



ANNEE ACADEMIQUE 2023-2024

CoucouAI : LA SURVEILLANCE INTELLIGENTE POUR SOUTENIR LES ELEVEURS DANS LA PRESERVATION DES ESPECES MENACEES

TRAVAIL DE FIN D'ETUDES

**présenté en juin 2024 par
HANINI Samia**

**en vue de l'obtention du grade
de Bachelier en INFORMATIQUE ORIENTATION
DEVELOPPEMENT D'APPLICATIONS**

Professeur-conseil: Madame, DE LUCA SILVANA



ANNEE ACADEMIQUE 2023-2024

CoucouAI : LA SURVEILLANCE INTELLIGENTE POUR SOUTENIR LES ELEVEURS DANS LA PRESERVATION DES ESPECES MENACEES

TRAVAIL DE FIN D'ETUDES

**présenté en juin 2024 par
HANINI Samia**

**en vue de l'obtention du grade
de Bachelier en INFORMATIQUE ORIENTATION
DEVELOPPEMENT D'APPLICATIONS**

Professeur-conseil: Madame, DE LUCA SILVANA

Remerciements

Je tiens avant tout à exprimer ma gratitude envers ma Professeure conseil, Madame Silvana De Luca, Professeure d'informatique à la HEPH Condorcet, pour ses précieux conseils et sa disponibilité qui ont su me guider à travers la rédaction de ce travail.

J'aimerais saisir cette occasion pour exprimer ma profonde reconnaissance envers mon maître de stage et développeur au sein du MIC, Monsieur Lloyd Colart, pour avoir pris le temps de partager son savoir-faire, de répondre à mes questions et pour son soutien dans le processus de rédaction de ce mémoire.

Je souhaite également partager mon immense gratitude envers toute l'équipe du MIC, Monsieur Antoine Smet, Monsieur Renaud Dumont, Madame Martine Meunier, Madame Laura Lieu, Monsieur Frédéric Carbonnelle et Madame Aurélie Couvreur pour leur accueil ainsi que pour le temps et l'énergie qu'ils ont consacré au bon déroulement de ce stage.

Je souhaite adresser mes remerciements au corps professoral pour l'enseignement qu'ils ont dispensé au cours de ces trois dernières années. Leur expertise et leur dévouement m'ont permis d'acquérir les outils nécessaires au bon déroulement de mon cursus.

Mes remerciements vont aussi vers mes parents, Madame et Monsieur Hanini, ainsi que mes frères, Yacine et Samir, pour leurs conseils, leur soutien sans faille et pour leur aide à la relecture de ce mémoire.

Un grand merci également à mes amies, Soukaina Mougas et Estelle Marlière, pour leurs encouragements et leur aide pour la relecture de ce travail.

Enfin, je souhaite adresser mes remerciements à toutes les personnes qui, de loin ou de près, ont contribué au bon déroulement de ce travail de fin d'étude.

Table des matières

<i>I.</i>	<i>Introduction</i>	5
<i>II.</i>	<i>Présentation de l'entreprise</i>	6
<i>III.</i>	<i>Le projet</i>	12
A.	Le porteur de projet	12
B.	Définition du projet CoucouAI	13
C.	L'équipe	14
D.	Analyse préliminaire	15
<i>IV.</i>	<i>Outils, langages et méthodologie utilisés</i>	17
A.	Outils de communication	17
B.	Environnements de programmation	18
C.	Versioning avec Git ¹⁰	19
D.	Langages de programmation	20
E.	Frameworks, bibliothèques et autres technologies	21
F.	Les services Microsoft Azure	23
1.	Introduction au Cloud Computing ²³	23
2.	Les Fonctions Azure ²⁴	24
3.	Les comptes de stockage ²⁵	25
		26
4.	Azure AI Custom Vision ²⁸	28
G.	Méthodologie Agile Scrum ²⁹	33
<i>V.</i>	<i>Description détaillée des différentes phases de développement</i>	34
<i>VI.</i>	<i>Implémentation</i>	48
A.	Infrastructure de stockage Azure	48
B.	Structure de l'API C#	50
C.	Structure de l'API Python	51
D.	Analyse du code	52
<i>VII.</i>	<i>Phase de test</i>	59
<i>VIII.</i>	<i>Améliorations potentielles</i>	59
<i>IX.</i>	<i>Activités et formations offertes par le MIC</i>	59
<i>X.</i>	<i>Compétences acquises</i>	59
a.	Compétences comportementales	59
b.	Compétences techniques	60
<i>XI.</i>	<i>Conclusion</i>	61
<i>XII.</i>	<i>Bibliographie</i>	62

Figure 1 : Arborescence de l'équipe du MIC	11
Figure 2 : logo MIC	11
Figure 3 : Monsieur Kurt EVERAERTS, directeur de K-Force et de ClearMedia South	12
Figure 4: Illustration du monde	13
Figure 5: Bradley FOURNEAU Développeur	14
Figure 6: Lloyd COLART Développeur et Product Owner	14
Figure 7 : Samia HANINI Développeuse	14
Figure 8: Diagramme de cas d'utilisation	16
Figure 9: Table périodique des services Azure	24
Figure 10: Schéma des différents types de services de stockage Azure.....	25
Figure 11: Tables de stockage Azure utilisées dans le projet	26
Figure 12: Interface regroupant les modèles Azure Custom Vision	28
Figure 13: Interface permettant la création d'un modèle de détection d'objets Azure Custom Vision.....	29
Figure 14: Aperçu du dataset dans l'interface Azure Custom Vision.....	30
Figure 15: Aperçu de l'étiquetage d'un élément sur image via Azure Custom Vision	30
Figure 16: Aperçu des types d'entraînement via Azure Custom Vision	31
Figure 17: Aperçu des données de performance après l'entraînement d'un modèle via Azure Custom Vision	31
Figure 18: Aperçu des différents formats d'exportation sur Azure Custom Vision	32
Figure 19: Schéma illustrant la méthode Agile Scrum	33
Figure 20: Aperçu de la page d'accueil du prototype CoucouAI	35
Figure 21: Aperçu du résultat d'une prédiction pour le prototype CoucouAI	36
Figure 22: Aperçu du téléchargement d'une image dans le prototype CoucouAI.....	36
Figure 23: Aperçu du résultat d'une prédiction pour le prototype CoucouAI	36
Figure 24: Aperçu du résultat obtenu du test YOLO	37
Figure 25: Aperçu du résultat obtenu pour la phase 2 du développement du projet CoucouAI.....	39
Figure 26: Aperçu de la sélection multiple de modèles	42
Figure 27: Aperçu de l'e-mail reçu par les employés disponibles	43
Figure 28: Aperçu de la page permettant d'accepter une alerte	44
Figure 29: Aperçu du résultat après avoir pris en charge une alerte	44
Figure 30: Aperçu du résultat si l'alerte a déjà été prise en charge	44
Figure 31: Aperçu des employés et de leurs informations	45
Figure 32: Aperçu de l'interface de modification des données d'un employé.....	45
Figure 33: Aperçu du logiciel Nx Witness.....	47
Figure 34: Colonnes de la table Azure "Alerts"	48
Figure 35: Colonnes de la table Azure "Employees"	49
Figure 36: Colonnes de la table Azure "GroupModels"	49
Figure 37: Conteneurs Blob Azure.....	50
Figure 38: Schéma de la structure de l'API C#	50
Figure 39: Structure des fichiers de l'API C#	50
Figure 40: Structure des fichiers de l'API Python	51
Figure 41: Code de la méthode GetAllCamera() de l'API C#	52
Figure 42: Extrait de code de la classe "UploadImageHostedService" de l'API C#.....	53
Figure 43: Extrait de code de la méthode UploadImage() de l'API C#.....	54

Figure 44: Structure des dossiers contenant les modèles et les labels dans l'API Python	55
Figure 45: Extrait de code de l'API Python pour la superposition des modèles	56
Figure 46: Extrait de code de l'API Python pour la redondance des alertes	57
Figure 47: Code de la méthode GetAllAlert() de l'API C#.....	58

I. Introduction

Dans le monde, plus de 40 000 espèces animales sont menacées. Certaines institutions ont alors vu le jour afin de mettre en place un système d'accueil pour préserver ces dernières. Cependant, ces institutions utilisent bien souvent des systèmes de vidéosurveillance en direct afin de monitorer les animaux qu'elles abritent. Seulement, ce système de sécurité demande une attention constante de la part d'opérateurs humains, sans quoi, le système en perd tout son sens. Entre les moyens humains colossaux mis en place et le risque d'erreurs humaines, cette approche n'est pas optimale et engendre des risques, tant sur le plan animal que financier.

Monsieur Kurt EVERAERTS, notre client, s'est rendu au MIC avec l'idée de développer une solution qui aidera les éleveurs à maintenir la sécurité des espèces qu'ils hébergent, tout en réduisant la dépendance au personnel pour la surveillance. Pour ce faire, il souhaitait intégrer des outils d'intelligence artificielle dans les systèmes de surveillance existants.

Dans une ère où la planète commence à exprimer son mécontentement après des années de négligence de son état, de nombreuses espèces animales en ont payé le prix. Il a été crucial pour moi de m'engager dans cette lutte à travers ce projet, qui a un impact aussi bien sur la progression de l'innovation que sur l'aide à la sauvegarde des espèces menacées. Dans notre cas d'étude, les nids de coucous sont au centre de l'attention. Chaque œuf de coucou est précieux et doit être préservé à tout prix.

Dans ce mémoire, nous nous interrogerons sur l'utilisation de l'intelligence artificielle dans le but de détecter des objets via un flux de caméra de vidéosurveillance en direct et ainsi alerter les éleveurs dès qu'un élément perturbateur est détecté.

Afin de répondre à ces interrogations, nous nous sommes engagés dans le développement de solutions fondées sur la recherche approfondie des moyens les plus adaptés pour répondre aux besoins du client. Nous avons suivi la méthodologie Agile Scrum, des réunions bimensuelles ont été entreprises avec le client afin que chaque phase de développement du projet se rapproche le plus de ses attentes. Ensuite, nous nous sommes aidés de professionnels aguerris et de ressources en ligne afin de trouver des solutions à nos problèmes.

L'objectif est de déterminer si oui ou non, l'intelligence artificiel peut jouer un rôle dans la détection d'éléments dans un flux de caméra en direct et ainsi agir dans la protection des œufs de coucous. De plus, nous cherchons à savoir si l'intelligence artificielle peut être utilisée de manière simple et intuitive par des développeurs junior.

II. Présentation de l'entreprise



Le MIC, c'est quoi ?

Le MIC, de son nom complet, Meet Innovate Create, est une association sans but lucratif créée en 2009, elle est supportée par le secteur privé ainsi que par le secteur public. Son site se situe 64, avenue des Bassins, 1er étage, 7000 Mons, Belgique.

Son objectif est de soutenir le développement de l'économie wallonne à travers la transformation numérique.

C'est une équipe, une communauté et un réseau prêt à aider les professionnels afin de progresser dans leurs idées, projets et compétences. Le MIC est également un lieu où trouver l'inspiration et découvrir des projets innovants qui construiront le monde de demain.

Leurs partenariats

Le MIC est soutenu par des leaders de l'industrie numérique tels que Microsoft, Proximus et la région Wallonne qui s'impliquent fortement dans la gouvernance de l'entreprise.

Leurs collaborations

Le MIC entretient une relation privilégiée avec les autres acteurs numériques de la région wallonne, tels que l'Infopole et DW4AI (DigitalWallonia4.ai). Ensemble, ils s'investissent dans développement de l'excellence numérique wallonne et dans la mise en place de complémentarités pour accélérer l'adoption des technologies numériques par les entreprises.

De plus, le MIC collabore avec la LME (La Maison de l'Entreprise) pour effectuer des diagnostics auprès des PME. Ils organisent, à travers le programme Industrie du Futur et l'EDIH (European Digital Innovation Hub) Walhub, des événements sur le thème de l'industrie 4.0. Enfin, le MIC, prend part à des programmes de formations tels que Technofutur, Technocité, Futurocité, etc.

L'entreprise possède aussi de nombreux partenaires, parmi ceux-ci, nous pouvons retrouver Dell pour la Softlab Academy ou encore Logitech pour l'Experience Center.

Les missions du MIC

Le chemin de la digitalisation

1. Sensibilisation

⇒ Approfondir la compréhension des enjeux et opportunités du numérique à travers des méthodes et des outils pédagogiques.

2. Démonstration

⇒ Découverte de projets développés pour des PME wallonnes.

3. Expérimentation

⇒ Prototyper des idées et les tester à moindre coût, réduisant ainsi les risques associés à l'implémentation de ces solutions innovantes.

4. Promotion et diffusion

⇒ Mise en lumière des produits ou services à travers des stratégies de marketing digital efficaces et accessibles.

Les missions du MIC envers les chefs d'entreprise

Le MIC s'engage à soutenir les PME et startup wallonnes, quelle que soit leur taille, dans l'adoption de nouvelles technologies numériques et ce, grâce à 2 piliers fondamentaux : l'accompagnement des entreprises dans leur processus de digitalisation et la définition claire de la valeur ajoutée de leurs services.

Les missions du MIC envers les développeurs

1. Partage d'expérience

⇒ Le partage d'expérience par d'autres professionnels du secteur permet d'évoluer dans différentes technologies à différents degrés d'approfondissement. L'informatique étant une science en constante évolution, chacun apporte sa pierre à l'édifice et bénéficie de l'expérience de ses pairs.

2. Développement des compétences

⇒ Il est, pour le MIC, primordial d'accompagner les développeurs dans l'acquisition de compétences sur des technologies émergentes pour les aider à diversifier leur offre de services et faciliter la production de nouveaux outils.

3. Réseautage

⇒ Le MIC a pour but d'aider la communauté des développeurs wallons à interagir et partager à travers différents canaux et événements rassembleurs tels que des hackathons, des conférences ou des journées de réseautage.

Les activités du MIC

Activités pour les entreprises

Softlab Academy

La Softlab Academy permet de mettre en œuvre les idées de PME wallonnes en seulement 3 mois par une équipe de jeunes développeurs.

Quick Start

Quick Start vise à concrétiser l'aspect technologique de l'idée d'une entreprise et de la confronter au marché tout en identifiant ses éventuels pièges et atouts.

Parcours d'adoption technologique

Ce parcours se base sur 3 étapes que sont le « VOIR-TESTER-CRÉER », permettant aux PME wallonnes d'accélérer leur transformation digitale à travers des exemples concrets et innovants.s

Tech The Lead

C'est un programme d'accompagnement d'un an pour les entreprises IT désireuses d'optimiser les outils cloud et l'environnement de travail chez leurs clients.

- Accompagnement business
- Espace technologique
- Espace de démonstration

Activités pour les développeurs

Power Branch

Mise à niveau de compétences techniques à travers différents formats comme des diffusions en direct ou encore des séminaires.

Hackathons (Hack in the Woods, Space Office, Kiss Your Teacher...)

Réalisation de prototypes en un week-end par des équipes de développeurs.

Open Space

Réseautage mensuel dans les bureaux du MIC autour d'un café.

Certifications

Valorisation des compétences en collaboration avec Pearson's View.

Focus sur la Softlab Academy



Publics visés

- ⇒ Les PME wallonnes
- ⇒ Les étudiants en dernière année de bachelier en informatique

Programme

Établi il y a 8 ans, le programme consiste au développement de prototypes informatiques par des étudiants sélectionnés avec soin en fonction de leur profil et de leurs compétences. Seuls 10 projets innovants sont retenus chaque année.

Objectifs

Pour les étudiants

La Softlab Academy donne l'opportunité aux étudiants de vivre un stage unique en son genre, encadré par des experts et leur permettant d'approcher la réalité du métier de développeur, tout en les intégrant dans la communauté des développeurs wallons.

Pour les entreprises

Les entreprises pourront faire développer un prototype IT innovant dans un délai et un budget définis, en misant sur les aspects risqués de leur idée.

L'équipe

L'équipe est composée de 6 personnes passionnées par la technologie.

