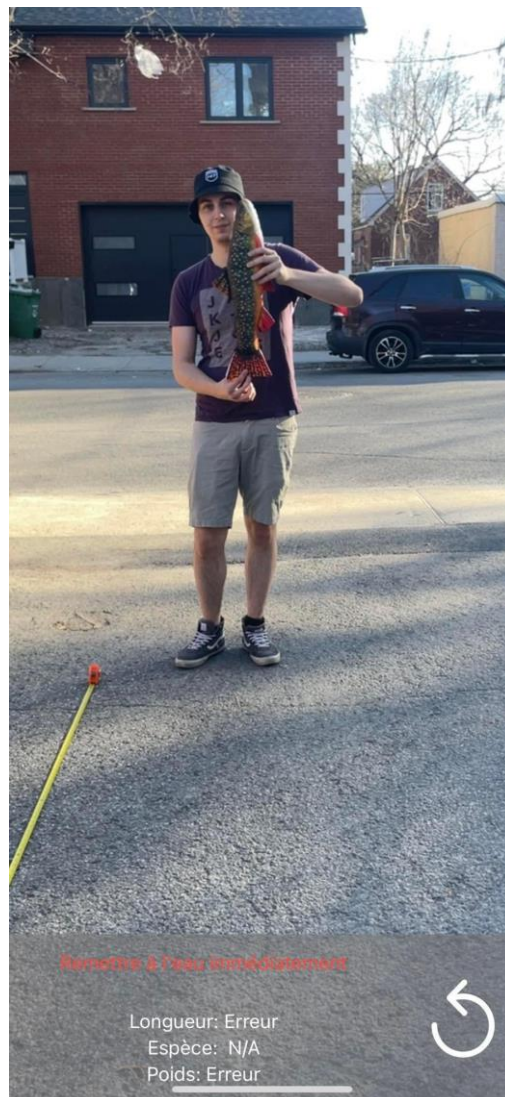
Un bouton de prise de photo est affiché au centre pour permettre la prise de photo.

2. Voici un exemple de l'interface lorsque la photo a été prise et que le poisson est conforme aux exigences:



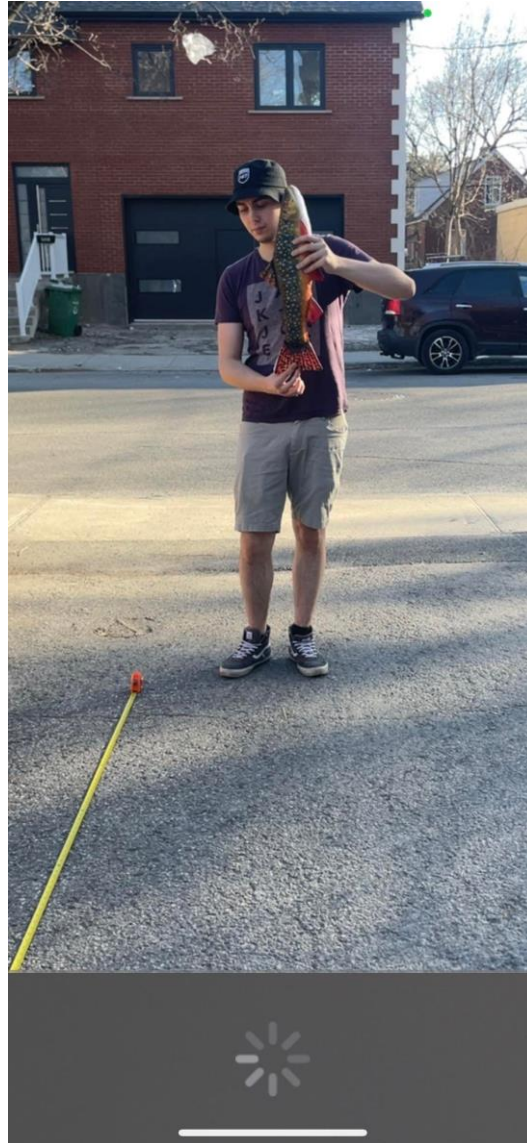
On voit que la mention 'Conformes aux exigences' est affichée. En dessous, la longueur, l'espèce et le poids sont spécifiés. Finalement un bouton pour retourner à la prise de photo est accessible à droite. Il est important de noter que l'espèce est seulement là à titre indicatif et que notre application ne la prédit pas réellement. On voit bel et bien que l'appli indique Brochet alors qu'en réalité nous avons une omble de fontaine. Dans le cas où la taille ne serait pas conforme, le texte sera affiché en rouge et il serait indiqué : « Remettre à l'eau immédiatement. »

3. Voici un exemple de l'interface lorsque le modèle n'arrive pas à détecter de poissons pour une raison quelconque.



L'interface est très similaire à celle montrée lorsque le poisson est conforme sauf que la mention de 'Remettre à l'eau immédiatement' est affichée.

4. Voici l'interface usager lorsque la photo a été prise et qu'on attend la réponse du serveur:



Une animation est alors affichée pour montrer à l'utilisateur que le traitement de l'image est en train d'être effectué et qu'il faut patienter. Après 15 secondes, il sera indiqué : «Server Timeout».

2.4 Interfaces logicielles et/ou matérielles importantes

Pour ce qui est des interfaces matérielles utilisées dans notre projet, nous avons fait appel à l'interface matérielle du LiDAR disponible sur certains modèles d'iPhone et d'iPad. Cette technologie utilise des impulsions laser pour mesurer la distance entre l'appareil et les objets qui l'entourent, ce qui permet de créer des cartes 3D précises de l'environnement en temps réel. Dans notre application, nous avons utilisé le LiDAR pour

mesurer la distance entre la caméra de l'appareil et le poisson que l'utilisateur souhaite mesurer dans une image. Grâce à cette fonctionnalité, notre application peut fournir des mesures précises de la taille et de la longueur du poisson. De plus, le LiDAR améliore également les performances de l'appareil photo en conditions de faible luminosité, ce qui permet d'obtenir des images plus claires et plus nettes.

