Cahier des charges

# Contexte du projet

Le frelon asiatique (*Vespa Velutina*) a été introduit accidentellement en France en 2004 et se répand depuis en Europe. Cette espèce invasive, dont la principale source de nourriture est les abeilles, est combattue par les autorités suisses depuis qu’elle a été identifiée pour la première fois à Genève en 2020. Aujourd’hui elle a colonisé le pied du Jura jusque dans la région bâloise.

Les rencontres avec des frelons, même européens (*Vespa Crabro*), restent exceptionnelle pour la majorité de la population. De ce fait, il n’est pas toujours évident d’identifier correctement l’espèce exacte rencontrée. D’autant plus que le caractère et la taille impressionnante de l’insecte suscite aisément la peur.

En ajoutant que les espèces endémiques doivent être conservées si celles-ci ne présentent aucune menace directe, le travail des autorités peut vite être ralenti si ces dernières interviennent sur des fausses alertes remontées par des citoyens n’ayant aucune connaissance sur l’apparence de l’espèce à éradiquer.

C’est dans ce contexte que nous souhaitons proposer une solution mobile cross-plateforme permettant l’identification de l’espèce nuisible à l’aide de la caméra du téléphone. L’idée principale étant de pouvoir réaliser cette identification à l’aide d’un modèle d’intelligence artificielle (machine learning) embarqué directement sur le téléphone. Une fois l’espèce nuisible identifiée, il sera possible de participer à la localisation du nid en fournissant les images capturées ainsi que les coordonnées géographiques ou les trajectoires de vols.

# Cible

Cette application s’adresse à un large public que celui-ci soit familier avec la technologie ou non. Il s’adresse à tout possesseur de smartphone équipé d’une caméra peu importe le fabricant de ce dernier. Cette solution trouvera son utilité auprès de personnes vivant sur le plateau helvétique ainsi que dans le jura, là où l’espèce a été majoritairement observée. Elle permet à tout un chacun de participer activement dans la lutte contre le frelon asiatique en s’assurant de ne pas fournir d’informations erronées en marquant des espèces endémiques.

L’aspect communautaire en partageant les données de localisations, de trajectoire de vol ainsi que les images capturées seront utiles à toutes les autorités compétentes participant à la destruction des nids. Elles permettent également aux biologistes de mieux observer le territoire occupé par l’espèce ainsi que l’évolution de ce dernier.

Les apiculteurs ou toutes personnes aptes à capturer l’insecte de façon sécurisée pourra également introduire les trajectoires de vols de l’insecte une fois celui-ci relâché. Ces données étant d’autant plus précieuses puisqu’elles permettent une triangulation du nid.

# Identification des besoins

Comme il a été présenté plus haut, l’application se doit de proposer plusieurs fonctionnalités afin de couvrir les besoins énoncés selon le contexte, ainsi que des besoins non-fonctionnel devant s’adapter au public cible.

## Besoins fonctionnels

* **Accès à l’appareil photo** : L’application doit pouvoir accéder à la caméra du téléphone pour prendre des photos et utiliser les captures obtenues dans l’application.
* **Identification du frelon sur image** : Chaque image capturée doit être analysée et identifiera la présence ou non d’un objet définit en amont par l’application (dans notre cas, le frelon asiatique).
* **Insertion des coordonnées géographiques** : Les utilisateurs doivent disposer d’une solution simple pour indiquer les coordonnées géographiques du lieu de capture de l’image.
* **Tracés sur une carte :** L’application doit permettre à l’utilisateur de saisir des trajectoires de vol sur une carte.
* **Envoi de données à un serveur** : Si le résultat de l’identification s’avère être l’espèce recherchée, l’utilisateur pourra fournir les coordonnées géographiques ainsi que les images capturées à un serveur.

## Besoins non-fonctionnels

* **Cross-platform** : L’application se doit d’être fonctionnelle que celle-ci soit lancé depuis un appareil Android ou iOS. Ceci doit être transparent pour l’utilisateur.
* **Précision du modèle**: La détection doit être suffisamment précise et ce même sur des images de faibles résolutions ou avec un sujet de petite taille afin que les utilisateurs ne se mettent pas en danger lors de la capture d’images.
* **UI/UX**: L’application doit être simple d’utilisation afin que le plus large publique puisse l’utiliser sans difficultés particulières.