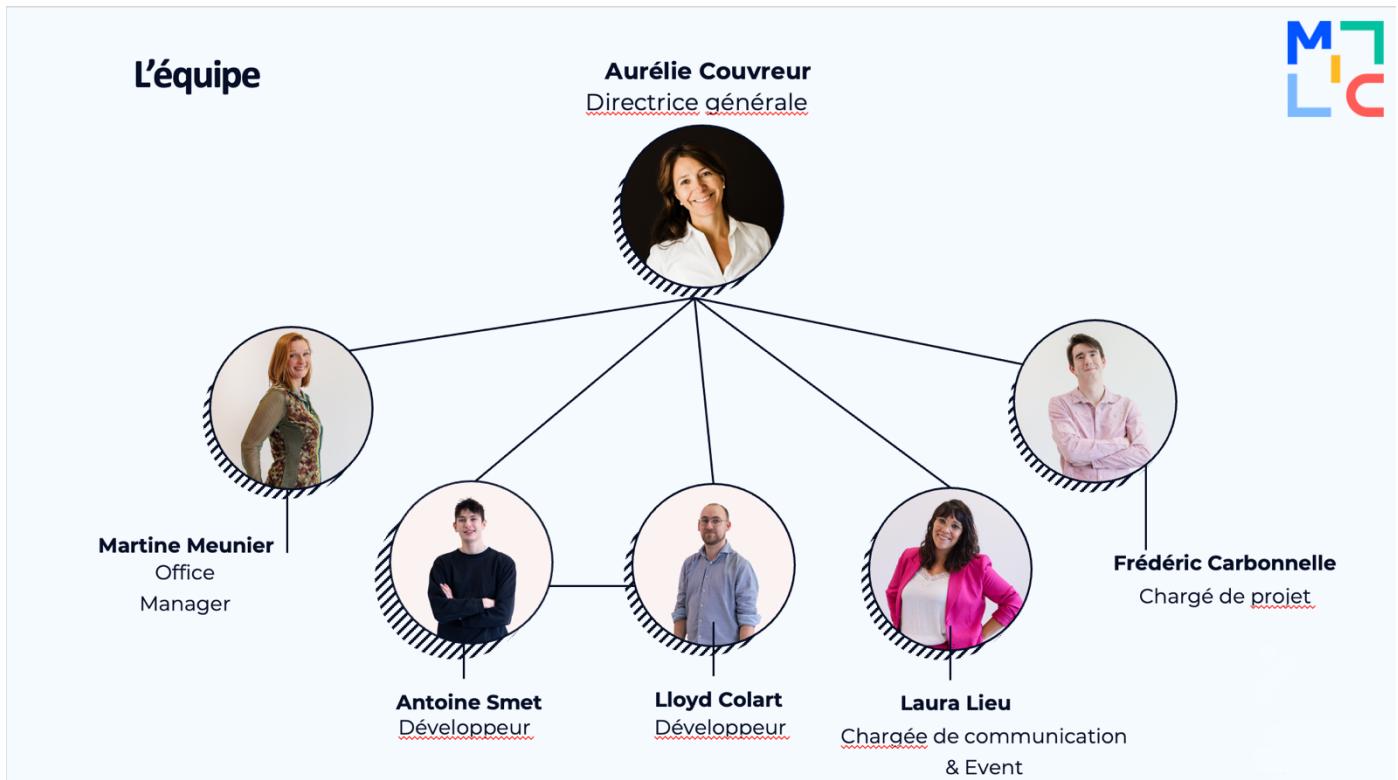


Figure 1 : Arborescence de l'équipe du MIC



Figure 2 : logo MIC

III. Le projet

A. Le porteur de projet

Monsieur Kurt EVERAERTS est actuellement le directeur général de K-Force¹ et de ClearMedia² South.

Conscient de l'importance cruciale de la transformation numérique et de l'impact l'innovation numérique, Monsieur Kurt EVERAERTS fonde en 2000 K-Force. Cette société propose des services dans le domaine des solutions informatiques, cloud et téléphonie pour les petites et moyennes entreprises.

En janvier 2024, Monsieur Kurt EVERAERTS a pris la direction de ClearMedia South. ClearMedia est une filiale de Proximus, proposant des solutions Cloud aux petites et moyennes entreprises.

Monsieur Kurt EVERAERTS a fait appel au programme de la Softlab Academy proposé par le MIC suite à une sollicitation par un zoo accueillant des espèces menacées. Le zoo souhaite obtenir une solution qui simplifiera la gestion des éléments potentiellement dangereux au sein de cages d'animaux, afin d'améliorer l'efficacité des interventions et diminuer les coûts engendrés par les pertes animales.

Monsieur EVERAERTS a sollicité le MIC car l'asbl est reconnue pour son utilisation de technologies innovantes. Dans le cadre de ce projet, l'intelligence artificielle est un élément clé qui permet le développement d'une solution efficace.



Figure 3 : Monsieur Kurt EVERAERTS, directeur de K-Force et de ClearMedia South

¹ K-Force Computer Solutions. (2024, 17 janvier). *Le partenaire pour votre ICT, digital workplace et téléphonie | K-Force*. K-Force. Consulté le 8 avril 2024, à l'adresse <https://k-force.be/fr/>

² ClearMedia. (2024, 7 mars). *ClearMedia - Découvrez les super pouvoirs de ClearMedia !* Consulté le 8 avril 2024, à l'adresse <https://www.clearmedia.be/fr/>

B. Définition du projet CoucouAI

Problématique

Certaines espèces en voie de disparition sont accueillies dans des établissements où elles font l'objet d'une surveillance constante. Cette surveillance accrue demande des ressources humaines importantes et expose à des risques d'inefficacité en raison de potentielles erreurs humaines. CoucouAI vise à combler ces lacunes en utilisant l'intelligence artificielle pour détecter les éléments perturbateurs dans l'environnement des animaux. Ainsi, les éleveurs pourront être alertés à tout moment et intervenir dans les plus brefs délais, sans avoir à consacrer l'entièreté de leur temps à assurer la garde via les caméras de vidéosurveillance.

Description du projet

CoucouAI est un projet axé sur le développement d'un système d'alerte se basant sur les prédictions de modèles intelligents de détection d'objets destinés à détecter en direct, via une caméra de surveillance, la présence d'éléments perturbateurs au sein de nids de coucous. Ces modèles devront être capable de repérer divers éléments tels que des œufs, des mains humaines et des souris. Cette technologie permettra de contacter un groupe d'employés choisis via leur boîte mail, ils pourront alors indiquer s'ils souhaitent intervenir ou non. Cette approche vise à augmenter les chances de survie des coucous gris grâce à la prise en charge rapide de situations potentiellement dangereuses pour leur vie.



Figure 4: Illustration du monde

C. L'équipe



Figure 6: Lloyd COLART
Développeur et Product Owner



Figure 5: Bradley FOURNEAU
Développeur



Figure 7 : Samia HANINI
Développeuse

Un total de 3 personnes ont travaillé sur du développement du projet.

1. Le Product Owner³, Lloyd COLART, est le responsable du projet, il est expert de la méthodologie Agile Scrum. Son rôle principal est de définir et de prioriser les différentes fonctionnalités du produit et d'assurer la communication entre l'équipe de développement et le client. Lloyd COLART est également développeur.
2. Les développeurs⁴, Bradley FOURNEAU et moi-même, sommes les techniciens chargés de la conception des programmes. Dans le cadre de notre travail, nous avons alterné entre backend et frontend afin d'acquérir un large éventail de compétences techniques et de mieux comprendre le fonctionnement global du projet que nous avons développé.

Le projet est donc développé par binôme. Mon binôme se nomme Bradley FOURNEAU, il est élève en dernière année de bachelier en informatique à la HELHa Mons, il est également stagiaire au MIC.

³ Fiche métier : Product Owner - Cegos. (s. d.). Consulté le 8 avril 2024, à l'adresse <https://www.cegos.fr/ressources/mag/fiches-metiers/les-metiers-de-la-gestion-de-projets/fiche-metier-product-owner>

⁴ Qu'est-ce qu'un développeur et quelles sont ses missions ? (s. d.). DeviensDev. Consulté le 8 avril 2024, à l'adresse <https://deviens.dev/metier-developpeur/cest-quoi-un-developpeur/>

D. Analyse préliminaire

Analyse de l'existant

Des caméras IP sont utilisées afin de surveiller les nids de coucous, la retransmission de ses images en direct est visible sur une application pour smartphone.

Objectifs fonctionnels

- Créer et entraîner des modèles de détection d'objets pour détecter les œufs, les souris et les mains humaines.
- Intégrer les modèles de détection d'objets avec les caméras de surveillance pour une détection en temps réel.
- Créer une application web comprenant un tableau de bord qui affiche en temps réel les alertes ainsi que les informations s'y associant.
- Mettre en place un système de gestion des employés via l'application web.
- Mettre en place un système permettant la sélection d'un ou de plusieurs modèles de détection d'objets pour un groupe de caméras.
- Mettre en place un système qui contactera les employés disponibles en cas d'alerte.
- Gérer la réponse des personnes recevant les alertes.

Objectifs non-fonctionnels

- S'assurer que le système d'alerte est réactif.
- Garantir la confidentialité des données des alertes et des employés.
- L'application web doit être facile d'utilisation.
- Réduire au maximum le coût de l'utilisation des différents services.

Les cas d'utilisation⁵

Un cas d'utilisation est un outil de modélisation qui permet de décrire les interactions entre un système et ses acteurs. Cette technique permet de capturer les exigences fonctionnelles en se concentrant sur les actions que l'utilisateur peut effectuer et les réponses qu'il va recevoir du système suite à ces actions.

Diagrammes de cas d'utilisation

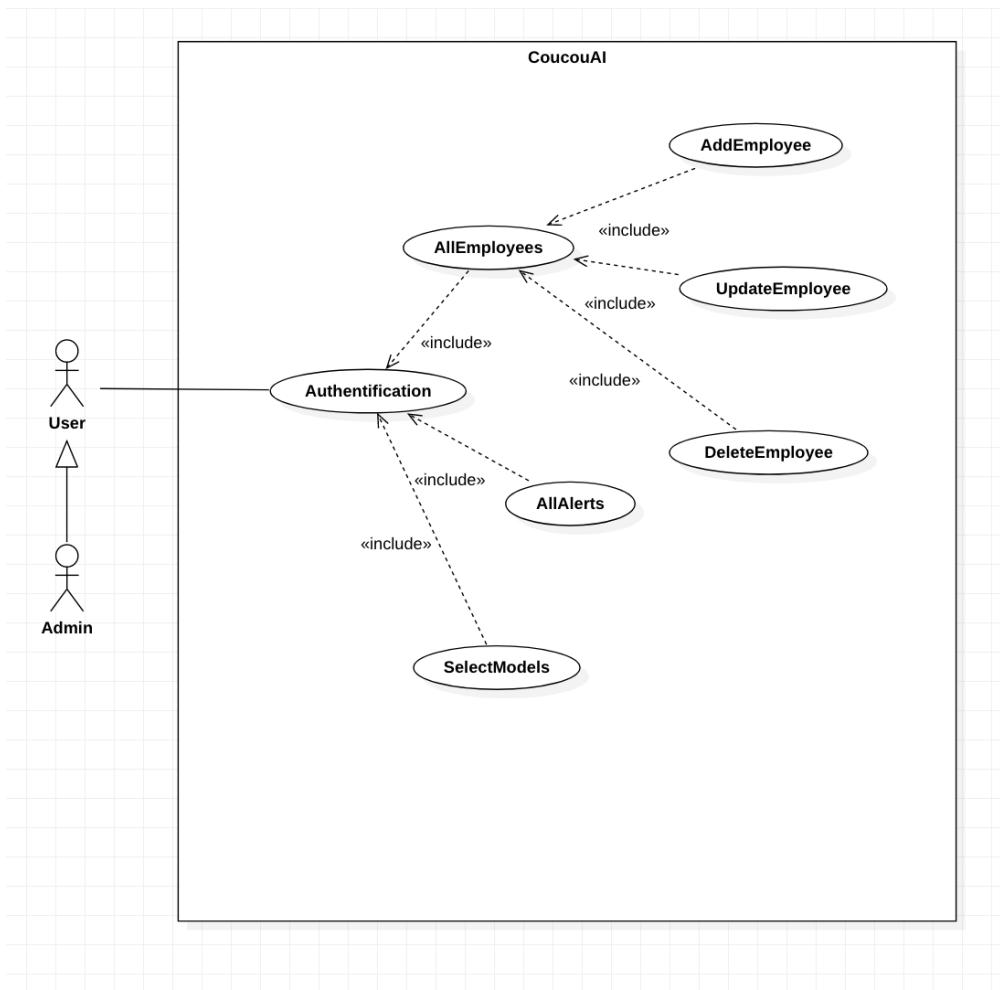


Figure 8: Diagramme de cas d'utilisation

⁵

IBM documentation. (s. d.). Consulté le 8 avril 2024, à l'adresse <https://www.ibm.com/docs/fr/engineering-lifecycle-management-suite/lifecycle-management/7.0.1?topic=requirements-defining-use-cases>

IV. Outils, langages et méthodologie utilisés

Dans cette section, nous aborderons les différents outils qui ont été utilisés dans le cadre du processus de développement du projet.

A. Outils de communication

Afin d'assurer une communication optimale entre les membres de l'équipe et les membres externes, j'ai eu l'occasion d'utiliser différents canaux de communication.



1. Slack⁶ est une application de communication pour les entreprises créée en 2013. Cette application a permis à mon équipe de partager des informations de manière instantanée et surtout, de pouvoir communiquer de manière optimale pendant les jours de télétravail.
2. Microsoft Teams⁷ est une plateforme de communication professionnelle lancée par Microsoft en 2017. Elle facilite la collaboration au sein d'équipes. Dans le cadre de mon stage, Teams m'a permis de communiquer avec mon binôme en visioconférence durant les jours de télétravail via des réunions Teams.

⁶ Slack. (s. d.). *Qu'est-ce que Slack ?* Slack Help Center. Consulté le 8 avril 2024, à l'adresse <https://slack.com/intl/fr-fr/help/articles/115004071768-Qu%E2%80%99est-ce-que-Slack->

⁷ *Qu'est-ce que Microsoft Teams ?* (s. d.). Consulté le 8 avril 2024, à l'adresse <https://support.microsoft.com/fr-fr/office/vid%C3%A9o-qu-est-ce-que-microsoft-teams-422bf3aa-9ae8-46f1-83a2-e65720e1a34d>

B. Environnements de programmation

Dans cette partie, vous retrouverez les différents environnements de programmation qui ont été utilisés, ceux-ci ont varié en fonction des technologies employées afin de garantir une intégration fluide des solutions développées.



1. Visual Studio Code⁸ est un éditeur de code gratuit développé par Microsoft en 2015. Il est utilisable pour toutes les technologies impliquées dans la conception du projet.



2. Visual Studio⁹ est un environnement de développement puissant créé par Microsoft. Il peut être utilisé pour développer toutes sortes d'applications comme des sites web, des applications web, des applications mobile, des bases de données et des services web. Par souci de facilité, nous avons utilisé Visual Studio pour exécuter les API Python et C# de notre projet.

⁸ Visual Studio : IDE et éditeur de code pour les développeurs de logiciels et les équipes. (2024, 15 mars). Visual Studio. Consulté le 8 avril 2024, à l'adresse <https://visualstudio.microsoft.com/fr/>

⁹ Anandmeg. (2023, 27 octobre). Qu'est-ce que l'IDE Visual Studio ? Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/fr-fr/visualstudio/get-started/visual-studio-ide?view=vs-2022>

C. Versioning avec Git¹⁰



Git est un outil de contrôle de version gratuit et open-source créé en 2005 par Linus Torvalds. À l'heure actuelle, Git est l'outil de versioning le plus populaire chez les développeurs. Le contrôle de version est le fait d'enregistrer différents fichiers au cours des différentes étapes de développement d'un projet afin de garder un historique et de pouvoir naviguer entre plusieurs versions du travail. La majorité des équipes de développement utilisent cet outil afin de développer des projets d'envergures diverses.