2.5 Description d'outils, bibliothèques ou cadre de travail (*framework*) utilisés

Tout d'abord, nous avons choisi d'utiliser le cadriciel Flask en Python, car il facilite la création rapide d'applications web en fournissant des outils pour gérer les requêtes HTTP. Nous avons choisi d'utiliser Flask pour plusieurs raisons. Premièrement, Flask est un cadriciel reconnu pour sa flexibilité ainsi que sa facilité d'utilisation. C'était parfait pour notre cas, car dans le futur le serveur ne sera pas réutilisé par l'équipe de notre client. Nous avons donc décidé d'utiliser une solution simple afin d'éviter d'avoir des problèmes lors du développement. Aussi, comme la plupart des bibliothèques d'intelligences artificielles et de visions numériques sont en Python, cela nous facilitait énormément la tâche sans créer d'inconvénients.

Ensuite, nous avons utilisé PyTorch, une bibliothèque à code ouvert d'apprentissage profond développée par Facebook. Elle permet de configurer et d'entraîner des modèles d'apprentissage profond et offre la possibilité d'accélérer le processus d'entraînement grâce à la prise en charge de la parallélisation sur GPU avec des processeurs CUDA. Enfin, cette bibliothèque facilite la sauvegarde et le chargement des modèles, permettant ainsi la réutilisation de modèles préentraînés.

Subséquentement, nous avons employé la bibliothèque NumPy qui apporte un soutien pour les tableaux multidimensionnels et diverses opérations matricielles. Les tableaux de NumPy sont plus performants que les listes Python pour les calculs de grande envergure tels que les opérations tensorielles. NumPy s'intègre aisément avec PyTorch, car il est utilisé pour manipuler des tableaux multidimensionnels, ce qui est pratique pour préparer les données avant l'entraînement des modèles d'apprentissage

profond. De surcroît, les tableaux NumPy peuvent être aisément convertis en tenseurs PyTorch, facilitant l'utilisation conjointe des deux bibliothèques. En somme, l'usage de PyTorch et NumPy permet une préparation des données plus rapide et efficace.

Par ailleurs, la bibliothèque OpenCV a été mise en œuvre pour manipuler des images avant l'entraînement du modèle. Cette bibliothèque offre des outils pour afficher et visualiser les résultats de l'apprentissage profond, tels que la détection d'objets. De plus, OpenCV propose également des outils pour lire et traiter les images ainsi que pour réaliser des transformations d'image.

En ce qui concerne l'application mobile de capture de photos de poissons et d'extraction de profondeur à l'aide du LiDAR, les bibliothèques utilisées sont SwiftUI, Foundation et AVFoundation. Ces outils logiciels sont employés pour prendre une photo avec un iPhone et accéder à la carte de profondeur générée. De plus, ces bibliothèques sont utilisées pour créer certains affichages dans l'application.

SwiftUI est une bibliothèque d'interface utilisateur qui facilite la création d'interfaces utilisateur réactives et modernes pour les applications iOS et macOS. Enfin, AVFoundation est une bibliothèque multimédia iOS qui permet aux développeurs de concevoir des applications de lecture et d'enregistrement de médias, y compris les applications de caméra et de vidéo, nécessaires pour ce projet. Pour ce qui est de Foundation, c'est une librairie à la base de nombreux autres cadres de Swift, y compris UIKit, CoreData et SpriteKit, et fournit des fonctionnalités essentielles telles que la gestion de chaînes de caractères, l'accès aux fichiers et aux réseaux, la gestion de la mémoire, le traitement des dates et heures, la localisation, la sécurité et la cryptographie, la gestion des erreurs, et bien plus encore.

Finalement, nous avons choisi d'utiliser le framework de vision par ordinateur Detectron2 pour la tâche de segmentation des poissons dans notre projet. Ce framework, développé par des ingénieurs de Facebook, propose une solution efficace pour créer des modèles d'apprentissage profond dédiés à la détection et à la segmentation d'objets. Detectron2 est basé sur la plateforme PyTorch, mais il facilite grandement son utilisation en fournissant des outils pour configurer facilement des modèles de détection d'objets

préentraînés, tels que Mask R-CNN. Par conséquent, l'adoption de Detectron2 a grandement simplifié notre processus de développement et amélioré l'efficacité de notre modèle de segmentation des poissons.

2.6 Quelques méthodes de tests appliquées pour valider la solution

Étant donné que notre solution repose sur un pipeline pour détecter la présence de poissons et estimer leur longueur dans nos images, nous n'avons pas vraiment la possibilité de faire des tests unitaires automatisés. Néanmoins, nous avons mené plusieurs tests manuels en utilisant diverses positions et distances des poissons par rapport à la caméra. Nous avons aussi pris en compte la réaction du LiDAR lors de la prise de vue en pleine lumière ou à contre-jour ainsi que lorsque le poisson est très réfléchissant. Nous avons également examiné les différentes sorties des différentes parties du pipeline pour évaluer leur précision et leur fiabilité. Bien que nous n'ayons pas pu automatiser nos tests, notre approche rigoureuse et approfondie des tests manuels nous a permis de garantir la qualité de notre solution. Nous avons également réalisé plusieurs mesures pour pouvoir valider la précision et borner l'incertitude de notre application.

3. Gestion du projet

3.1 Identification des tâches principales du projet

Initialement, le projet visait à développer une solution de développement pour détecter et évaluer la taille réelle des poissons dans une image selon trois grandes sphères: la segmentation des poissons, l'estimation de la pose du poisson et l'estimation de la profondeur du poisson dans l'image. Plus précisément, ces trois sphères avaient, en ordre, l'objectif suivant:

- Détecter le poisson sur la photo prise par l'utilisateur;
- Développer une application qui permet de capturer une image du poisson;

- Évaluer la distance entre le poisson et la caméra dans le but de calculer sa taille réelle;
- Affichage de l'espèce, le poids ainsi que la mesure du poisson.

Afin de segmenter les poissons, diverses tâches sont indispensables. Il est impératif de prendre plusieurs images de chaque espèce de poisson, puis de les annoter pour repérer le contour des poissons. De plus, il est nécessaire de créer un programme d'apprentissage de segmentation d'instances en utilisant Mask R-CNN qui, après un apprentissage rigoureux, nous permettra de déterminer le contour des poissons automatiquement. Cependant, avec la plateforme Detectron2, nous avons la possibilité d'utiliser un modèle préentraîné pour l'appliquer à nos poissons.