# Objectif

L’objectif principal de ce travail de bachelor est d’explorer la faisabilité d’embarquer un modèle deep learning de reconnaissance d’objets ou de classification d’images dans une solution cross-plateforme. Le défi étant que l’exécution du modèle ne doit pas être déléguée à un serveur, mais réalisée directement sur le smartphone.

Le contexte du projet étant très précis, il est possible que certaines limitations se présentent, notamment concernant la recherche d’un dataset d’images de frelons asiatiques suffisamment fourni et annoté correctement, nécessaire à la construction d’un modèle. Dans l’hypothèse où une telle limitation venait à se présenter, nous préfèrerons contourner les contraintes du contexte initial en choisissant d’autres datasets, sur d’autres cas d’utilisation, afin de mener à terme l’objectif principal.

# Fonctionnalités

Nous avons distingué les besoins en fonctionnalités principales, devant obligatoirement être inclues dans le livrable final, et en fonctionnalités optionnelles qui seront implémentées si le temps le permet.

## Fonctionnalités principales

* **Photographier** : L’application accèdera à la caméra du smartphone et permettra la capture d’images directement depuis l’application.
* **Analyser l’image** : Une fois la capture réalisée, l’utilisateur pourra la soumettre à une analyse par un modèle de deep learning embarqué qui indiquera si l’image contient bel et bien un frelon asiatique dans le cas où le contexte initial aura pu être respecté.

Si aucune limitation n’a été rencontrée, et que le temps alloué à l’intégration d’un modèle deep learning cross-plateforme aura pris moins de temps qu’initialement prévu, les fonctionnalités suivantes pourront être intégrées :

* **Se localiser** : Afin de faciliter l’entrée de la position où le sujet a été identifié, l’application mettra en place un service de géolocalisation afin de connaître sa position.
* **Envoi des coordonnées géographiques** : L’utilisateur pourra indiquer sur une carte la position relativement exacte de l’endroit où a été identifié le sujet en vue de transmettre les coordonnées géographiques à un serveur distant.
* **Envoi d’images** : Si l’utilisateur le souhaite, il pourra ajouter les images saisies en plus des coordonnées géographiques. Celle-ci devront être encodée pour pouvoir être transmise au serveur distant.

## Fonctionnalités optionnelles

* **Sauvegarde locale** : Un système de sauvegarde local permettra à l’utilisateur de reprendre une saisie interrompue de façon volontaire ou involontaire.
* **Tracé de trajectoire de vol** : Si l’utilisateur souhaite réaliser une triangulation du nid en ayant capturé des spécimens identifiés en amont, il pourra indiquer sur une carte la direction du vol du sujet à l’aide d’un glissement de doigt.
* **Envoi des trajectoires de vol** : De la même façon que l’utilisateur transmets des coordonnées géographiques, il pourra transmettre les trajectoires indiquées
* **Consultation de l’historique des données** : Le serveur distant stockera les données reçues et il sera possible de consulter ces données à tout moment.

# Planification et livrables

Ce travail sera réalisé dans un laps de temps d’environ 14 semaines pour un total de 450h de travail. Cette section présente une répartition globale de ce temps à disposition ainsi que les livrables associés. Chaque sous-section est annotée WP pour « Work Package » suivi d’un identifiant chiffré et d’une estimation globale de la durée.

## WP1 - Analyse et état de l’art (100h)

### Contenu

Dans un premier temps, il sera capital de trouver un dataset adapté au contexte. Comme il a été susmentionné, un tel jeu de données, d’une taille suffisante, pourrait s’avérer complexe à trouver. Le temps alloué ici inclura également la recherche d’une alternative en cas d’obstacles.

En parallèle, des recherches seront effectuées pour identifier les différentes architectures de réseaux de neurones adaptées à la problématique énoncée. Il s’agira de réaliser un état de l’art des algorithmes identifiés dans la recherche et des implémentations disponibles sur le marché.

Deuxièmement, il sera réalisé ici une recherche des solutions existantes répondant à la problématique. En particulier l’accent sera mis sur la recherche d’un logiciel ou d’une librairie permettant d’intégrer un modèle deep learning dans une solution cross-platform.

Dans une vision idéale, le modèle réalisé devrait pouvoir s’intégrer sur n’importe quel langage cross-platform. Du temps sera donc alloué à l’analyse de cette faisabilité.