Git est décentralisé, c'est-à-dire que chaque copie d'un projet est un dépôt contenant l'historique de toutes les modifications ayant été effectuées sur ce travail, ce qui offre une grande flexibilité dans les workflows de développement. Git permet d'effectuer différentes actions comme ajouter des modifications, créer des branches, comparer les versions ou encore fusionner des changements, tout cela de manière optimisée. Git est réputé pour ses performances élevées grâce aux algorithmes utilisés pour gérer l'historique et les modifications de manière optimale.

De plus, la priorité absolue lors du développement de Git a été l'intégrité du code source. Tous les éléments du dépôt ont été sécurisés par des algorithmes de hachage cryptographiques, notamment l'algorithme « SHA1 ». Il veille à protéger le code ainsi que l'historique contre toute tentative de modification malveillante ou accidentelle.

Enfin, Git est flexible, en effet, il est capable de prendre en charge divers workflows de développement, d'élaborer différents projets de différentes tailles et est compatible avec de nombreux protocoles et systèmes existants.

Son utilisation dans le projet CoucouAI

Pour chaque nouvelle fonctionnalité, une branche a été créée depuis la branche principale « develop » avec la dénomination suivante : « feature/{fonctionnalité} ». Par exemple, pour la création du système permettant l'envoi de mails contenant l'alerte aux éleveurs, nous avons créé une branche nommée « feature/emailer » sur notre dépôt Git. Une fois la fonctionnalité prête, nous procédions alors à la création d'une « pull request », ce qui signifie que la branche est intégrée à la branche principale « develop » après vérification et validation. Il était interdit de fusionner une branche non fonctionnelle ou de travailler directement sur la branche « develop ».

10

Atlassian. (s. d.-b). *Qu'est-ce que Git ? | Atlassian Git Tutorial*. Consulté le 2 mai 2024, à l'adresse <https://www.atlassian.com/fr/git/tutorials/what-is-git>

D. Langages de programmation

Ci-dessous, vous retrouvez les langages de programmation définis qui ont été utilisés dans le développement de nos solutions.



⇒ TypeScript¹¹ est langage de programmation open-source développé par Microsoft, c'est un sur-ensemble syntaxique de JavaScript qui ajoute un typage statique optionnel et des fonctionnalités avancées à JavaScript. Il a été utilisé pour développer l'application web de notre projet.



⇒ C#¹² est un langage de programmation moderne, orienté objet et de type sécurisé créé par Microsoft. C# permet aux développeurs de créer de nombreux types d'applications sécurisées et robustes qui s'exécutent dans .NET. Nous avons choisi C# afin de développer notre API principale.



⇒ Python¹³ est langage de programmation interprété de haut niveau, gratuit et multiplateforme. C'est un langage orienté objet qui est relativement simple à prendre en main, il est couramment utilisé en analyse de données ou en intelligence artificielle. Pour CoucouAI, Python intervient dans l'utilisation des modèles intelligents de détections d'objets créés et entraînés sur la plateforme Azure AI Custom Vision.

¹¹ Kinsta. (2023, 24 mai). *Qu'est-ce que TypeScript ? Un guide complet - Kinsta®*. Kinsta®. Consulté le 8 avril 2024, à l'adresse <https://kinsta.com/fr/base-de-connaissances/guide-complet-typescript/>

¹² C sharp - Définition et Explications. (s. d.). Techno-Science.net. Consulté le 8 avril 2024, à l'adresse <https://www.techno-science.net/glossaire-definition/C-sharp.html>

¹³ RÉ ; Daction, L. (2020, 31 mars). Python ; : dé ; finition et utilisation de ce langage informatique. Consulté le 8 avril 2024, à l'adresse <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>

E. Frameworks, bibliothèques et autres technologies

Voici les frameworks, bibliothèques et autres technologies utilisés pour développer CoucouAI :



- ⇒ .NET¹⁴ est une plateforme de développement gratuite permettant de créer des applications web, mobiles et bureaux pouvant s'exécuter nativement sur n'importe quel système d'exploitation. C'est avec .NET que nous avons développé notre API principale en C#.



- ⇒ Next.js¹⁵ est un framework React permettant la création de sites web statiques et d'applications web performantes, optimisés pour le référencement. Il a été utilisé afin de développer notre application web.



- ⇒ Node.js¹⁶ est une plateforme de développement possédant des bibliothèques permettant la réalisation de différentes actions comme ouvrir et fermer des connections réseaux, créer un fichier ou écrire sur la sortie standard. Il est important de souligner que Node.js n'est ni un framework, ni un serveur mais une plateforme logicielle gratuite qui permet d'exécuter le code JavaScript du côté serveur.



- ⇒ Flask¹⁷ est un micro framework de développement web gratuit écrit en Python. Ce framework nous a permis de développer notre API Python.

¹⁴ Qu'est-ce que .Net ? – .Net expliqué – AWS. (s. d.). Amazon Web Services, Inc. Consulté le 8 avril 2024, à l'adresse <https://aws.amazon.com/fr/what-is/net/>

¹⁵ Kinsta. (2022, 19 décembre). Qu'est-ce que Next.js ? Un regard sur le framework JavaScript populaire. Kinsta®. Consulté le 8 avril 2024, à l'adresse <https://kinsta.com/fr/base-de-connaissances/next-js/>

¹⁶ Introduction à Node.js | Makina Corpus. (s. d.). Makina Corpus. Consulté le 8 avril 2024, à l'adresse <https://makina-corpus.com/front-end/introduction-nodejs>

¹⁷ Dyouri, A. (s. d.). Créer une application Web avec Flask Python. Developpez.com. Consulté le 8 avril 2024, à l'adresse <https://python.developpez.com/tutoriel/intro-flask-python3/>



⇒ TensorFlow Lite¹⁸ est ensemble d'outils permettant le déploiement de modèles de machine learning sur des appareils mobiles, des systèmes embarqués et d'autres plateformes possédant des ressources limitées. C'est une version légère et optimisée de TensorFlow. C'est au format TensorFlow Lite que nos modèles ont été exportés et utilisés.



⇒ OBS¹⁹ abréviation de Open Broadcaster Software est un programme gratuit permettant l'enregistrement d'écran et le streaming. Il nous a permis de simuler une webcam virtuelle capturant les vidéos de cage de coucous fournies par le client.



⇒ Nx Witness²⁰ est un logiciel de vidéosurveillance multiplateforme développé par Network Optix créé pour découvrir, visualiser, enregistrer et gérer des caméras vidéo IP.



⇒ Nx Cloud²¹ est une plateforme de collaboration et d'optimisation pour les projets développés avec Nx.



⇒ MailKit²² est une bibliothèque .NET qui permet l'envoi et la réception d'e-mails. Nous l'avons sélectionnée pour envoyer des alertes par e-mails personnalisés à tous les éleveurs disponibles.

18 TensorFlow lite. (s. d.). TensorFlow. Consulté le 8 avril 2024, à l'adresse <https://www.tensorflow.org/lite/guide?hl=fr>

19 Open Broadcaster Software | OBS. (s. d.). Consulté le 12 mai 2024, à l'adresse <https://obsproject.com/fr>

20 Security, L. F. +. (2022, août 15). Nx Witness v5.0 est arrivé. *Stentor 6.1 Demo*. Consulté le 10 mai 2024, à l'adresse <https://www.lobeco.be/fr/pageid/newsitem-2022-07-28>

21 Why NX Cloud ? (s. d.). Nx. Consulté le 10 mai 2024, à l'adresse <https://nx.dev/ci/intro/why-nx-cloud>

22 15 Intégration de MailKit avec le service de transmission de messages. (s. d.). (C) Copyright 2024. Consulté le 10 mai 2024, à l'adresse <https://docs.oracle.com/fr-ca/iaas>

F. Les services Microsoft Azure



Dans cette section, nous allons expliquer l'usage et le fonctionnement des services Microsoft Azure utilisés pour le projet CoucouAI.

Parmi ceux-ci vous pouvez retrouver :

- ⇒ Les fonctions Azure
- ⇒ Les comptes de stockage Azure
- ⇒ Azure AI Custom Vision

1. Introduction au Cloud Computing²³

Étant donné l'utilisation de nombreux services Azure dans le développement du projet, il a été primordiale de comprendre ce qu'est le Cloud Computing, c'est pourquoi nous allons l'expliquer brièvement dans cette section.

Le Cloud Computing est le fait d'utiliser des services hébergés sur Internet plutôt que sur des serveurs locaux. Microsoft Azure propose différents services tels que le stockage de données, l'hébergement d'applications web, la gestion des identités et des accès, et bien d'autres.

Le Cloud Computing présente de nombreux avantages notamment la scalabilité, elle permet aux utilisateurs de s'adapter rapidement en cas de changement des besoins sans devoir investir dans des ressources matérielles onéreuses ou supporter des temps d'arrêts prolongés. Par exemple, pour CoucouAI, nous pouvons augmenter ou diminuer aisément la capacité de stockage en fonction de la demande de Monsieur Kurt EVERAERTS, notre client.

Un aspect crucial dans développement de CoucouAI réside dans la capacité de pouvoir monitorer et gérer les coûts engendrés par l'entraînement des modèles d'intelligence artificielle. Le système de facturation à l'utilisation est un autre point fort des services Azure. Ce système permet aux utilisateurs de payer uniquement ce qu'ils consomment, ce qui offre une meilleure gestion des coûts. En fonction des besoins de l'utilisateur, ce modèle peut entraîner des économies significatives à long terme. Grâce à ce système, nous avons pu limiter les dépenses liées à l'utilisation des services Azure lorsque cela était nécessaire.

²³

What is cloud computing ? | Microsoft Azure. (s. d.). Consulté le 9 mai 2024, à l'adresse <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing>

Voici un aperçu de la table périodique illustrant la variété des services offerts par Azure :

Figure 9: Table périodique des services Azure

2. Les Fonctions Azure²⁴

Dans notre projet

Dans le cadre de notre projet, les fonctions Azure sont utilisées dans la phase 1.B du processus de développement. La phase 1.B consiste à l'utilisation des fonctions Azure afin d'effectuer les prédictions sur les images téléchargées, à stocker ces prédictions et à sécuriser l'accès à nos conteneurs Blob Azure.

De quoi s'agit-il ?

Les fonctions Azure sont des solutions hébergées sur la plateforme Azure pouvant être exécutées par divers événements tels que des requêtes HTTP, des modifications dans une base de données, etc. L'avantage principal des fonctions Azure est leur nature serverless, c'est-à-dire que nous pouvons déployer et gérer du code sans devoir se soucier de la gestion de l'infrastructure sous-jacente.

3. Les comptes de stockage²⁵

Dans notre projet

Pour CoucouAI, nous avons stocké toutes nos données sur différents services de stockage Azure.

De quoi s'agit-il ?

Un compte de stockage Azure contient tous les objets de stockage Azure : blobs, fichiers, files d'attente et tables. Les comptes de stockage fournissent un espace de noms unique pour les données de stockage Azure, accessible depuis n'importe où dans le monde via HTTP ou HTTPS. Les données d'un compte de stockage sont durables et hautement disponibles, sécurisées et offrent une évolutivité considérable.

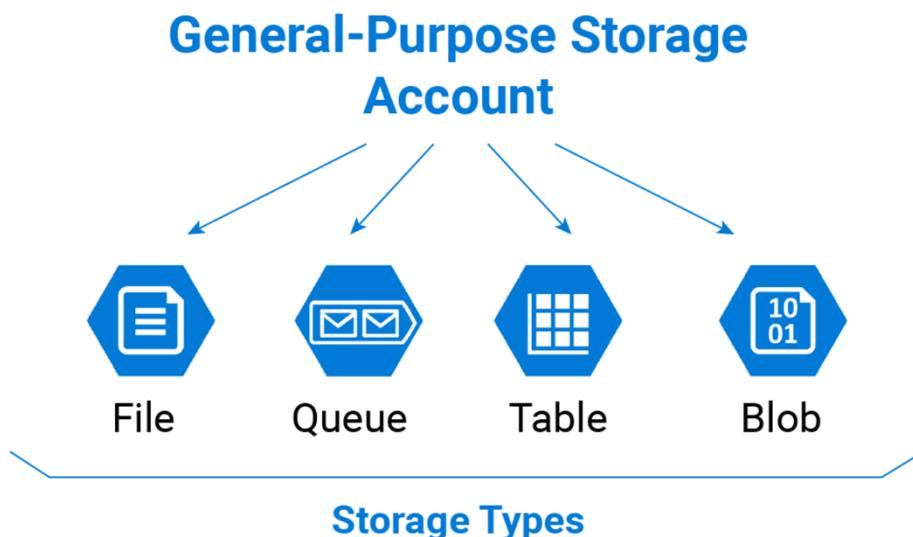


Figure 10: Schéma des différents types de services de stockage Azure

25

Aakashdubey-Ms. (2023b, décembre 10). *Vue d'ensemble du compte de stockage - Azure Storage*. Microsoft Learn. Consulté le 10 mai 2024, à l'adresse <https://learn.microsoft.com/fr-fr/azure/storage/common/storage-account-overview>

Les différents types de services de stockage Azure

Table de stockage Azure²⁶

Dans notre projet

Dans le contexte de notre projet, nous avons fait usage des tables afin de stocker les informations relatives aux alertes, aux employés et aux groupes de caméras de surveillances avec les modèles de détection d'objets qui y sont associés.

Voici un aperçu des tables utilisées :

<input type="checkbox"/>	Name
<input type="checkbox"/>	Alert
<input type="checkbox"/>	Employees
<input type="checkbox"/>	GroupModels

Figure 11: Tables de stockage Azure utilisées dans le projet

De quoi s'agit-il ?

La table de stockage Azure est un service de stockage NoSQL basé sur le cloud qui permet de stocker de grandes quantités de données structurées de manière non relationnelle. C'est un magasin de clés/attributs avec un design sans schéma. Elle possède de nombreux avantages comme sa capacité à s'adapter à des volumes de données croissants, son modèle de tarification à l'utilisation et sa haute disponibilité.

Qu'en est-il des identifiants ?

La PartitionKey²³ et la RowKey²³ sont des éléments essentiels de la conception de notre base de données. Ensemble, elles forment la clé primaire d'une entité, permettant ainsi de l'identifier de manière unique.

La PartitionKey correspond à la première partie de la clé primaire d'une entité, elle permet d'organiser les données de la table en partition.

Une partition est une plage consécutive d'entités possédant la même valeur de clé de partition. Cela permet de faciliter le stockage et la récupération des entités.

La RowKey est la deuxième partie de la clé primaire, c'est un identifiant unique pour une entité au sein d'une partition donnée.

26

Aakashdubey-Ms. (2023, 29 mars). *Introduction au Stockage (d'objets) Blob - Azure Storage*. Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/fr-fr/azure/storage/blobs/storage-blobs-introduction>

Stockage Blob Azure²⁷



Dans notre projet

Ce service est utilisé afin de stocker nos images contenant les éléments perturbateurs détectés et des clés SAS.

De quoi s'agit-il ?

Le stockage blob Azure est un service Azure cloud qui permet stocker de grandes quantités de données non structurées.

Les stockages Blob prennent en charge 3 types de données :

1. Objets blob de bloc : stockent du texte et des données binaires.
2. Objets blob d'ajout : ils sont similaires aux objets blob de bloc mais ceux-ci sont optimisés pour les opérations d'ajout.
3. Objets blob de pages : stockent des fichiers à accès aléatoire.

²⁷

Seanmcc-Msft. (2023c, juin 27). *Understanding the Table service data model (REST API) - Azure Storage*. Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/en-us/rest/api/storageservices/understanding-the-table-service-data-model>

4. Azure AI Custom Vision²⁸



Dans notre projet

Azure AI Custom Vision est un élément clé dans le développement de CoucouAI. Ce service nous a permis de créer nos modèles intelligents de détection d'objets, de les entraîner et de les exporter afin de les implémenter dans notre solution. Nous avons créé et entraîné un total de trois modèles : un modèle de détection d'œufs, un modèle de détection de mains et un modèle de détection de souris.

De quoi s'agit-il ?