Pour estimer la position relative du poisson en 3D, il est d'abord nécessaire d'obtenir le contour du poisson concerné. Celui-ci est déterminé en utilisant l'algorithme de Canny. Une fois le contour obtenu, il faut le faire correspondre avec un masque de l'espèce du poisson. Dans ce contexte, il est nécessaire de développer un programme qui effectue les transformations requises et utilise la fonction de perte de Chamfer afin de déterminer le meilleur rayon de courbure à appliquer au masque afin de faire correspondre les deux nuages de points.

L'évaluation de la profondeur du poisson dans l'image, quant à elle, dépend du développement d'une application iOS utilisant le LiDAR. Tout d'abord, l'application doit être capable de prendre une photo en tenant compte de la profondeur. Ensuite, il est nécessaire de récupérer les données de profondeur de l'image, qui seront envoyées à l'ordinateur contenant l'algorithme d'estimation de la position du poisson. Enfin, ces données doivent être traitées et présentées à l'utilisateur.

Le développement de notre prototype nous a toutefois amené à laisser de côté l'algorithme d'estimation de la pose du poisson, étant donné les résultats non concluants. En effet, le paramètre de courbure converge très lentement et ne donne pas de résultats significatifs. Suite à notre échange avec le client, nous avons décidé de ne pas allouer plus de temps pour développer une solution capable de gérer le cas où le poisson apparaît courbé dans l'image. Nous avons conclu que ce cas peut être considéré comme exceptionnel et peu probable, donc il n'a pas été jugé nécessaire d'y accorder davantage

de ressources de développement. Nous avons alors décidé d'utiliser simplement la largeur du rectangle détectée par notre programme d'apprentissage de segmentation d'instances comme la taille en pixel du poisson détecté.

3.2 Répartition des tâches et responsabilités dans l'équipe

Pour collaborer efficacement et atteindre les objectifs fixés, il est essentiel de répartir les tâches et les responsabilités au sein d'une équipe. Dans notre équipe de recherche, nous avons opté pour une division des trois tâches principales en trois équipes de deux personnes chacune pour une meilleure répartition des responsabilités. La première équipe avait comme objectif de s'occuper de la segmentation de l'image, la deuxième équipe se chargeait de l'estimation de la pose et la troisième équipe avait pour mission de déterminer la profondeur. Cette méthode de travail en équipe permet à chaque membre de se focaliser sur une tâche précise et de coopérer étroitement avec son partenaire pour optimiser l'efficacité et les résultats de l'équipe. Les membres de l'équipe ont également acquis certains rôles suite à leur travail dans les sous-équipes:

Tableau 1. Rôles et fonctions de gestion des membres de l'équipe

Rôles	Félix G	Wissale	Félix D	Guillaume	Steven	Émile
Gestion du projet et communication	✓	✓				
Développeur	✓	✓	✓	✓	✓	✓
Expert AI	✓		✓			✓
Testeur	✓	✓		✓	✓	

Malgré la division en trois équipes distinctes, nous avons maintenu une collaboration étroite pour nous soutenir mutuellement. Les membres de chaque équipe étaient incités à communiquer avec les autres équipes pour garantir que chaque étape du processus se déroule sans problème. Cette approche garantit que toutes les équipes

travaillent ensemble de manière efficace pour atteindre leur objectif commun et fournir des résultats précis et fiables.

Afin de garantir la livraison des livrables, il est essentiel de mettre en place une bonne gestion de projet. Au sein de notre équipe, nous avons opté pour les méthodologies Scrum et Agile, qui offrent une approche itérative et une grande capacité d'adaptation aux changements en cours de route. Dans un premier temps, nous nous sommes réunis pour comprendre les exigences du projet et déterminer les livrables clés. À partir de cette réunion, nous avons établi un plan de développement initial en identifiant les objectifs, les tâches à réaliser, les délais estimés, les ressources nécessaires et les responsabilités de chaque membre de l'équipe. Chaque vendredi matin, nous planifions notre sprint pour la semaine à venir, en identifiant les tâches à accomplir et en affectant les membres de l'équipe concernés. La gestion de ce processus est assurée par la plateforme Trello. Chaque membre de l'équipe travaille une vingtaine d'heures sur ses tâches assignées pendant la semaine et fait un rapport à la fin du sprint, le jeudi soir, pour discuter des résultats et déterminer si les objectifs ont été atteints. En outre, deux réunions par semaine de scrum sont organisées pour discuter de l'avancement, des problèmes rencontrés et des problèmes potentiels qui pourraient entraver la progression de l'équipe.

Dès le début du projet et à chaque début de sprint, notre équipe accorde une grande importance à la gestion des risques. Nous avons identifié les risques afin de mettre en place des stratégies visant à les réduire. Nous avons également prévu plusieurs solutions alternatives pour faire face à d'éventuels problèmes. Il convient de souligner que ce projet ne se limite pas à l'aspect informatique, mais inclut également des aspects de recherche impliquant des hypothèses qui doivent être testées avant d'être validées en tant que solutions utilisables et fonctionnelles pour le projet.

Nous avons mis en place des pratiques de communication régulières avec le client dès le début de notre collaboration avec Jonathan Jalbert et Yves Paradis. Chaque vendredi à 9h00, des réunions sur la plateforme Teams étaient organisées pour présenter les avancées du projet. Afin de faciliter la communication des progrès hebdomadaires du

projet, le client a choisi d'utiliser des présentations PowerPoint. Ces réunions hebdomadaires nous ont non seulement permis de tenir le client informé de l'évolution du projet, mais également d'adapter nos approches en fonction de ses commentaires et de ses besoins. Nous avons ainsi été en mesure de répondre rapidement à toutes les incertitudes ou modifications éventuelles.

3.3 Principales difficultés de gestion rencontrées durant la deuxième partie du projet

Au cours de la deuxième partie de ce projet, plusieurs difficultés de gestion ont été rencontrées, principalement autour de la gestion du temps et des ressources. Trois domaines clés ont été identifiés comme étant les principaux points de blocage: l'estimation de pose, l'utilisation du LiDAR pour trouver la profondeur et les ressources à notre disposition.