Azure AI Custom Vision est un service intelligent de reconnaissance d'image proposé par Azure, il offre la possibilité de créer et d'entraîner des modèles de détection d'objets personnalisés via une interface graphique conviviale, ce qui facilite le processus de création et d'entraînement de modèles. Une fois que le modèle est créé et entraîné, celui-ci peut être déployé en tant que service web Azure afin d'être utilisé en temps réel ou être exporté sous différents formats. Dans notre cas, nous avons utilisé le format .tflite (TensorFlow Lite), qui est le format le plus adapté à notre cas d'utilisation étant donné son efficacité en termes de taille et de performances pour le déploiement sur des appareils embarqués ou mobiles.

Voici un aperçu des étapes à effectuer afin de procéder à la création et à l'entraînement d'un modèle :

1. Création du modèle

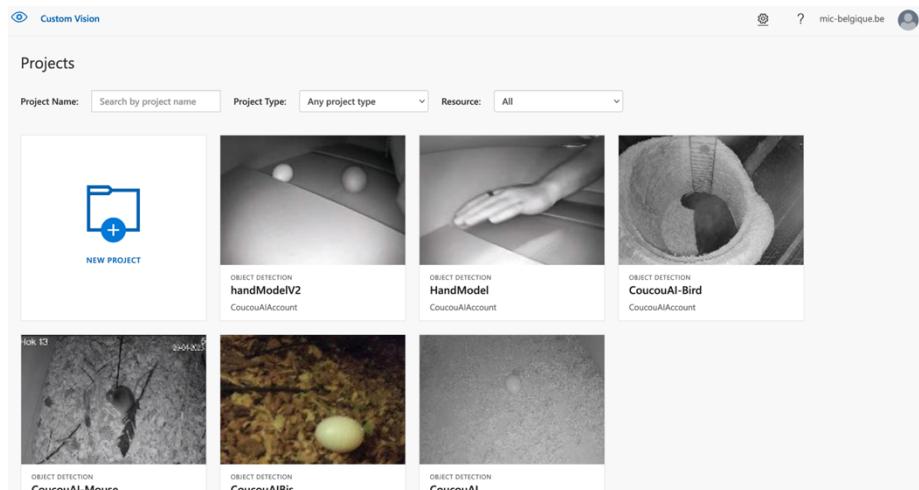


Figure 12: Interface regroupant les modèles Azure Custom Vision

28

PatrickFarley. (2024, 21 mars). *Qu'est-ce que Custom Vision ? - Azure AI services*. Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/fr-fr/azure/ai-services/custom-vision-service/overview>

Create new project

X

Name*

NomDuProjet

Description

Enter project description

- Please Choose -

Main Subscription - Microsoft Azure Sponsorship

CoucouAIAccount [S0]

[Manage Resource Permissions](#)

Project Types ⓘ

- Classification
- Object Detection

Classification Types ⓘ

- Multilabel (Multiple tags per image)
- Multiclass (Single tag per image)

Domains:

- General [A2]
- General [A1]
- General
- Food
- Landmarks
- Retail
- General (compact) [S1]
- General (compact)
- Food (compact)

Figure 13: Interface permettant la création d'un modèle de détection d'objets Azure Custom Vision

Premièrement, il est important de donner un nom de modèle pertinent ainsi que de choisir la bonne ressource, elle va permettre d'optimiser la gestion des coûts du service ainsi que ses performances.

Ensuite, un type de projet doit être choisi, il en existe deux : la classification et la détection d'objet. La classification consiste à attribuer une étiquette à une image dans son ensemble contrairement à la détection d'objet qui consiste à détecter des objets spécifiques dans une image. À titre d'exemple, pour notre première phase de développement du projet, nous avons utilisé un modèle de classification pour classifier des images de chiens et de chats. Par après, nous avons utilisé des modèles de détection d'objets afin de détecter des éléments dans une image, dans notre cas, des œufs, des mains et des souris.

2. Fournir un ensemble de données

La deuxième étape est de fournir un ensemble de données, aussi appelé dataset, qui est une collection d'images contenant les éléments que l'on souhaite détecter ou au contraire, des images ne les contenant pas.

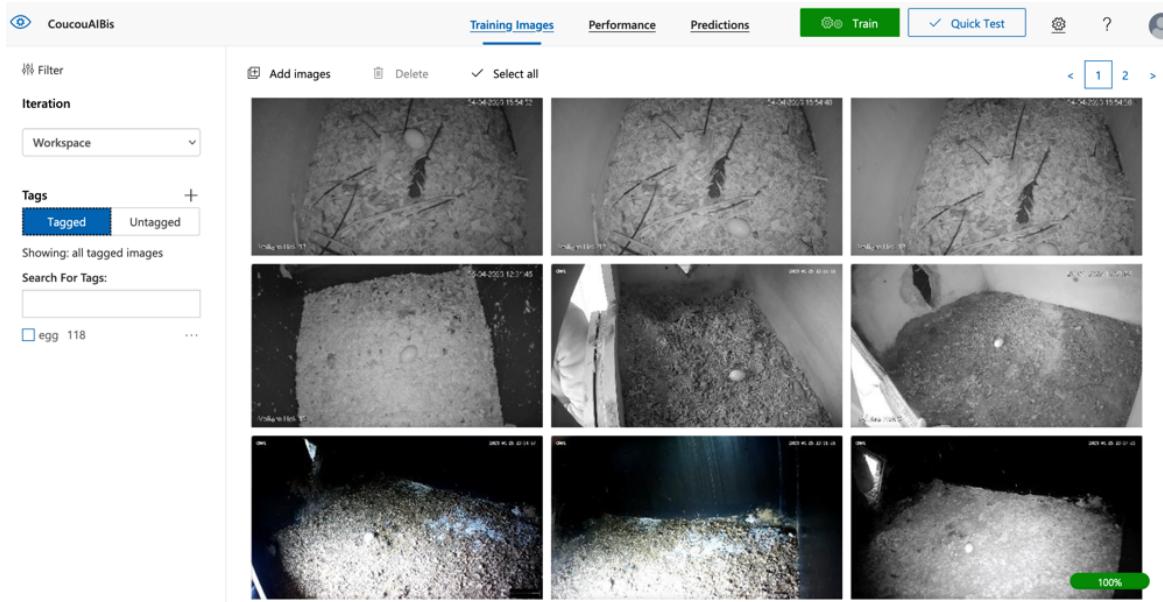


Figure 14: Aperçu du dataset dans l'interface Azure Custom Vision

3. Étiquetage de nos images

Ensuite, il est nécessaire d'étiqueter chaque image avec nos propres libellés. Pour CoucouAI, nous avons fourni des images de nids de coucous avec des œufs, que nous avons entourés et étiquetés afin de créer notre modèle de détection d'œufs.

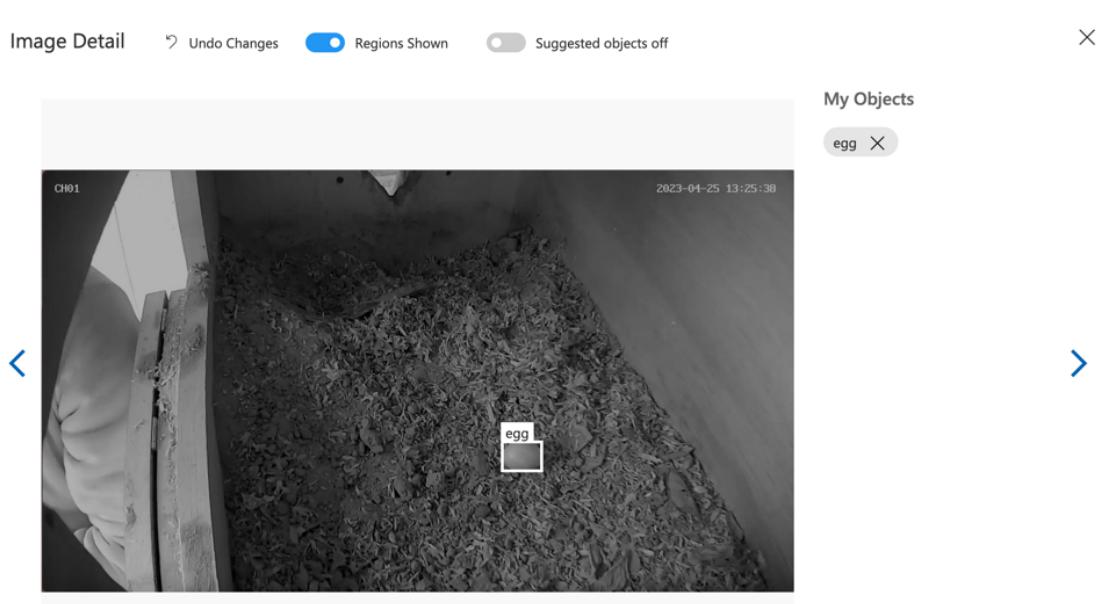


Figure 15: Aperçu de l'étiquetage d'un élément sur image via Azure Custom Vision

4. Entrainement du modèle

L'avant dernière étape consiste à l'entraînement du modèle, il est possible d'entraîner plusieurs fois un modèle, ce qui crée plusieurs itérations de ce dernier.

L'entraînement peut se faire de différentes manières, il existe l'entraînement rapide et l'entraînement avancé. Chaque entraînement engendre des frais plus ou moins conséquents selon sa durée.

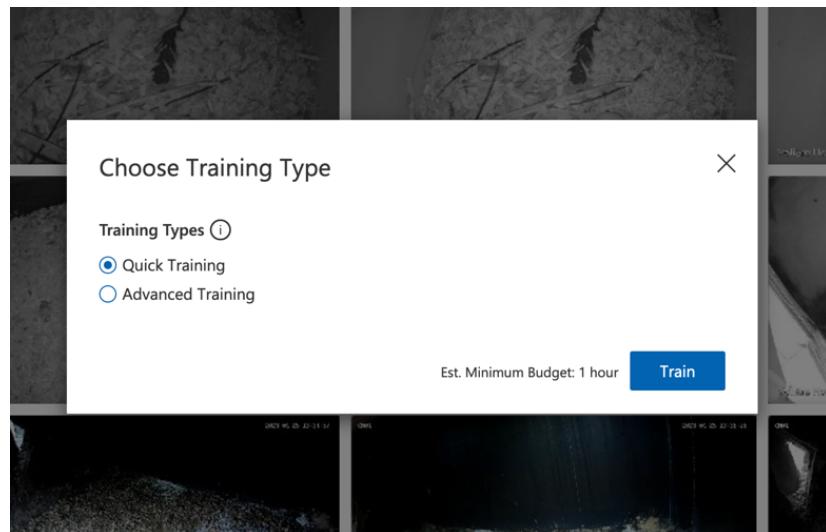


Figure 16: Aperçu des types d'entraînement via Azure Custom Vision

Dès que l'entraînement est terminé, nous obtenons différentes données comme la précision, le rappel et le mAP.

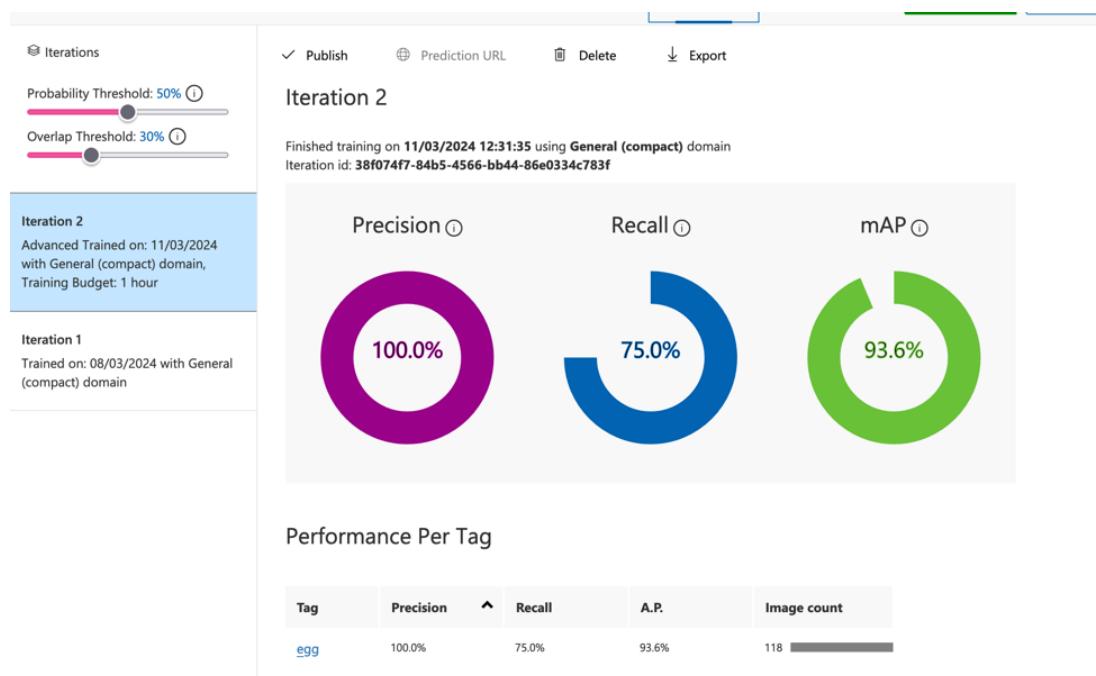


Figure 17: Aperçu des données de performance après l'entraînement d'un modèle via Azure Custom Vision

La précision mesure la proportion d'éléments détectés par le modèle qui sont réellement pertinents, en d'autres termes, elle indique la capacité du modèle à ne pas détecter à tort des éléments comme existants.

$$Précision = \frac{Nombre\ de\ vrais\ positifs}{Nombre\ de\ vrais\ positifs + Nombre\ de\ faux\ positifs}$$

Le rappel est la capacité du modèle à détecter correctement tous les éléments pertinents.

$$Rappel = \frac{Nombre\ de\ vrais\ positifs}{Nombre\ de\ vrais\ positifs + Nombre\ de\ faux\ négatifs}$$

Le mAP, mean Average Precision permet de mesurer la performance globale du modèle en prenant en compte la précision et le rappel.

5. Exportation du modèle

Enfin, chaque itération peut être exportée sous différents formats ou utilisée via un URL.

Choose your platform ×

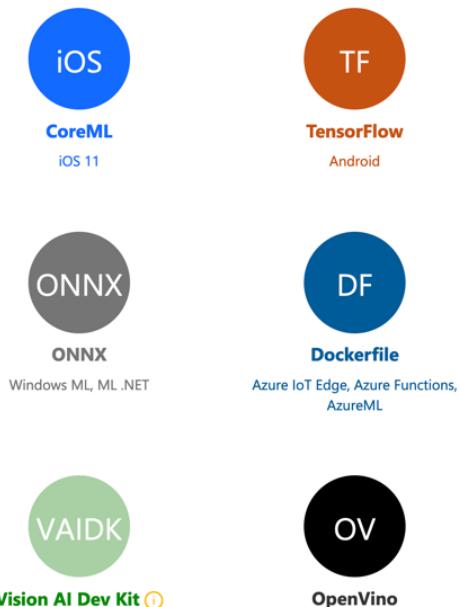


Figure 18: Aperçu des différents formats d'exportation sur Azure Custom Vision

G. Méthodologie Agile Scrum²⁹

La méthode Agile Scrum est une méthode de gestion projet qui repose sur le développement par itérations.

Scrum (= mélée) est le diminutif du mot « scrummage », un terme qui provient du monde du rugby, il signifie reprendre le jeu après une faute mineure. À la reprise du jeu, les joueurs se rassemblent pour prendre possession de la balle et travailler ensemble pour avancer sur le terrain. Cette équipe de rugby représente une équipe de développement travaillant ensemble pour faire avancer un projet et cela de manière itérative. Une itération, aussi appelée sprint, peut durer de deux à quatre semaines.

Pour chaque sprint, la première étape est la définition d'une liste d'objectifs précis à atteindre, chaque objectif aura un degré d'importance, les plus importants seront développés en premier. Un sprint est rythmé par des brèves réunions quotidiennes appelées aussi « Daily Meetings ». L'objectif des Daily Meetings est de se réunir avec d'autres professionnels pour partager ce qu'on a réalisé, les objectifs du jour et de partager les éventuelles difficultés rencontrées.

À la fin de chaque sprint, une revue et une rétrospective sont effectuées. La revue consiste à la démonstration du travail qui a été effectué et au rassemblement des différents retours sur celui-ci. La rétrospective implique d'examiner ce qui s'est bien passé, ce qui ne s'est pas bien passé et définir les éléments à améliorer pour la prochaine itération.

Pour notre projet, nous avons travaillé avec des itérations d'une durée de deux semaines. Après chaque sprint, une réunion était tenue avec Monsieur Kurt EVERAERTS, le porteur de projet, celui-ci partageait alors son niveau de satisfaction par rapport aux avancées du projet, ainsi que les nouveaux éléments à ajouter ou les améliorations à apporter.

Chaque jour à 9h20, une réunion quotidienne de quinze minutes était tenue avec les groupes de stagiaires utilisant des technologies similaires aux nôtres.



Figure 19: Schéma illustrant la méthode Agile Scrum

29

Agile scrum methodology : what to know | Inflectra. (s. d.). Consulté le 8 avril 2024, à l'adresse <https://www.inflectra.com/Solutions/Methodologies/Scrum.aspx>

V. Description détaillée des différentes phases de développement

Dans cette partie, nous trouverez la description détaillée de chaque phase de développement du projet, incluant les objectifs pour chaque sprint, l'état d'avancement du projet à la fin de ces derniers, les technologies utilisées, et les divers retours du porteur de projet, Monsieur Kurt EVERAERTS.

Phase 1 : Découverte de l'environnement de travail, Documentation et Prototypage

Phase 1.A

Technologies utilisées : Next.js, Azure Custom Vision, API Fetch

La première tâche qui nous a été confiée a été de se documenter sur les nombreuses technologies qui nous étaient jusqu'ici inconnues, parmi celles-ci, on peut retrouver le stockage Blob Azure, les fonctions Azure, Next.js et Azure Custom Vision. Il a été primordial de bien comprendre le fonctionnement des services Azure afin de mener à bien notre travail.

Ensuite, nous avons pu entamer la phase A de prototypage qui consiste au développement d'une application web en utilisant Next.js. Cette application web doit intégrer une fonctionnalité de zone drag and drop permettant le dépôt d'images par les utilisateurs. Ensuite, un tableau affichera un aperçu de l'image, ses caractéristiques et la prédition obtenue par notre modèle de prédition Azure Custom Vision déployé sur Azure. Dans notre cas, le modèle est capable de détecter si l'animal contenu dans l'image est un chien ou un chat.

Phase 1.B

Technologies utilisées : Next.js, Azure Custom Vision, Compte de stockage Azure, Fonctions Azure.

La phase B consiste à l'utilisation des fonctions Azure afin d'effectuer les prédictions sur les images téléchargées, à stocker ces prédictions et à sécuriser l'accès à nos conteneurs Blob Azure. Cette approche vise à améliorer la gestion des ressources en évitant d'avoir un backend hébergé en permanence et ainsi, réduire les coûts liés à l'utilisation des services Cloud.

Nous avons utilisé deux types de fonctions différentes :

- 1) Une fonction de type « blob trigger », à chaque image téléchargée et stockée dans un conteneur précis, la fonction sera déclenchée, elle permettra alors le traitement de l'image par Azure Custom Vision et le stockage de la prédiction obtenue dans un conteneur blob.
- 2) La deuxième fonction utilisée est de type « HTTP trigger », celle-ci permet de générer une clé SAS temporaire pour un conteneur de stockage Blob, la stocke dans un conteneur et la renvoie si nécessaire. Elle vérifie également l'expiration de la clé et la renouvelle en cas de besoin. Cette fonction permet donc de fournir un accès temporaire et sécurisé aux conteneurs de stockage Blob sans risque d'exposer les clés d'accès.

A la fin de cette phase, le prototype était fonctionnel, celui-ci a été présenté lors du premier rendez-vous avec notre porteur de projet, Monsieur Kurt EVERAERTS.

Voici un aperçu du résultat obtenu :

The screenshot shows the homepage of the CoucouAI Prototype. On the left, there is a large dashed rectangular area with the placeholder text "Drag and drop an image here" and a downward arrow icon. Below this area, it says "Only .jpg, .jpeg, .png, .bmp, .gif files allowed". At the bottom of this section is a teal-colored "Upload" button. To the right of this area is a table with a light blue header row containing the columns: "Image", "Name", "Type", and "Prediction". The main body of the table is currently empty, with the message "Table's feeling a bit shy – no elements in the spotlight yet! 😊".

Figure 20: Aperçu de la page d'accueil du prototype CoucouAI

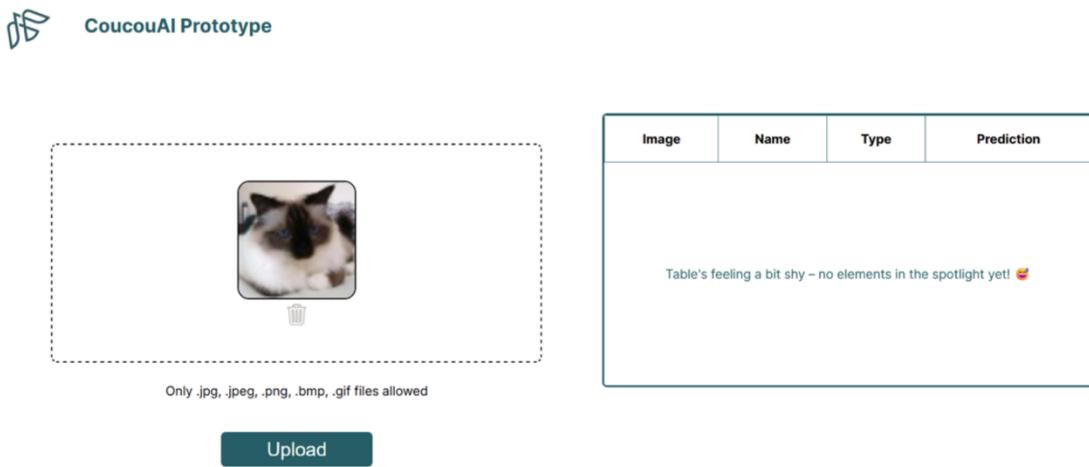


Figure 22: Aperçu du téléchargement d'une image dans le prototype CoucouAI

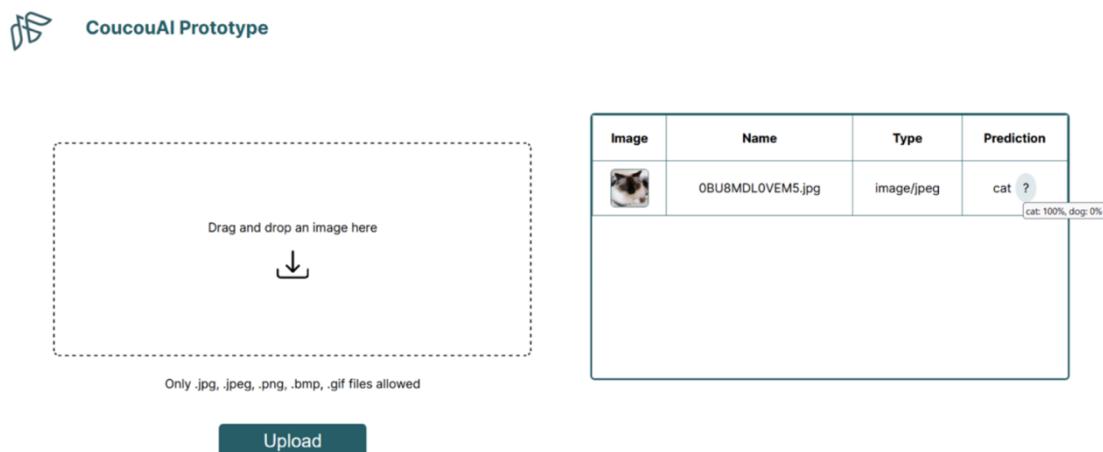


Figure 21: Aperçu du résultat d'une prédiction pour le prototype CoucouAI

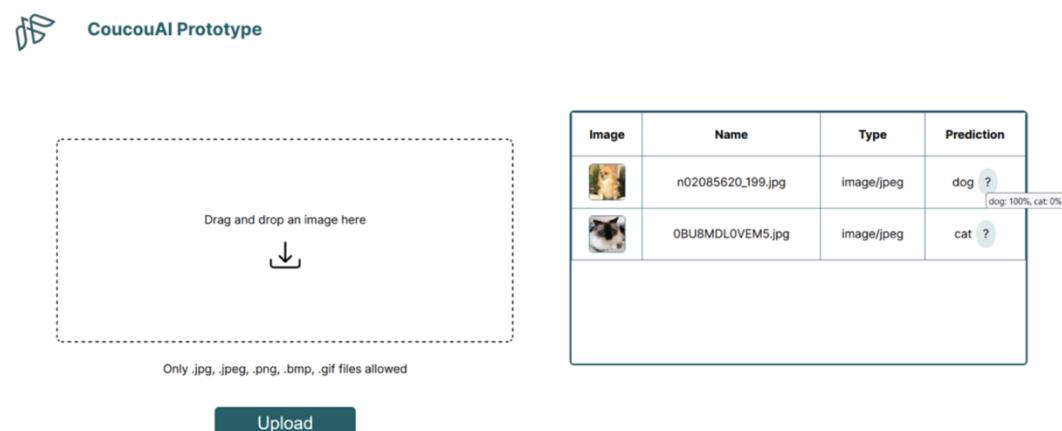


Figure 23: Aperçu du résultat d'une prédiction pour le prototype CoucouAI

Après avoir achevé nos objectifs pour le prototype, notre maître de stage, Monsieur Lloyd COLART, nous a assigné une nouvelle technologie sur laquelle nous devions nous documenter. Cette technologie se nomme YOLO.

YOLO, « You Only Look Once », est un algorithme de détection d'objets dans une image.

Après m'être documentée, je me suis lancée dans un petit projet afin de tester cet algorithme.

Le projet utilise le langage de programmation Python et le dataset COCO afin d'analyser le contenu d'une vidéo.

Voici un aperçu du résultat obtenu :

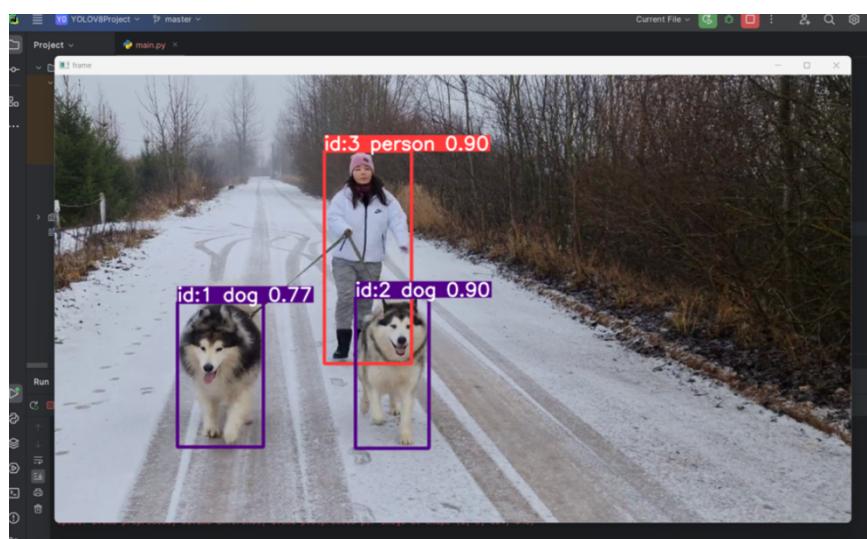


Figure 24: Aperçu du résultat obtenu du test YOLO

Cette découverte était enrichissante mais la technologie n'a finalement pas été utilisée pour CoucouAI.

Phase 2 : Premier sprint, Définition claire du projet et Premiers aléas

Technologies utilisées : Next.js, Azure Custom Vision, compte de stockage Azure, C# (.NET).

Le 21 février, nous avons eu notre premier sprint meeting avec Monsieur Kurt EVERAERTS, cet entretien nous a permis de débuter notre premier sprint officiel qui a débuté le lundi 26 février.

L'objectif de ce sprint a été le développement d'une application web permettant aux utilisateurs de créer et d'entrainer leurs propres modèles Azure Custom Vision et d'ensuite, pouvoir les utiliser via la webcam de leur ordinateur. Le site devait être composé d'un onglet pour la création et l'entraînement de modèles. L'utilisateur devait pouvoir sélectionner un modèle existant, déposer dans la zone drag and drop un dossier avec les images permettant l'entraînement du modèle et les étiqueter une à une. Enfin, l'utilisateur pourra utiliser le modèle qu'il a entraîné afin de procéder à une détection d'images en direct via sa webcam.

Ce processus a nécessité beaucoup de réflexion, notamment afin de déterminer la manière adéquate de récupérer les modèles existants et leurs paramètres. Pour ce faire, nous avons entrepris le développement d'une Web API en C#.

Ce choix de projet a été motivé par le désir du client d'appliquer cette technologie à plusieurs espèces d'animaux et donc, d'offrir la possibilité aux utilisateurs de créer et d'entrainer leurs propres modèles.

Alors que l'implémentation était presque achevée, l'idée a été abandonnée car le client souhaitait finalement utiliser la plateforme Azure Custom Vision afin de créer et entraîner les modèles. En effet, il n'y avait pas de plus-value réelle sur la reproduction d'une plateforme déjà effective et contenant des fonctionnalités avancées.

Ce processus nous a permis de consolider nos connaissances en C# et en développement d'API, tout en améliorant la structure globale de notre projet.

L'objectif a alors changé, nous devions développer une interface contenant un tableau de bord affichant les alertes lorsqu'un œuf de coucou a été détecté et un aperçu de la webcam. Les éléments détectés seront entourés d'un cadre rouge sur l'aperçu de la webcam.

Nous avons entraîné un modèle avec les données que le client, Monsieur Kurt EVERAERTS, nous a fournies via un SharePoint. Nous avons ensuite intégré le modèle dans notre page web, pour détecter la présence d'œufs via la webcam.

Pour ce faire, nous avons utilisé une caméra virtuelle à l'aide du logiciel OBS afin de transmettre une vidéo issue d'une caméra de surveillance de cage de coucous.

À la fin du sprint, nous sommes parvenus à obtenir le résultat suivant :

The screenshot shows the CoucouAI software interface. At the top left is the logo 'CoucouAI'. Below it is a video feed window showing a circular bird cage. A small red square indicates the area being analyzed by the AI. The timestamp in the top right corner of the video feed is '25-04-2023 12:38:00'. To the right of the video feed is a data table with four columns: 'Type', 'Cage', 'Time', and 'In Charge'. Two rows of data are present:

Type	Cage	Time	In Charge
Egg	0	08/03/2024 11:58:25	Kurt
Egg	0	08/03/2024 11:58:31	Kurt

Below the table is a message: 'Table's feeling a bit shy – no elements in the spotlight yet! 😊'. At the bottom left is a 'Stop' button, and at the bottom right is a 'Refresh' button.

Figure 25: Aperçu du résultat obtenu pour la phase 2 du développement du projet CoucouAI

Phase 3 : Deuxième sprint, Réduction des coûts d'utilisation des services Azure, Sélection du modèle et Gestion des employés

Technologies utilisées : Next.js, Azure Custom Vision, compte de stockage Azure, C # (.NET).

Ce sprint a été axé sur trois principaux objectifs :

1. Transition vers une solution local d'analyse de vidéo.
2. Sélection du modèle via l'interface web.
3. Gestion des intervenants via l'interface web.

Pour remplir notre objectif, nous sommes passés d'un modèle de classification à un modèle de détection d'objets, cela a été facilité par l'interface graphique d'Azure Custom Vision. Cette transformation vise principalement à obtenir les coordonnées des objets détectés dans l'image, ce qui n'était pas possible avec notre modèle de prédiction initial. Nous analyserons des vidéos plutôt que des images statiques.