Tout d'abord, l'estimation de pose a été un défi majeur. Bien que des ressources considérables aient été consacrées à cette tâche, le résultat final n'a pas été à la hauteur des attentes. Les délais de développement ont été plus longs que prévu et les résultats n'ont pas été aussi précis qu'espérés. L'approche globale de notre projet limitait notre capacité à exploiter pleinement le potentiel de notre idée initiale. Bien que nous ayons pu atteindre un niveau de précision moyen avec le paramètre de courbure, nous n'avons pas pu aller au-delà de cette limite pour obtenir des résultats encore plus précis. Malgré tout, notre idée initiale reste pertinente et pourrait être un point de recherche potentiel dans le futur. En effet, des améliorations futures dans l'estimation de pose pourraient grandement améliorer notre prototype et notre capacité à capturer des mouvements en temps réel.

De plus, l'utilisation de LiDAR pour trouver la profondeur a également été une difficulté de gestion majeure. Comme la technologie était essentielle pour le projet, il a été accordé beaucoup de temps et de ressources pour cette tâche. Nous n'avions pas d'autre choix que d'y accorder une grande importance puisque la possibilité de calculer la profondeur reposait sur cette technologie. Malgré les difficultés rencontrées pour extraire la donnée, nous avons réussi à surmonter nos lacunes de connaissances et une

fois cette étape franchie, la collecte de données et les calculs pour obtenir la taille réelle se sont déroulés sans encombre.

Le dernier défi majeur rencontré lors de la deuxième partie du projet a été le manque de ressources matérielles à notre disposition. En effet, nous avons seulement quelques maquettes de poisson pour entraîner notre modèle et obtenir des résultats. Cette situation a donc créé des limites importantes, car il était plus difficile de reproduire des conditions réelles avec différentes tailles de poisson, des déformations ou encore des reflets. Également, nous n'avions qu'un seul iPhone avec la technologie LiDAR, ce qui a ralenti considérablement le développement de l'application. En plus, seul un des membres de l'équipe avait un Mac qui est nécessaire pour développer des applications mobiles iOS. Ces deux membres devaient donc constamment se déplacer pour pouvoir travailler en équipe. Cela a demandé beaucoup d'adaptation et de résilience de leur part, car ils étaient responsables de toute la partie interface utilisateur du projet et ils étaient seulement deux. Heureusement, tout s'est quand même assez bien déroulé, mais il est évident que si tous les membres avaient eu accès à un ordinateur portable Mac ainsi qu'un téléphone équipé d'un LiDAR, nous aurions pu accomplir encore plus de choses. Malgré cela, nous avons eu beaucoup de bons commentaires de nos clients et ces derniers semblaient très satisfaits de nos avancements.

Pour nous adapter à cette situation, nous avons dû trouver des alternatives pour tester notre modèle. Nous avons ainsi décidé de tester notre modèle sur des objets de plus petites tailles, tels qu'une bouteille d'eau Gatorade verte et orange, puisque ces couleurs ressemblent sensiblement à celles d'un poisson. Nous avons également utilisé des collants réfléchissants et de petits miroirs afin de simuler un reflet d'eau. Enfin, nous avons pris des photos à contre-jour pour faire comme si nous étions sur un lac.

Bien que ces alternatives aient permis de poursuivre le projet malgré le manque de ressources, cela a également créé des limites dans notre modèle. En effet, il n'a pas été possible de reproduire toutes les conditions réelles que l'on peut retrouver en milieu naturel, ce qui a peut venir impacter la précision et la fiabilité de notre modèle.

Également, dû au nombre de maquettes limitées et au temps, il était impossible d'annoter un grand nombre d'images de poissons pour avoir un modèle qui soit assez entraîné pour reconnaître et différencier les différentes espèces de poissons. Nous avons tenté de pallier en augmentant le temps d'entraînement, mais sans succès. Il était toujours impossible pour notre modèle d'avoir une base d'images assez conséquente pour reconnaître les différentes espèces de façon précise et efficace.

4. Conclusion

4.1 Retour sommaire sur le travail complété et ce qui n'a pu être réalisé (Q3.5)

En date de la remise du rapport final, nous sommes fiers de présenter notre prototype qui permet de prendre une photo d'un poisson et de fournir à son utilisateur des informations précieuses sur l'espèce, la taille et le poids du poisson en question. De plus, notre application est capable de déterminer si une remise à l'eau est nécessaire ou pas, ce qui peut avoir un impact considérable sur la conservation des espèces marines.

Pour être en mesure de déployer ce prototype final, notre équipe a dû se concentrer sur trois principales phases de développement:

- Reconnaître le poisson dans l'image;
- Déterminer la profondeur du poisson dans l'image;
- Transmettre les données du téléphone à l'ordinateur.

La reconnaissance du poisson est une étape cruciale pour l'application. Pour y arriver, notre équipe a dû entraîner un modèle de segmentation d'image en utilisant la technologie de pointe de Detectron2. Grâce à cette technologie, le modèle est en mesure de repérer le poisson dans l'image transmise par l'utilisateur. Cette technique de segmentation d'image a nécessité des heures d'annotation d'images par les membres de

notre équipe, afin de garantir que le modèle soit capable de reconnaître et d'identifier le poisson avec une grande précision. Le modèle est alors en mesure de trouver le poisson dans l'image transmise et ainsi l'entourer d'un rectangle permettant de déterminer sa longueur.

La détection de la profondeur du poisson est un processus crucial pour estimer la taille réelle du poisson capturé. Pour ce faire, notre équipe a utilisé la technologie de télémétrie LiDAR disponible sur les téléphones récents. Cette technologie fonctionne grâce à un laser infrarouge qui émet des ondes lumineuses et mesure le temps qu'il faut pour que la lumière soit renvoyée. En analysant la différence de temps entre l'émission et la réception du signal, le téléphone peut déterminer la distance entre le poisson et la caméra. Cette mesure permet ensuite de ramener la taille du rectangle calculé par la segmentation sur la caméra et ainsi trouver la taille réelle du poisson. Grâce à cette technique avancée, notre application est capable de fournir des mesures de taille précises pour aider les pêcheurs à évaluer la taille des poissons qu'ils capturent.