Après avoir créé notre modèle et réalisé un entraînement avancé, nous avons exporté celui-ci, constatant qu'il était au format TensorFlow. TensorFlow, rappelons-le, est un framework open-source développé par Google, largement utilisé pour l'apprentissage automatique, notamment le deep learning. Cette découverte nous a menés à envisager l'utilisation de TensorFlow Lite, une alternative plus légère, moins coûteuse et plus facile à déployer.

Cette transition vers une solution locale vise à minimiser les coûts associés à l'utilisation des services cloud Azure et à améliorer la réactivité de notre système. Pour simplifier les choses, nous avons décidé d'utiliser une API Python avec TensorFlow Lite dans une image Docker, en remplacement des outils Azure Custom Vision et du SDK Azure Custom Vision. Cette API Python utilise Flask, ce qui facilite grandement son déploiement et son utilisation. De plus, nous avons choisi d'utiliser les modèles compacts au format TensorFlow Lite, offerts par Azure Custom Vision, car cela nous permet de convertir aisément nos anciens modèles en nouveaux modèles TensorFlow Lite.

Le traitement vidéo fonctionne de la manière suivante : chaque seconde, le client envoie une capture vidéo simulée par notre webcam à l'API C#. Cette API C# appelle ensuite l'API Python pour récupérer la prédiction pour cette image. Les prédictions sont ensuite renvoyées au client qui affiche des rectangles colorés autour des objets détectés directement sur le flux vidéo. Pour augmenter la clarté et la précision des alertes générées, nous sauvegardons les zones de détection ainsi que les étiquettes des objets détectés, visualisés sous forme de rectangles rouges.

De plus, l'alerte est enregistrée dans un blob contenant une image de l'alerte avec l'élément détecté entouré d'un rectangle rouge.

Grâce à cette méthode, le client peut consulter en direct les alertes depuis son navigateur. Toutefois, lors de la présentation de nos progrès au client, une demande spécifique a été formulée : éviter l'envoi d'alertes multiples pour un même élément détecté. En effet, la répétition des alertes peut être problématique, car il est superflu de signaler un œuf déjà détecté. Nous allons devoir trouver une solution à ce problème lors de la prochaine étape de développement.

Nous avons ensuite réussi à afficher une liste qui permet de sélectionner le modèle que nous souhaitons appliquer. Cette approche nous permet d'envisager l'utilisation de plusieurs modèles choisis simultanément. Par exemple, nous pourrions décider d'utiliser le modèle de détection d'œufs et le modèle de détections de souris. Le modèle de détection de souris pourra également être utilisé indépendamment, par exemple, pour détecter des souris dans une cage à poules.

Enfin, la dernière étape de ce sprint a été le développement d'un nouvel onglet dédié à la gestion des différents intervenants. Il permettra la sélection des intervenants devant être contactés en cas d'alerte. Cet onglet de paramétrage offre les fonctionnalités d'un CRUD classique : consultation, ajout, modification et suppression.

En parallèle, nous nous sommes documentés sur FlexIO (Proximus), une technologie qui nous aurait permise de transmettre les alertes par appel téléphonique. Cependant, nous avons été dans l'obligation de mettre cette option de côté car l'accès à l'environnement FlexIO était impossible à notre échelle.

Phase 4 : Troisième sprint, superposition de modèles et redondances des alertes

Technologies utilisées : Next.js, compte de stockage Azure, Python (Flask), C# (.NET), Tensorflow Lite.

Nous avons eu une réunion avec Monsieur Kurt EVERAERTS qui était ravi de l'état d'avancement du projet. Nous avions un sprint d'avance.

Les prochaines étapes du développement du projet impliquent deux aspects principaux :

1. La superposition des modèles de détection d'objets.
2. La gestion de la redondance des alertes.

Nous avons introduit une fonctionnalité permettant de choisir le ou les modèles à utiliser pour effectuer les prédictions. Cette sélection est effectuée depuis l'interface web client et est gérée par l'API Python. Il est essentiel de garantir que cette API peut gérer plusieurs modèles simultanément sans affecter les temps de prédiction.

Cette amélioration va au-delà de la simple gestion des alertes, elle englobe également une personnalisation approfondie de la détection selon les besoins spécifiques de chaque enclos. Par exemple, pour une cage de coucous, nous pourrions utiliser un modèle dédié à ce type spécifique d'œuf, ainsi que des modèles plus généraux comme pour détecter les souris et les mains humaines. Cette approche adaptable permet d'ajuster la détection à chaque environnement spécifique, répondant ainsi aux divers besoins des clients.

Le côté frontend envoie une capture d'écran à l'API de prédiction en incluant les noms des modèles sélectionnés. Ensuite, les prédictions sont renvoyées par l'API pour chaque modèle et les résultats sont combinés. La redondance des alertes a également été gérée avec succès via l'API Python.

⇒ Sélection multiple de modèles.



Figure 26: Aperçu de la sélection multiple de modèles

Phase 5 : Quatrième sprint, Envoi d'e-mails et Prise en charge des alertes

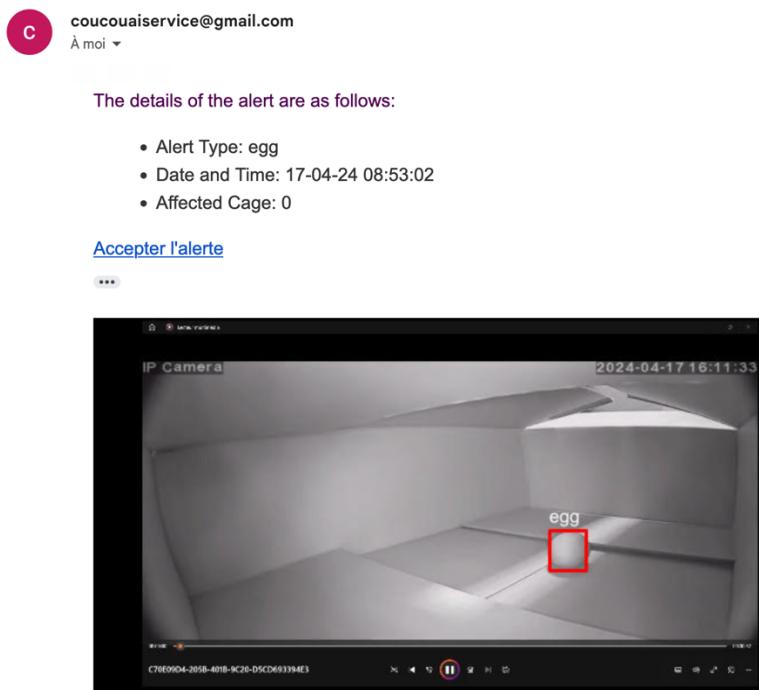
Technologies utilisées : Next.js, compte de stockage Azure, Python (Flask), C# (.NET), Tensorflow Lite, MailKit, MikeKit.

L'objectif principal de ce quatrième sprint était l'envoi d'e-mails aux intervenants disponibles en cas d'alertes.

À l'origine, le client souhaitait recevoir les alertes via FlexIO, mais étant donné que cette plateforme n'était pas disponible, nous avons déployé un système alternatif pour transmettre les alertes par e-mails. Les employés disponibles reçoivent ainsi un e-mail comprenant les détails de l'alerte ainsi qu'une image de l'élément perturbateur détecté. À travers cet e-mail, l'employé peut prendre en charge l'alerte ; il recevra alors une confirmation de prise en charge ou sera informé si l'alerte a déjà été traitée par quelqu'un d'autre. Ce module a été développé en C# et utilise les bibliothèques MikeKit et MailKit. L'envoi des e-mails est géré par un projet distinct, intégré à notre API via un DLL. Par ailleurs, nous avons décidé de ne plus utiliser l'image Docker, celle-ci étant actuellement superflue. Nous disposons désormais de deux API fonctionnant de manière autonome.

Voici un aperçu des résultats obtenus :

⇒ Mail reçu par les employés disponibles



Cordialement

Figure 27: Aperçu de l'e-mail reçu par les employés disponibles

- ⇒ Page permettant d'accepter l'alerte.

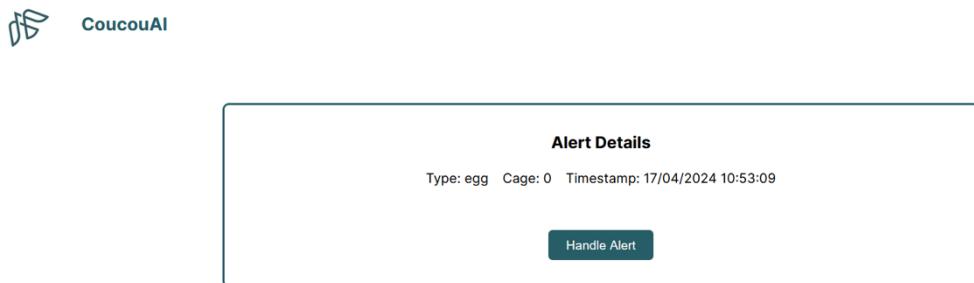


Figure 28: Aperçu de la page permettant d'accepter une alerte

- ⇒ Après avoir pris en charge l'alerte qui n'a pas encore été prise en charge.

A screenshot of the same web application interface as Figure 28. It shows the "Alert Details" page with the same alert information. However, there is a green notification bar at the top with a checkmark icon and the text "The alert has been successfully handled." In the bottom right corner of the main box, the text "You have successfully handled the alert." is visible.

Figure 29: Aperçu du résultat après avoir pris en charge une alerte

- ⇒ Après avoir tenté de prendre en charge l'alerte, mais celle-ci a déjà été prise en charge.

A screenshot of the web application interface. It features the "Alert Details" page with the same alert details as previous figures. A red notification bar at the top contains an exclamation mark icon and the text "The alert has already been handled.". In the bottom right corner of the main box, the text "Already handled." is displayed.

Figure 30: Aperçu du résultat si l'alerte a déjà été prise en charge

⇒ Onglet permettant de gérer les employés

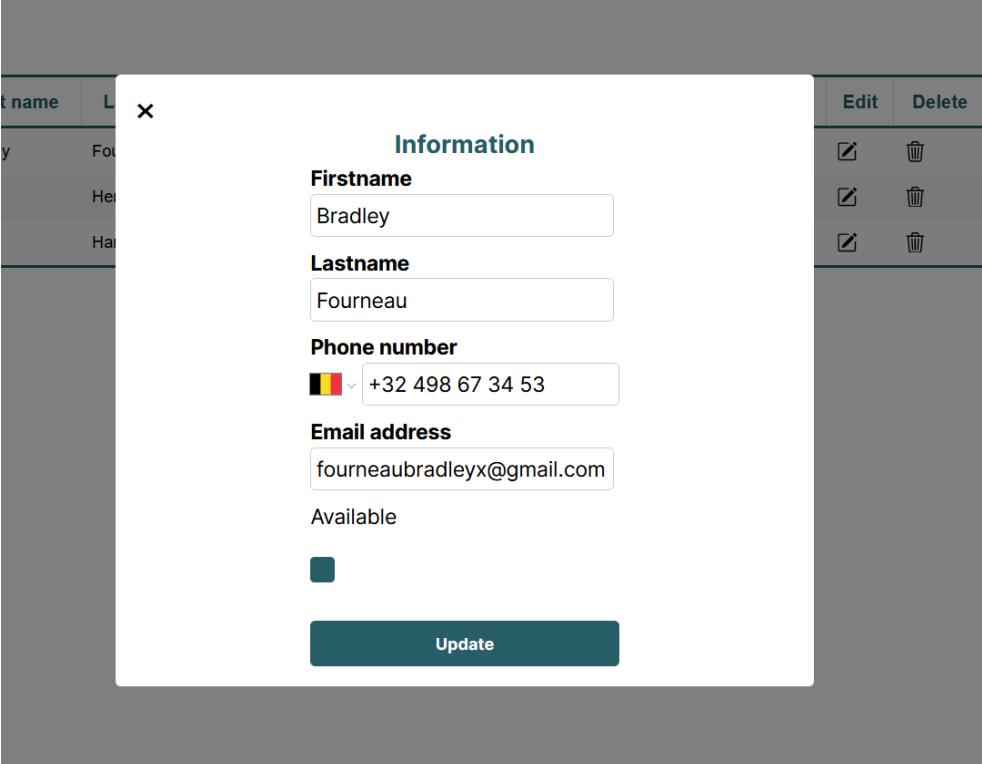


First name	Last name	Phone number	Email address	Availability	Edit	Delete
Bradley	Fourneau	+32498673453	fourneaubradleyx@gmail.com	Yes		
Terry	Henrard	+32484862661	terry.henrard@condorcet.be	Yes		
Samia	Hanini	+32484862661	samlahn.pro@gmail.com	Yes		

New employee



Figure 31: Aperçu des employés et de leurs informations



A screenshot of the employee edit interface. At the top, there is a modal window titled "Information" with fields for "Firstname" (Bradley), "Lastname" (Fourneau), "Phone number" (+32 498 67 34 53), and "Email address" (fourneaubradleyx@gmail.com). Below these fields is a "Available" status indicator with a blue square. At the bottom of the modal is a "Update" button. To the right of the modal, a portion of the employee list table is visible, showing rows for Bradley, Terry, and Samia. Each row has "Edit" and "Delete" buttons next to it.

Figure 32: Aperçu de l'interface de modification des données d'un employé

Phase 6 : Cinquième sprint, Récupération du flux de caméra en direct

Technologies utilisées : Next.js, compte de stockage Azure, Python (Flask), C# (.NET), Tensorflow Lite, MailKit, MikeKit, NxCloud, Nx Witness.

Nous avons eu l'occasion de nous entretenir à nouveau avec notre client, Monsieur Kurt EVEVAERTS dans le cadre de nos réunions sprint bimensuelles. Celui-ci était ravi des progrès réalisés, il a partagé son souhait que nous trouvions une solution afin d'utiliser un flux vidéo en direct.

Jusqu'ici, nous avons fonctionné avec des extraits vidéos fournis par Monsieur Bart DESMET, directeur de Security First Bv, entreprise se chargeant de l'équipement de surveillance des cages d'animaux. Il a eu la gentillesse de nous envoyer une caméra IP afin que l'on puisse procéder à des tests au MIC.

Après nous être attardés sur la recherche d'informations sur le protocole onvif, nous avons tenté de récupérer le flux de caméra en direct. Malheureusement, ce processus de recherche a dû être interrompu car le flux de caméra n'était plus accessible via l'application mobile. Le problème a été causé par une anomalie lors de la tentative de connexion à la caméra, en effet, même après réinitialisation de la caméra via l'application mobile, aucun mot de passe ne fonctionnait, ni celui défini par défaut, ni le mot de passe personnalisé qui fonctionnait précédemment. Nous avons tenté de faire une réinitialisation usine mais cela n'a pas été possible car aucun bouton n'a été prévu à cette fin.

Suite au souci rencontré, nous avons pris contact avec Monsieur Bart DESMET. Celui-ci est revenu vers nous afin de nous aider à résoudre cet imprévu en venant directement dans les bureaux du MIC.

Nous avons reçu deux nouvelles caméras afin de pouvoir tester la gestion de plusieurs flux de caméras simultanément. Par après, Monsieur DESMET nous a présenté la solution qu'il avait découverte afin de récupérer le flux. Pour arriver à nos fins, nous avons fait usage du logiciel Nx Witness qui a permis de gérer et stocker les flux de caméras.

Voici un aperçu du logiciel Nx Witness :

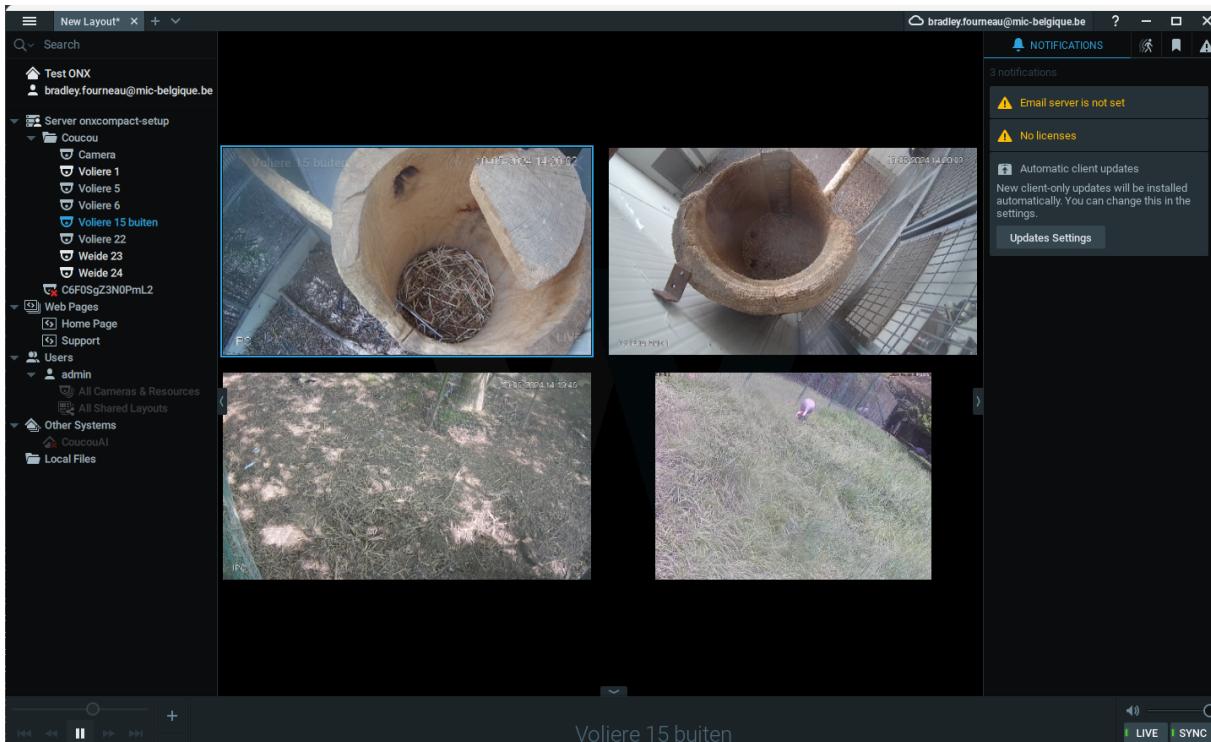


Figure 33: Aperçu du logiciel Nx Witness

VI. Implémentation

A. Infrastructure de stockage Azure

Dans notre infrastructure Azure, nous utilisons des tables pour stocker des données sur les alertes détectées, les informations sur les employés et les informations sur les modèles appliqués aux groupes de caméras. La table « Alerts » enregistre les détails des alertes tels que l'heure, le type d'alerte, l'identifiant de la cage et la personne responsable. De même, la table « Employees » conserve les informations des employés comme leur nom, leur prénom, leur adresse e-mail , leur numéro de téléphone et leur disponibilité. La table « GroupModels » permet de stocker les groupes de caméras ainsi que le nom des modèles de détection y étant appliqués. Parallèlement, nous avons mis en place deux conteneurs blob pour stocker les images liées aux alertes et mettre en évidence les éléments perturbateurs détectés. Un autre conteneur est consacré au stockage sécurisé des clés de stockage, garantissant la sécurité et le bon accès aux données sensibles.

Table Azure « Alerts »

Cette table stocke les éléments détectés par nos modèles, ainsi que leurs informations.

Elle est composée des colonnes suivantes :

- Timestamp : la date et l'heure auxquelles l'alerte est stockée.
- TypeAlert : enregistre le type de l'élément détecté.
- IdCage : Stocke l'identifiant de la cage où l'alerte s'est produite.
- IdPersonInCharge : Contient l'identifiant de la personne responsable de cette alerte.

Property Name	Type
PartitionKey	String
RowKey	String
Timestamp	DateTi...
TypeAlert	String
IdCage	String

Figure 34: Colonnes de la table Azure "Alerts"

Table Azure « Employees »

Cette table stocke les informations pour chaque employé.

Elle est composée des colonnes suivantes :

- Firstname : prénom de l'employé.
- Lastname : nom de l'employé.
- PhoneNumber : numéro de téléphone de l'employé.
- EmailAddress : adresse e-mail de l'employé.
- IsAvailable : booléen permettant de déterminer la disponibilité de l'employé

Property Name	Type
PartitionKey	String
RowKey	String
Timestamp	DateTi...
Firstname	String
Lastname	String
PhoneNumber	String
EmailAddress	String
IsAvailable	Boolean

Figure 35: Colonnes de la table Azure "Employees"

Table Azure « GroupModels»

Cette table stocke les informations pour les groupes de caméras et leurs modèles.

Elle est composée des colonnes suivantes :

- IdGroup : identifiant du groupe de caméras.
- ModeName : nom du modèle de détection d'objets.

Property Name	Type
PartitionKey	String
RowKey	String
Timestamp	DateTi...
IdGroup	String

Figure 36: Colonnes de la table Azure "GroupModels"

Stockages Blob « alert-image » et « key-storage »

Ces stockages blob stockent les images des éléments détectés et les clés de stockage SAS.

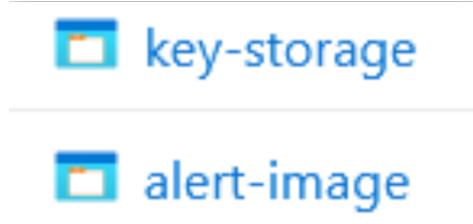
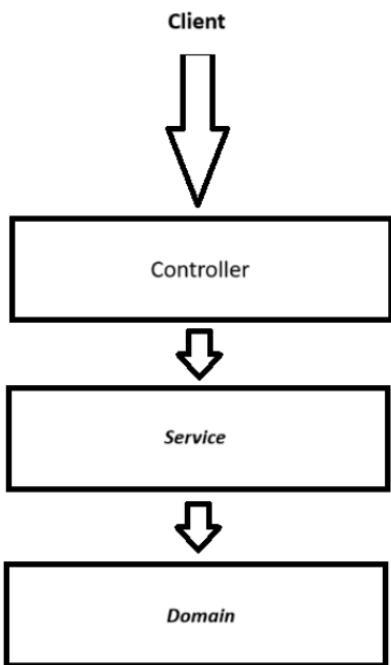


Figure 37: Conteneurs Blob Azure

B. Structure de l'API C#

Afin de construire notre API, nous nous sommes basés sur une architecture organisée en plusieurs couches distinctes.



Name
..
.config
Controllers
Domain
Properties
Services
python/DockerCoucouAI
MapperProfile.cs
Program.cs
appsettings.Development.json
appsettings.json
capture.png
coucouai-webapi.csproj
coucouai-webapi.sln
ffmpeg.exe

Figure 38: Schéma de la structure de l'API C#

Figure 39: Structure des fichiers de l'API C#

La couche Controller

Cette couche est l'interface qui communique directement avec notre application frontend. Les contrôleurs servent de point d'entrée pour l'ensemble du système, c'est cette couche qui reçoit les requêtes HTTP du frontend. Ils permettent de gérer la logique de routage des requêtes et dirigent les appels vers les services appropriés pour le traitement ultérieur.

La couche Service

La couche service se charge de la logique métier de notre application. Elle va recevoir et traiter les demandes envoyées par les contrôleurs et renvoyer les résultats ou les réponses obtenus après avoir effectué les opérations nécessaires. Elle agit comme une couche intermédiaire entre les données et les objets DTO (Data Transfer Objects). Les objets DTO sont utilisés pour transférer des données entre les différentes couches d'une application, ils contiennent uniquement les données nécessaires pour des opérations spécifiques.

La couche Domain

Cette couche est composée les entités métier de notre application.

Communication avec les services Azure

Notre API C# communique avec différents services Azure via les SDK appropriés fournis par Microsoft.

C. Structure de l'API Python

Name
..
Properties/ServiceDependencies/CoucouAI-WebAPI - W...
env
envt/share/man/man1
model
tags
app.py
coucouai-webapiPython.pyproj
clexport.manifest
libman.json
metadata_properties.json
object_detection.py
predict.py

Nous avons développé une Web API Python à l'aide du framework Flask, celle-ci reçoit les images des flux de caméras en entrée et les soumet aux modèles TensorFlow Lite pré-entraînés afin qu'elles soient analysées. Ensuite, les résultats sont renvoyés. Les modèles sont sélectionnés dynamiquement sur base des noms des modèles fournis par le frontend, ce qui offre une grande flexibilité dans le choix des modèles de prédiction. Le choix du langage Python afin de développer cette API s'explique par le fait que TensorFlow Lite est disponible uniquement en Python.

Figure 40: Structure des fichiers de l'API Python

D. Analyse du code

Dans cette section, nous allons parcourir certains extraits de code afin de faciliter la compréhension de la solution.

NB : Cette section est sujette à modification, le développement de certaines fonctionnalités est toujours en cours au moment du dépôt de cette partie de TFE.

1. Récupération des flux de caméras en direct

API C#

La méthode « getAllCamera() » utilise un HttpClient pour envoyer une requête à une API externe, récupère les informations sur les caméras, extrait les détails nécessaires pour chaque caméra et retourne une liste de caméras sous forme d'objets de type « Camera ». Cette liste va nous permettre d'extraire les images à analyser par les modèles ainsi que d'afficher un aperçu des flux de caméras de surveillances sur la page web.

```
public async Task<List<Camera>> GetAllCamera()
{
    HttpClient httpClient = new HttpClient();
    string sas = await new SasService().GetSasNx();
    httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", sas);
    String idServeur = Environment.GetEnvironmentVariable("idServeurNx");
    HttpResponseMessage responce = await httpClient.GetAsync($"https://{{idServeur}}.relay.vmsproxy.com/rest/v2/devices?deviceType=Camera");
    string jsonResult = await responce.Content.ReadAsStringAsync();
    JArray jsonArray = JArray.Parse(jsonResult);
    List<Camera> cameras = new List<Camera>();
    foreach(JObject cameraItem in jsonArray)
    {
        if (cameraItem.TryGetValue("parameters", out var groupParameter))
        {
            cameraItem.TryGetValue("name", out var cameraName);
            String cameraId = (String)cameraItem["id"];
            cameraId=cameraId.Replace("{", "");
            cameraId=cameraId.Replace("}", "");
            String groupName = (String)cameraItem["parameters"]["customGroupId"];
            if(groupName != null)
            {
                Camera cam = new Camera()
                {
                    CameraName = (string)cameraName,
                    StreamUrl = $"https://{{idServeur}}.relay.vmsproxy.com/media/{{cameraId}}.mp4",
                    GroupName = groupName
                };
                cameras.Add(cam);
            }
        }
    }
    return cameras;
}
```

Figure 41: Code de la méthode GetAllCamera() de l'API C#

2. Extraction des images des flux de caméras de vidéosurveillance et envoi à l'API Python pour analyse.

API C#

L'extrait du code suivant provient de la classe « UploadImageHostedService » qui est hérité de « BackgroundService » et qui permet ainsi d'exécuter des tâches en arrière-plan. Cette classe est composée de plusieurs éléments. Dans le cadre jaune, vous retrouvez le fragment permettant de récupérer le flux vidéo grâce au données récupérées suite à l'appel la méthode « GetAllCamera() » de la classe « CameraService ». En rouge, vous retrouvez le code correspondant à la capture de l'image à partir du flux vidéo, enfin, en bleu, vous retrouvez l'envoi de l'image à la méthode « UploadImage » de la classe « ModelService » que nous analysons brièvement plus bas.

```
var modelService = scope.ServiceProvider.GetRequiredService<ModelService>();
var cameraService = scope.ServiceProvider.GetRequiredService<CameraService>();
using (HttpClient client = new HttpClient())
{
    string sas = await new SasService().GetSasNx();
    client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", sas);
    var absoluteFilePath = Path.GetTempFileName();
    foreach (Camera camera in await cameraService.GetAllCamera())
    {
        Console.WriteLine("camera " + camera.CameraName);
        Console.WriteLine("camera " + camera.StreamUrl);
        try
        {
            List<string> models = await modelService.GetModelsByGroupName(camera.GroupName);

            string videoUrl = camera.StreamUrl;
            using (var response = await client.GetAsync(videoUrl, HttpCompletionOption.ResponseHeadersRead))
            {
                var stream = await response.Content.ReadAsStreamAsync();
                try
                {
                    FFMpegArguments
                        .FromPipeInput(new StreamPipeSource(stream),
                            options => options
                                .ForceFormat("mp4"))
                        .OutputToFile("capture.png", true,
                            options => options
                                .WithVideoCodec(VideoCodec.Png)
                                .WithFrameOutputCount(1))
                        .ProcessSynchronously(true);
                }
                catch (Exception e)
                {
                }
            }
            using (var fileStream = File.OpenRead("capture.png"))
            {
                IFormFile formFile = new FormFile(fileStream, 0, fileStream.Length, "imageData", "capture.png");
                var result = await modelService.UploadImage(formFile, models);
            };
        }
    }
}
```

Figure 42: Extrait de code de la classe "UploadImageHostedService" de l'API C#

La méthode « UploadImage » de la classe « ModelService » remplit plusieurs missions cruciales :

- > La récupération des modèles.
- > L'envoi de l'image et des modèles via une requête HTTP POST au serveur Python.
- > Le traitement des prédictions renvoyés par l'API Python.
- > La création des entités de prédiction pour les alertes non redondantes.
- > L'envoi de l'image et des détails via mail.
- > Le retour de la liste des prédictions.

Ci-dessous, vous pouvez observer l'envoi d'une image et des modèles à l'API Python, ainsi que la récupération de la réponse au format JSON.

```
HttpClient client = new HttpClient();
var formdata = new MultipartFormDataContent();
formdata.Add(new StreamContent(file.OpenReadStream()), "imageData", file.FileName);
StringContent stringContent = new StringContent(json, Encoding.UTF8, "application/json");
formdata.Add(stringContent, "modelNames");
HttpResponseMessage responce = await client.PostAsync("http://127.0.0.1:8080/image", formdata);
string jsonResult = await responce.Content.ReadAsStringAsync();
JArray jsonArray = JArray.Parse(jsonResult);
```

Figure 43: Extrait de code de la méthode UploadImage() de l'API C#

3. Récupération des modèles et prédition sur image

API Python

Afin de récupérer nos modèles, nous avons en premier lieu organisé nos modèles de la manière suivante :

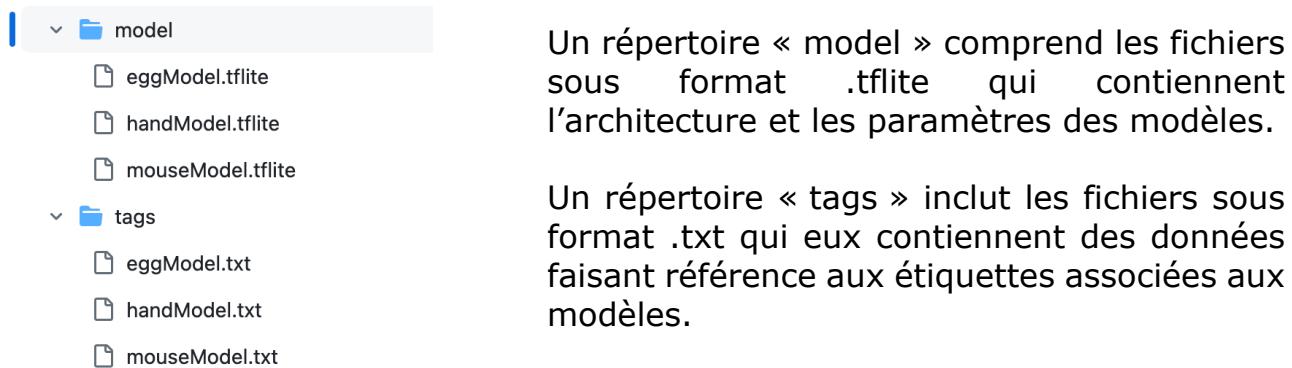


Figure 44: Structure des dossiers contenant les modèles et les labels dans l'API Python

App.py

➔ Renvoie de la liste des modèles de détection d'objets

```
@app.route('/models', methods=['GET'])
def models():
    return os.listdir("./model")
```

Predict.py

C'est la méthode « predict_image » qui va permettre d'effectuer des prédictions sur une image.

```
def predict_image(image, modelName):
```

Dans cette méthode, un objet TFObjectDetection est créé en utilisant un modèle spécifique et une liste de labels. La méthode « predict_image » de l'objet TFObjectDetection est ensuite appelée pour effectuer la détection d'objets sur l'image fournie. Cette méthode utilise l'interpréteur TensorFlow Lite pour charger le modèle et effectuer la prédition. Après avoir analysé la prédition pour déterminer si celle-ci est redondante ou non, le résultat de la prédition est renvoyé.

```
labels = [s.strip() for s in lines[2:]]
od_model = TFObjectDetection("./model/" + modelName, labels, float(threshold))
predictions = od_model.predict_image(image)
```

4. Gestion de la superposition des modèles

Ce fragment de code issu du module « app.py » permet d'extraire la liste des noms de modèles des données de la requête. Ensuite, la fonction « load_and_predict » charge une image à partir des données fournies et appelle la fonction « predict_image » du module « predict.py » avec l'image et le nom de modèle spécifique. Enfin, il utilise un ThreadPoolExecutor pour charger et prédire de manière simultanée avec chaque modèle, ce qui permet le traitement des images avec plusieurs modèles en parallèle.