Afin d'être en mesure de développer notre prototype sur un ordinateur et un téléphone, une communication entre l'application iOS et le terminal nous était nécessaire. Pour ce faire, nous avons utilisé un serveur Flask, un outil qui facilite l'échange de données entre les deux systèmes. Grâce à cette technologie, notre application est en mesure d'envoyer facilement l'image prise par l'utilisateur au terminal, qui effectuera les calculs nécessaires pour déterminer la taille du poisson. Le serveur est également capable de renvoyer rapidement les données analysées par le terminal à l'application, permettant ainsi à l'utilisateur de connaître rapidement les informations importantes concernant le poisson capturé. La mise en place d'une communication fluide entre les différents éléments du système est essentielle pour assurer une expérience utilisateur satisfaisante et un fonctionnement efficace de notre prototype. Ainsi, nous avons investi beaucoup de temps et d'efforts dans la mise en place d'une architecture de communication solide et efficace pour assurer la qualité de notre produit final.

L'objectif final de ce projet est toutefois de rendre l'application entièrement autonome en éliminant le besoin d'un serveur et d'un terminal. La direction future du projet devrait donc être d'implémenter les calculs directement sur le téléphone. Cette amélioration permettra aux utilisateurs de capturer des photos de poissons et d'obtenir immédiatement toutes les informations importantes, sans avoir besoin d'un équipement supplémentaire. Cette évolution de l'application la rendra encore plus pratique et utile pour les pêcheurs amateurs et professionnels.

Dans les phases initiales du développement, lorsque nous étions encore en phase de recherche afin d'établir un plan solide, nous avons lu un article qui ressemblait grandement à notre objectif et qui présentait une phase de déterminer la courbure du poisson afin de déterminer sa taille à plat. L'équipe avait donc décidé d'ajouter un module d'estimation de pose dans le développement du prototype. Nous avons donc conçu un algorithme qui utilisait un paramètre de courbure sur deux masques, un masque du poisson pris en photo et un de l'espèce du poisson, dans le but de faire converger la distance de Chamfer, une mesure de distance entre deux ensembles de points dans un espace euclidien. Malheureusement, les résultats obtenus n'étaient pas très concluants par rapport à nos attentes initiales et ne présentaient pas une réelle amélioration sur la taille calculée. Comme le temps nous était limité et que les recherches et le développement de cet algorithme consommaient plus de ressources qu'elles n'apportaient de résultats concluants, l'équipe a décidé de laisser de côté ce module.

Il est aussi important de mentionner que le développement de ce prototype a été réalisé à partir de zéro. L'équipe n'avait aucune compétence préalable en matière de développement d'applications de ce type et aucune structure de base ne nous avait été fournie. Nous avons dû apprendre de nouvelles compétences techniques en cours de route et acquérir des connaissances sur les technologies telles que Detectron2 et LiDAR pour atteindre nos objectifs de développement. Malheureusement, en raison de la courte durée du projet et du matériel à notre disposition, nous n'avons pas été en mesure de travailler directement sur l'intégration du prototype sur le téléphone, ce qui aurait été notre objectif final. Cependant, nous avons pris en compte cet objectif final dans la conception

du prototype et nous avons optimisé les algorithmes pour une exécution efficace sur un téléphone portable.

4.2 Causes des succès et difficultés rencontrées (Q3.6)

Les difficultés rencontrées durant le développement étaient principalement liées à notre niveau de connaissance des diverses technologies et des orientations de développement existantes. En effet, aucun membre de l'équipe n'avait d'expertise concrète des meilleures techniques pour identifier un objet dans une image et personne ne connaissait les différents outils permettant de détecter la profondeur d'un objet dans une image ou encore.

La première difficulté est survenue lors de la création du modèle de segmentation d'image, plus précisément lors du choix de cadriciel de développement pour notre modèle de segmentation d'instance. Plusieurs choix s'offraient à nous:

- TensorFlow, une librairie sur le thème de l'apprentissage profond permettant la création de modèles de segmentation d'instances développés par Google;
- PyTorch, une autre librairie portant sur le même sujet que TensorFlow, mais réputée comme ayant une courbe d'apprentissage moins brusque et plus adaptée à la recherche et l'expérimentation.
- Detectron2, une librairie dédiée à la vision par ordinateur supportant plusieurs dérivées des réseaux de neurones convolutions basées sur les régions plus communément nommées « Region based Convolutional Neural Networks (R-CNN) »

Notre choix s'est finalement arrêté sur Detectron2. Ce framework, basé sur PyTorch, offre une architecture modulaire qui permet aux utilisateurs de personnaliser et d'ajouter facilement de nouvelles fonctionnalités. Nous en avons donc conclu qu'il était celui qui répondait le plus à nos exigences. En plus, Detectron2 offre plusieurs modèles de segmentation d'instances préentraînés. À un certain moment du projet, nous avons même entraîné un modèle de détection d'objet avec cette plateforme. Nous avons cru que c'était intéressant, puisque la base de données de nos clients était déjà annotée pour

cette tâche. Malheureusement, nous avons découvert que les modèles de détections d'objets n'étaient pas réellement adaptés à notre problème, car il est important d'avoir une mesure très précise. En utilisant seulement la détection d'objets, nous n'aurions pas pu obtenir une mesure aussi précise que nous avons réussi à avoir avec les masques de segmentation.