App.py

```
@app.route('/image', methods=['POST'])
def predict_image_handler():
    try:
        modelNames_json = request.form.get('modelNames')
        if modelNames_json:
            modelNames = json.loads(modelNames_json)
        else:
            return jsonify({"error": "modelNames JSON not found"}), 400
        imageData = None
        if ('imageData' in request.files):
            imageData = request.files['imageData'].read()
        elif ('imageData' in request.form):
            imageData = request.form['imageData']
        else:
            imageData = request.get_data()
        results = []

        def load_and_predict(modelNameT):
            img_data = io.BytesIO(imageData)
            img = Image.open(img_data)
            return predict_image(img, modelNameT)

        with ThreadPoolExecutor(max_workers=5) as executor:
            futures = [executor.submit(load_and_predict, modelName) for modelName in modelNames]
        for future in futures:
            results.append(future.result())

    return results
    except Exception as e:
        print('EXCEPTION APP:', str(e))
        return 'Error processing image', 500
```

Figure 45: Extrait de code de l'API Python pour la superposition des modèles

5. Gestion de la redondance des alertes

En premier lieu, ce fragment de code provenant de la méthode « predict_image » vérifie l'existence d'une prédition précédente. Ensuite, il calcule un nouveau temps de détection en ajoutant quinze secondes au dernier temps de détection. Enfin, il compare le nouveau temps de détection avec l'heure actuelle, si le nouveau temps de détection est antérieur à l'heure actuelle, cela signifie que la prédition est redondante.

Predict.py

```
for prediction in predictions:  
    if(lastDetected!="1"):  
        date_Detected = datetime.fromisoformat(lastDetected)+timedelta(seconds=15)  
        if(date_Detected<date_now):  
            lines[1] = "lastTime ;"+str(date_now.isoformat())  
            file_path.write_text('\n'.join(lines))  
            prediction["isRedundant"]=False  
        else:  
            lines[1] = "lastTime ;"+str(date_now.isoformat())  
            file_path.write_text('\n'.join(lines))  
            prediction["isRedundant"]=True  
    else:  
        lines[1] = "lastTime ;"+str(date_now.isoformat())  
        file_path.write_text('\n'.join(lines))  
        prediction["isRedundant"]=False
```

Figure 46: Extrait de code de l'API Python pour la redondance des alertes

6. Récupération des alertes

API C#

La méthode asynchrone « GetAllAlert() » a été développée afin d'effectuer une requête pour accéder à la table de stockage Azure et récupérer les données relatives aux alertes détectées. Elle va récupérer le timestamp, le type d'alerte, l'identifiant de la cage et de la personne en charge. Pour chaque entité qui contient un champ « TypeAlert », la méthode récupère des informations supplémentaires à partir d'un conteneur blob Azure en utilisant un SAS (Shared Access Signature) pour l'accès sécurisé. Enfin, elle renvoie une liste d'objets « DtoOutputAlert » contenant les informations qui serviront alors à alimenter le tableau de bord de notre page web.

```
public async Task<List<DtoOutputAlert>> GetAllAlert()
{
    List<DtoOutputAlert> alerts = new List<DtoOutputAlert>();
    var tableClient = _tableService.GetTableClient("Alert");
    var items = tableClient.Query<TableEntity>("");
    string sas = await new SasService().GetSas("alert-image");
    var blobServiceClient = new BlobServiceClient(new Uri(sas));
    BlobContainerClient blobContainer = blobServiceClient.GetBlobContainerClient("");

    var allAlerts = items.ToList();

    var sortedAlerts = allAlerts.OrderByDescending(a => a.GetDateTime("Timestamp"));

    foreach (var item in sortedAlerts)
    {
        if (item.ContainsKey("TypeAlert"))
        {
            string alert = item.GetString("TypeAlert");
            DateTime time = (DateTime)item.GetDateTime("Timestamp");
            string idCage = item.GetString("IdCage");
            var blob = blobContainer.GetBlobClient(item.GetString("RowKey"));
            if (alert != null)
            {
                DtoOutputAlert outputAlert = new DtoOutputAlert()
                {
                    Timestamp = time,
                    TypeAlert = alert,
                    IdCage = idCage
                };
                alerts.Add(outputAlert);
            }
        }
    }
    return alerts;
}
```

Figure 47: Code de la méthode GetAllAlert() de l'API C#

VII. Phase de test

VIII. Améliorations potentielles

IX. Activités et formations offertes par le MIC

X. Compétences acquises

Au cours de mon stage, j'ai eu l'opportunité d'acquérir et de perfectionner des compétences essentielles pour ma vie professionnelle. En voici quelques exemples.

a. Compétences comportementales

➔ Communication et esprit d'équipe

Grâce au travail effectué en binôme et aux réunions tenues avec le client, je me sens désormais beaucoup plus à l'aise dans la collaboration en équipe et dans la communication avec d'autres professionnels.

➔ Gestion du temps

Notre client, Monsieur Kurt EVERAERTS, attendait des résultats concrets à la fin de chaque sprint. Il était donc essentiel de gérer efficacement notre temps afin de présenter un résultat satisfaisant au client. Grâce à l'utilisation de multiples outils d'organisation, nous avons pu répondre à cette attente avec succès.

➔ Adaptabilité

Effectuer un stage au MIC est une expérience unique. Les prototypes développés utilisent tous des technologies innovantes et parfois, l'utilisation de ces technologies émergentes peut provoquer certains problèmes inattendus. Il a donc fallu s'adapter et trouver des solutions rapidement à des problèmes d'envergure afin de mener à bien notre projet. De plus, les besoins du client peuvent évoluer rapidement, ce qui nécessite une adaptation constante.

b. Compétences techniques

- ➔ L'utilisation des services Azure.
- ➔ Le développement en Python.
- ➔ Le développement en C#.
- ➔ Le développement en Next.js.
- ➔ La création, l'entraînement et l'intégration à notre solution de modèles intelligents de détection d'objets.
- ➔ L'utilisation de la méthodologie Agile Scrum.

XI. Conclusion

Ces quinze semaines de stage passées au sein du MIC ont été pour moi une expérience épanouissante et formatrice. J'ai pu m'intégrer à un groupe de professionnels constamment engagés à travailler avec de nouvelles technologies. Ainsi, je me suis mise dans la peau d'une vraie professionnelle en rencontrant mon client régulièrement, en travaillant en équipe et en collaborant avec des professionnels externes pour répondre aux attentes du client.

À la fin de ce stage, je constate une nette amélioration de mes compétences, tant sur le plan technique que sur le plan relationnel. J'ai pu découvrir et perfectionner mes connaissances dans des domaines tels que Python, C# et Azure Custom Vision. Un autre aspect primordial de ce stage a été la mise en situation réelle, qui m'a permis de développer des réflexes professionnels essentiels lors des interactions avec un client, ainsi que de renforcer l'esprit d'équipe en travaillant en binôme sur le projet.

Au fil de mes études, je me questionnais sur l'impact que les développeurs peuvent avoir sur la société. Mon aspiration a toujours été de contribuer positivement à la vie à travers ma carrière professionnelle. C'est l'une des raisons pour lesquelles le sujet du projet CoucouAI m'a particulièrement touchée. J'ai ainsi découvert la contribution positive qu'un développeur pouvait avoir sur la préservation d'espèces animales menacées. L'enjeu n'était pas anodin, c'est pourquoi les attentes du client étaient élevées. Le développement de ce projet a exigé que je me dépasse, à la fois sur le plan technique que relationnel.

Au cours de ces semaines passées au MIC, j'ai trouvé ma place en tant que développeuse femme. J'ai été accueillie par une équipe bienveillante qui avait confiance en mes capacités et qui a mis l'accent sur l'égalité des traitements entre les genres. Je sors de cette aventure plus confiante et épanouie. Quand j'ai pris la décision de déposer ma candidature au MIC, j'avais conscience que cette aventure serait aussi un défi personnel. Je suis ravie d'avoir participé au développement de CoucouAI et d'avoir pu faire des rencontres exceptionnelles.

En conclusion, cette expérience a été exceptionnelle pour moi. J'ai développé de nombreuses compétences et trouvé ma voie dans un domaine qui me passionne. À la fin de ce stage, j'estime avoir développé une solution innovante qui simplifiera le travail des éleveurs et permettra une préservation plus efficace des coucous.

XII. Bibliographie

- 1 K-Force Computer Solutions. (2024, 17 janvier). *Le partenaire pour votre ICT, digital workplace et téléphonie | K-Force.* K-Force. Consulté le 8 avril 2024, à l'adresse <https://k-force.be/fr/>
- 2 ClearMedia. (2024, 7 mars). *ClearMedia - Découvrez les super pouvoirs de ClearMedia !* Consulté le 8 avril 2024, à l'adresse <https://www.clearmedia.be/fr/>
- 3 Fiche métier : Product Owner - Cegos. (s. d.). Consulté le 8 avril 2024, à l'adresse <https://www.cegos.fr/ressources/mag/fiches-metiers/les-metiers-de-la-gestion-de-projets/fiche-metier-product-owner>
- 4 Qu'est-ce qu'un développeur et quelles sont ses missions ? (s. d.). DeviensDev. Consulté le 8 avril 2024, à l'adresse <https://deviens.dev/metier-developpeur/cest-quoi-un-developpeur>
- 5 IBM documentation. (s. d.). Consulté le 8 avril 2024, à l'adresse <https://www.ibm.com/docs/fr/engineering-lifecycle-management-suite/lifecycle-management/7.0.1?topic=requirements-defining-use-cases>
- 6 Slack. (s. d.). Qu'est-ce que Slack ? Slack Help Center. Consulté le 8 avril 2024, à l'adresse <https://slack.com/intl/fr-fr/help/articles/115004071768-Qu%20est%20ce%20que%20Slack->
- 7 Qu'est-ce que Microsoft Teams ? (s. d.). Consulté le 8 avril 2024, à l'adresse <https://support.microsoft.com/fr-fr/office/vid%C3%A9o%20qu%20est%20ce%20que%20microsoft-teams-422bf3aa-9ae8-46f1-83a2-e65720e1a34d>
- 8 Visual Studio : IDE et éditeur de code pour les développeurs de logiciels et les équipes. (2024, 15 mars). Visual Studio. Consulté le 8 avril 2024, à l'adresse <https://visualstudio.microsoft.com/fr/>
- 9 Anandmeg. (2023, 27 octobre). Qu'est-ce que l'IDE Visual Studio ? Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/fr-fr/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- 10 Atlassian. (s. d.-b). Qu'est-ce que Git ? | Atlassian Git Tutorial. Consulté le 2 mai 2024, à l'adresse <https://www.atlassian.com/fr/git/tutorials/what-is-git>
- 11 Kinsta. (2023, 24 mai). Qu'est-ce que TypeScript ? Un guide complet - Kinsta®. Kinsta®. Consulté le 8 avril 2024, à l'adresse <https://kinsta.com/fr/base-de-connaissances/guide-complet-typescript/>
- 12 C sharp - Définition et Explications. (s. d.). Techno-Science.net. Consulté le 8 avril 2024, à l'adresse <https://www.techno-science.net/glossaire-definition/C-sharp.html>
- 13 RÉ ; Daction, L. (2020, 31 mars). Python ; : dé ; finition et utilisation de ce langage informatique. Consulté le 8 avril 2024, à l'adresse <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>
- 14 Qu'est-ce que .Net ? – .Net expliqué – AWS. (s. d.). Amazon Web Services, Inc. Consulté le 8 avril 2024, à l'adresse <https://aws.amazon.com/fr/what-is/net/>
- 15 Kinsta. (2022, 19 décembre). Qu'est-ce que Next.js ? Un regard sur le framework JavaScript populaire. Kinsta®. Consulté le 8 avril 2024, à l'adresse <https://kinsta.com/fr/base-de-connaissances/next-js/>
- 16 Introduction à Node.js | Makina Corpus. (s. d.). Makina Corpus. Consulté le 8 avril 2024, à l'adresse <https://makina-corpus.com/front-end/introduction-nodejs>
- 17 Dyouri, A. (s. d.). Créer une application Web avec Flask Python. Developpez.com. Consulté le 8 avril 2024, à l'adresse <https://python.developpez.com/tutoriel/intro-flask-python3/>
- 18 TensorFlow lite. (s. d.). TensorFlow. Consulté le 8 avril 2024, à l'adresse <https://www.tensorflow.org/lite/guide?hl=fr>

- 19 *Open Broadcaster Software | OBS.* (s. d.). Consulté le 12 mai 2024, à l'adresse <https://obsproject.com/fr>
- 20 Security, L. F. +. (2022, août 15). Nx Witness v5.0 est arrivé. *Stentor 6.1 Demo*. Consulté le 10 mai 2024, à l'adresse <https://www.lobeco.be/fr/pageid/newsitem-2022-07-28>
- 21 *Why NX Cloud ?* (s. d.). Nx. Consulté le 10 mai 2024, à l'adresse <https://nx.dev/ci/intro/why-nx-cloud>
- 22 15 *Intégration de MailKit avec le service de transmission de messages.* (s. d.). (C) Copyright 2024. Consulté le 10 mai 2024, à l'adresse <https://docs.oracle.com/fr-ca/iaas>
- 23 *What is cloud computing ? | Microsoft Azure.* (s. d.). Consulté le 9 mai 2024, à l'adresse <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing>
- 24 Ggailey. (2023, 24 mai). *Azure Functions Overview*. Microsoft Learn. Consulté le 9 mai 2024, à l'adresse <https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview?pivots=programming-language-csharp>
- 25 Akashdubey-Ms. (2023b, décembre 10). *Vue d'ensemble du compte de stockage - Azure Storage.* Microsoft Learn. Consulté le 10 mai 2024, à l'adresse <https://learn.microsoft.com/fr-fr/azure/storage/common/storage-account-overview>
- 26 Akashdubey-Ms. (2023, 29 mars). *Introduction au Stockage (d'objets) Blob - Azure Storage.* Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/fr-fr/azure/storage/blobs/storage-blobs-introduction>
- 27 Seanmcc-Msft. (2023c, juin 27). *Understanding the Table service data model (REST API) - Azure Storage.* Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/en-us/rest/api/storageservices/understanding-the-table-service-data-model>
- 28 PatrickFarley. (2024, 21 mars). *Qu'est-ce que Custom Vision ? - Azure AI services.* Microsoft Learn. Consulté le 8 avril 2024, à l'adresse <https://learn.microsoft.com/fr-fr/azure/ai-services/custom-vision-service/overview>
- 29 *Agile scrum methodology : what to know | Inflectra.* (s. d.). Consulté le 8 avril 2024, à l'adresse <https://www.inflectra.com/Solutions/Methodologies/Scrum.aspx>