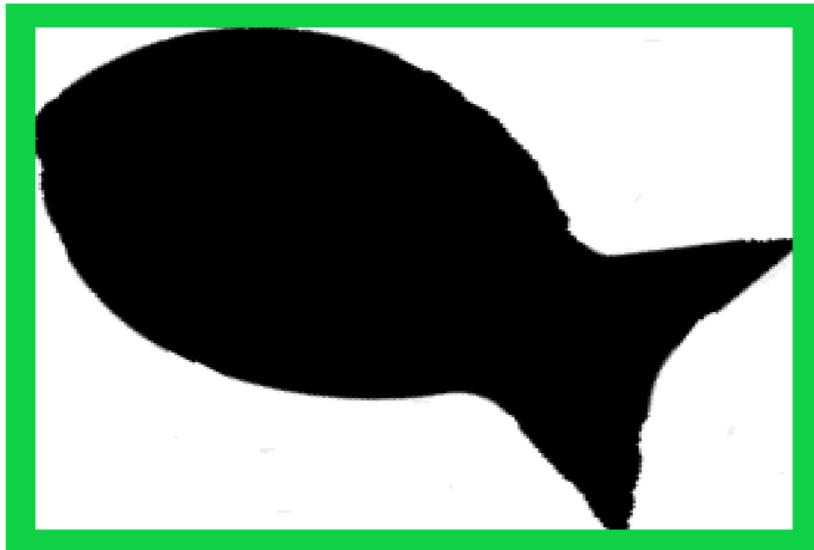


Figure 2: Problème de précision avec la détection d'objets

Comme on peut le voir à la figure 2, si on utilise seulement la détection d'objet, nous n'avons pas accès à l'orientation du poisson ce qui affecte énormément la précision. En utilisant la segmentation comme avec Mask-R-CNN, nous pouvons orienter le rectangle et ainsi trouver le plus petit rectangle autour du masque pour trouver une longueur encore plus précise. Ainsi, l'orientation du poisson dans le plan perpendiculaire à la caméra du téléphone n'a aucun impact sur la mesure.

La seconde difficulté rencontrée est venue de l'estimation de la pose du poisson. Concrètement, nous n'arrivions pas à implémenter un paramètre de courbure dans notre module de comparaison de masque. Nous avions initialement prévu d'utiliser la taille en pixel provenant du module d'estimation de la pose. Cependant, nous avons rapidement réalisé que la qualité des résultats n'était pas suffisante pour permettre une reconnaissance précise. Malgré nos efforts pour affiner les paramètres de notre algorithme de reconnaissance, nous n'avons pas réussi à atteindre les résultats

souhaités. En fin de compte, nous avons conclu qu'il était plus judicieux de mettre de côté le développement de ce module.

La dernière difficulté rencontrée provient de l'utilisation de la technologie LiDAR. Puisque cette technologie est plutôt récente sur le marché, la documentation expliquant son fonctionnement n'est pas très développée. Les avancements de la détection de la profondeur dans l'image ont donc été relativement lents par rapport au développement des autres sphères du prototype. Plus spécifiquement, la difficulté provenait de la récupération de la matrice de profondeur du LiDAR. Visuellement, nous étions en mesure de voir sur l'application qu'elle était en mesure de détecter les objets à de diverses profondeurs, nous n'arrivions tout simplement pas à aller chercher ces profondeurs. En redoublant d'efforts et en cherchant énormément sur le sujet, l'équipe a finalement réussi à aller extraire ces données permettant de calculer la taille réelle des pixels.

Le succès de notre projet repose sur la répartition efficace des tâches et des responsabilités au sein de notre équipe de recherche ainsi que sur l'attitude positive et impliquée de chaque membre de l'équipe. En divisant les tâches principales en trois équipes de deux personnes chacune, nous avons permis à chaque membre de se concentrer sur une tâche spécifique tout en travaillant étroitement avec son partenaire pour optimiser l'efficacité de l'équipe dans son ensemble. De plus, dès le début de notre projet, nous avons organisé au mieux de nos compétences un plan de travail sur Trello en séparant chacune des tâches dans nos sous-équipes. Cela nous a permis de coordonner nos différentes tâches en nous assurant de commencer en priorité les tâches bloquantes. Ainsi les sous-équipes n'étaient pas bloquées par les autres ce qui nous a permis d'avancer très rapidement dans le projet.

Cette approche de travail en équipe a permis à chaque membre de se spécialiser dans un domaine spécifique, ce qui a augmenté la qualité du travail accompli et réduit les risques d'erreurs ou de malentendus. Les membres de chaque équipe étaient toutefois incités à communiquer régulièrement avec les autres équipes pour garantir que chaque étape du processus se déroulait sans problème. Cette collaboration étroite a renforcé la

communication et la coordination entre les membres de l'équipe, ce qui a permis d'optimiser les résultats obtenus.

La mise en place d'une gestion de projet efficace a également été cruciale pour le succès de notre projet. En utilisant les méthodologies Scrum et Agile, nous avons pu adopter une approche itérative et flexible qui nous a permis de nous adapter rapidement aux changements en cours de route. L'identification des objectifs, des tâches à réaliser ainsi que des responsabilités de chaque membre de l'équipe en début de projet a été grandement bénéfique au bon déroulement.

La communication régulière avec le client au moyen de réunions hebdomadaires et de présentations PowerPoint nous a permis d'adapter nos approches en fonction des commentaires et des besoins, et de modifier rapidement notre stratégie de développement.

Enfin, l'implication continue de chaque membre de l'équipe et la bonne foi de chacun a permis de créer un environnement cohésif et de garder une ambiance agréable tout au long du projet. Chaque membre s'impliquait non seulement dans le développement de son module, mais aussi dans la résolution des problèmes des autres sous-équipes. Bref, l'ingéniosité de chaque membre est ce qui a permis une résolution de problèmes rapides et un développement de prototype amusant.

En résumé, la répartition efficace des tâches et des responsabilités, la mise en place d'une gestion de projet efficace, la communication régulière avec notre client et l'attitude des membres ont été les principales causes du succès de l'avancement de notre projet. Grâce à ces éléments, nous avons été en mesure de maximiser notre efficacité et d'optimiser les résultats obtenus.

5. Apprentissage en continu (Q12)

5.1 Lacunes identifiées dans ses savoirs et savoir-faire durant le projet

Guillaume:

Une de mes lacunes était ma connaissance des librairies, et du langage de développement d'iPhone. Apple a créé plusieurs librairies dont SwiftUI, Foundation, AVFoundation, et d'autres. Aussi, Apple utilise Swift comme langage de programmation. Je n'avais aucune connaissance préalable sur ces sujets avant de commencer le projet, autre que des connaissances sur le développement en interface usager.

Steven:

Lorsque j'ai commencé à travailler sur le projet, j'ai réalisé que mes connaissances impliquant l'apprentissage automatisé étaient plutôt limitées, ce qui représentait une lacune pour moi. J'avais tendance à vouloir me lancer rapidement dans le développement sans prendre le temps d'effectuer des recherches approfondies, ce qui pouvait causer des erreurs facilement évitables dans mon code. Comme mes connaissances par rapport à certaines technologies étaient moins développées que certains de mes collègues, j'avais l'impression d'être moins efficace, ce qui m'a poussé à vouloir rédiger du code plus rapidement, créant parfois des erreurs idiotes.

Wissale:

Tout d'abord, puisque je n'avais pas d'expérience concrète en intelligence artificielle, cela a limité ma capacité à contribuer pleinement à la partie du projet impliquant l'IA. Bien que j'ai pris des mesures pour combler cette lacune en lisant des documents techniques pertinents, je reconnais qu'il me manque encore une expérience pratique pour être totalement à l'aise dans ce domaine. Surtout puisque les plus compétents dans le domaine ont pris en charge cet aspect du projet.

De plus, bien que j'ai une expérience de développement, je n'avais aucune connaissance en ce qui a trait au développement iOS. Je suis consciente que mes compétences dans ces langages ne sont pas encore à un niveau avancé. Bien que je sois capable de coder des solutions de base, il est possible que je rencontre des difficultés lors de la résolution de problèmes plus complexes, ce qui pourrait ralentir le développement du projet.

En somme, bien que j'étais enthousiaste à l'idée de travailler sur ce projet, je dois reconnaître mes limites en matière de connaissances et compétences techniques. Cependant, je suis convaincu que j'ai pu être en mesure de combler ces lacunes grâce à mon désir d'apprendre et à mon engagement envers le projet.

Felix D:

Dès le début du projet, je savais que je n'avais pas beaucoup d'expériences sur le développement de système d'intelligence artificielle. C'est pourquoi j'ai voulu pousser ma compréhension du sujet en prenant un projet final qui incorpore ces intelligences, car c'est une technologie qui est en grand essor. J'ai donc pu comprendre les bibliothèques Python PyTorch et NumPy qui sont grandement utilisées dans les industries. Également, j'ai pu voir de façon concrète le lien entre l'entraînement d'un modèle et son utilisation afin de prédire des résultats sur un traitement d'images.

Felix G:

Une de mes lacunes que j'ai observées durant le projet est mon manque de connaissances en nuages de points en 3 dimensions. Dans mes expériences professionnelles, j'ai acquis beaucoup de connaissances sur le traitement d'image en 2 dimensions, mais c'était la première fois que je devais traiter de l'information en 3 dimensions. Je ne connaissais pas vraiment d'algorithmes existants dans l'enregistrement des nuages de points, un sujet en informatique ayant pour but d'appliquer des transformations à un nuage de point pour le faire correspondre à un autre. Ainsi, en ayant peu de connaissances, cela nous a fait perdre un peu de temps dans la partie de l'estimation de pose des poissons.

Emile:

La plus grande lacune que j'ai identifiée durant le projet est mon manque de connaissances des bibliothèques PyTorch et NumPy. En effet, j'avais déjà fait de la programmation en Python lors d'un stage, mais je n'avais jamais utilisé de bibliothèque d'apprentissage automatique ou de traitement de données. J'ai passé beaucoup de temps à essayer d'implémenter un modèle lors des premières semaines du projet sans connaître NumPy et PyTorch ce qui a demandé un effort considérable.

5.2 Méthodes prises pour y remédier

Guillaume:

Pour remédier à cette lacune dans ma connaissance des bibliothèques et du langage de développement d'iPhone, j'ai entrepris de rechercher de la documentation et de regarder des vidéos sur ces sujets. J'ai passé du temps à étudier les différentes bibliothèques disponibles sur iOS, notamment Swift et Foundation, ainsi que le langage de programmation Swift. En regardant des tutoriels et en suivant des exemples de code, j'ai commencé à acquérir une compréhension de base de ces sujets. Grâce à ces efforts, j'ai pu améliorer ma compréhension des bibliothèques et du langage de développement d'iPhone, et je suis devenu plus confiant dans ma capacité à travailler efficacement sur des projets iOS à l'avenir.

Steven:

Pour surmonter mes lacunes en matière de connaissances techniques, je me suis obligé à faire un plan de mes idées. J'ai tout d'abord noté mes objectifs, en veillant à ce qu'ils soient clairs et cohérents. Ensuite, j'ai pris le temps de faire des recherches approfondies sur les technologies et les outils dont j'avais besoin pour atteindre mes objectifs. J'ai lu plusieurs articles et forums et j'ai visionné des vidéos en ligne pour acquérir une compréhension plus approfondie de ces sujets. Ce processus m'a permis d'améliorer considérablement mes compétences de développement et ma méthodologie de travail, et donc devenir plus efficace dans la rédaction de code.

Wissale:

Tout d'abord, pour améliorer mes compétences en matière d'intelligence artificielle, j'ai lu plusieurs articles concernant l'IA ainsi que sur les outils utilisés. De plus, je posais plusieurs questions à mes partenaires plus expérimentés. En ce qui concerne mes compétences en développement iOS, j'ai regardé plusieurs vidéos en ligne ainsi que lu des discussions sur des forums ou de la documentation d'Apple elle-même pour améliorer ma compréhension. Dans l'ensemble, je suis restée ouverte à apprendre tout au long du projet pour remédier à mes limitations. En travaillant dur pour améliorer mes compétences et ma compréhension du génie logiciel, j'ai pu être en mesure de contribuer plus efficacement au projet et de développer mes compétences pour les projets futurs.

Felix G:

Pour remédier à mon manque de connaissances, j'ai d'abord fait plusieurs recherches sur Internet pour trouver des techniques permettant de réaliser cette tâche. J'ai donc lu beaucoup sur les forums tels que « StackOverflow » et « Medium » afin d'en apprendre davantage et découvrir de nouvelles méthodes. Cela m'a permis de comprendre plusieurs aspects comme la fonction de perte de Chamfer qui est très utile dans ce type de tâche. Également, cela m'a fait approfondir mes connaissances sur certains acquis que j'avais notamment la descente de gradient stochastique. De plus, j'ai découvert plusieurs algorithmes comme la méthode itérative du point connue sous le nom de « Iterative Closest Point (ICP) » dans la communauté scientifique. J'ai également pu apprendre sur plusieurs algorithmes qui s'inspirent de cette solution, plus particulièrement des algorithmes qui introduisent des transformations non rigides ce qui était essentiel pour notre projet

Felix D:

Tout au long du projet, j'ai dû faire beaucoup de recherches afin de me remettre à niveau. Beaucoup de temps pour ma part a été passé à lire sur la documentation de PyTorch et j'ai dû poser beaucoup de questions sur les différentes librairies, notamment à Félix G. J'ai aussi recherché et lu beaucoup d'articles sur le sujet afin de comprendre les différentes fonctionnalités et les différents formats de données souvent utilisés pour l'apprentissage machine.

Emile:

Afin de me familiariser avec les librairies NumPy et PyTorch, j'ai dû lire beaucoup de documentation en ligne sur leur utilisation. En effet, j'ai effectué plusieurs recherches au début du projet afin de comprendre les concepts importants liés aux deux librairies. J'ai également lu plusieurs articles expliquant comment entraîner un modèle de base avec PyTorch. Cependant, il était plus difficile d'intégrer certains tutoriels à notre code puisque notre problématique était un peu plus complexe.

5.3 Identifier comment cet aspect aurait pu être amélioré

Guillaume:

Pour améliorer ma compréhension des librairies et du langage de développement d'iPhone, j'aurais pu explorer davantage les ressources en ligne disponibles, telles que les forums de développeurs, les blogues et les sites web de documentation. En outre, j'aurais pu chercher des projets code ouvert sur GitHub qui utilisent les librairies et le langage de développement d'iPhone pour apprendre de manière pratique. Enfin, j'aurais pu envisager de suivre un cours ou une formation en ligne sur les librairies et le langage de développement d'iPhone pour obtenir une formation structurée et approfondie.

Steven:

Dans le but d'être plus efficace dès le début du projet, il m'aurait fallu me familiariser avec les différentes technologies utilisées préalablement. Si j'avais créé un projet personnel pour explorer les technologies clés avant de commencer le projet, j'aurais eu une meilleure compréhension de leur utilisation et de leurs avantages et inconvénients, ce qui m'aurait permis de faire des choix plus éclairés en matière de conception et de développement. Une connaissance préalable des différentes technologies m'aurait aussi permis de démarrer plus rapidement et plus efficacement. J'aurais donc non seulement été productif dès le départ, mais j'aurais aussi été en mesure d'être une source de réponse pour mes collègues.

Wissale:

Pour éviter la perte de temps, j'aurais pu prendre des formations en même temps que le projet ou dès que j'ai su que nous allions développer en Swift. J'aurais aussi pu demander des conseils à des gens plus expérimentés dans le domaine pour me donner des pistes et des conseils. Je dirais que j'aurais pu faire exactement la même chose pour l'AI, mais en me créant un petit projet personnel ou j'aurais pu tester individuellement comment chaque outil fonctionne en regardant sur Youtube pour me guider ou bien en lisant des forums.

Felix G:

Afin d'avoir des connaissances plus solides, j'aurais dû lire davantage d'articles scientifiques sur l'enregistrement des nuages de points. En effet, si je m'étais basé plus sur la littérature scientifique j'aurais pu mieux comprendre ces algorithmes. Également, cela m'aurait aussi probablement fait sauver du temps, car je n'aurais pas eu à essayer certains des algorithmes si je les avais mieux compris, car j'aurais réalisé qu'ils ne sont pas pertinents dans notre cas. De plus, j'aurais probablement sauvé beaucoup de temps lors de l'implémentation de ces algorithmes, car je les aurais beaucoup mieux compris.

Felix D:

Premièrement, j'aurais dû suivre des cours de polytechnique sur l'intelligence artificielle. En effet, je n'avais aucune idée de la procédure et du fonctionnement de réseaux convolutifs. Deuxièmement, plusieurs cours de traitement d'images sont disponibles en option dans le cursus. Pour accélérer grandement mon apprentissage des notions, j'aurais dû suivre ces cours afin de comprendre plus les opérations qui sont possibles sur l'analyse et le traitement d'images numériques. Finalement, il est toujours bon de lire des articles scientifiques sur le sujet pour s'ouvrir l'horizon et les connaissances à des concepts que je n'aurais jamais pensé m'être utiles.

Emile:

Afin de mieux commencer le projet, j'aurais pu suivre un cours en ligne d'apprentissage automatique avec PyTorch et NumPy afin de me familiariser davantage avec les bibliothèques. Lire la documentation et faire des recherches en ligne n'est pas souvent la meilleure option lorsqu'un n'a pas beaucoup d'expérience avec un sujet. Par ailleurs, j'aurais également pu aller voir les communautés en ligne de développeurs et de scientifiques des données travaillant avec NumPy et PyTorch. J'aurais pu consulter les forums de discussion en ligne afin de poser des questions.

6. Références utilisées

[1] K. He, G. Gkioxari, P. Dollár, R. Girshick (2018) Mask R-CNN. [En ligne].

Disponible : <https://paperswithcode.com/paper/mask-r-cnn> Consultée le 28 janvier 2023

[2] J. Mei, J. Hwang, S. Romain, C. Rose, B. Moore, K. Magrane (2021) Absolute 3D Pose Estimation and Length Measurement of Severely Deformed Fish from Monocular Videos in Longline Fishing. [En ligne].

Disponible : <https://arxiv.org/abs/2102.04639#:~:text=9%20Feb%202021%5D-,Absolute%203D%20Pose%20Estimation%20and%20Length%20Measurement%20of%20Severely%20Deformed,Monocular%20Videos%20in%20Longline%20Fishing&text=M,ocular%20absolute%203D%20fish%20pose,deformation%20during%20the%20catching%20process>. Consultée le 13 janvier 2023.

[3] P. Chase, K. Clarke, A. Hawkes, S. Jabari1, J. Jakus (2022) APPLE iPHONE 13 PRO LIDAR ACCURACY ASSESSMENT FOR ENGINEERING APPLICATIONS.

[En ligne]. Disponible : <https://conferences.lib.unb.ca/index.php/tcrc/article/view/645/113>

Consultée le 20 janvier 2023.