Dans tous les cas, l’architecture global de l’application sera dressée tout en étant accompagnée de schémas graphiques illustrant son intégration avec une interface utilisateur.

Finalement, tout élément étudié devra se conclure par le choix réalisé en le motivant.

### Livrable

Le livrable sera constitué d’un rapport intermédiaire contenant des informations complètes sur un état de l’art sur la solution souhaitant être réalisée. Ainsi, il fera mention des différents solutions et applications existantes pour une problématique identique ou similaire.

Ce rapport mentionnera également les différents algorithmes et technologies existants dans le domaine de la reconnaissance d’image, et prendra parti sur la technologie s’adaptant le mieux à la situation énoncée. Après cela, le rendu parlera des différents modèles de deep learning existants dans la catégorie choisie en indiquant les avantages et inconvénients de chacun.

Il sera également fait mention des différentes technologies cross-platform existantes en mettant en lumière les avantages que celle-ci ont à offrir pour répondre au mieux à la problématique de ce travail.

La documentation mentionnera également comment l’algorithme choisi s’intègre dans une solution cross-platform en mettant en évidence les défis et limitations techniques que pourrait imposer une plateforme mobile. Entre autres, on tiendra compte des ressources disponibles comme la puissance de calcul, la mémoire et la consommation de batterie.

Divers schéma et diagrammes viendront appuyer les choix architecturaux fait afin de réaliser au mieux l’application souhaitée. Une maquette sera également documentée afin d’illustrer les visuels du programme.

## WP 2 - Réalisation du modèle (50h)

### Contenu

La réalisation d’un modèle deep learning comprend plusieurs étapes nécessaires. Le temps estimé ici servira donc à obtenir une architecture complète, fonctionnelle et démontrable.

Premièrement, il sera probablement nécessaire d’adapter le(s) jeu(x) de données sélectionné(s), en modifiant les libellés ou en procédant à de l’augmentation de donnée si ce(s) dernier(s) s’avère(nt) trop restreint(s).

Deuxièmement, il conviendra de modifier le modèle préconstruit afin de l’adapter aux besoins de ce travail. Cela comprend l’ajout ou la modification de couches dans le réseau de neurones (comme la couche de sortie), ainsi que la sélection d’hyperparamètres.

Ensuite, il faudra procéder à l’entraînement du modèle. Au préalable, il sera peut-être nécessaire de prétraiter les images en entrées afin de les uniformiser. Une fois l’entrainement terminé, il faudra valider le modèle au moyen d’un jeu de donnée n’ayant pas été utiliser à l’entraînement.

La réalisation d’un modèle deep learning est un processus itératif. Ainsi, l’entraînement et la validation devront probablement se répéter jusqu’à obtentions de résultats satisfaisants. La réalisation d’un modèle n’est toutefois pas l’objectif principal de ce projet, nous souhaitons pouvoir disposer d’un modèle “fonctionnel” permettant de démontrer son intégration dans un démonstrateur mobile.

### Livrable

Le rendu de cette section contiendra le modèle de reconnaissance de frelons dans une image ainsi que l’ensemble des fichiers, scripts et exécutables ayant servi à la réalisation de celui-ci. Plus précisément, le livrable sera composé des différents datasets utilisés, des techniques employées afin de procéder à l’augmentation de données, des divers scripts ayant servi à l’entraînement ainsi que de la validation et les scores du modèle.

## WP 3 - Réalisation de l’application (200h)

### Contenu

La majeure partie du contenu dans l’application a été explicité dans la section « Fonctionnalités ». Toutefois, précisons ici que la majeure partie du temps alloué à cette section sera dans le but de réaliser l’intégration du modèle dans des applications cross-platform.

Le temps de cette partie inclut les éventuels revirements d’implémentation qui seraient survenus à la suite d’obstacles infranchissables pour respecter les contraintes de temps.

### Livrable

Le livrable minimal sera l’application de démonstration réalisée contenant toutes les fonctionnalités principales décrites dans la section éponyme. En particulier le démonstrateur intégrera le modèle deep learning conçu dans le WP2. Si le temps le permet, ce délivrable pourra également être agrémenté d’une ou plusieurs fonctionnalités optionnelles. De plus, ce livrable devra être fonctionnel, peu importe la plateforme sur laquelle il est exécuté.

## WP4 - Tests et validations (50h)

### Contenu

Au fur et à mesure de développement, certaines sections critiques devront être testés de façon automatique et manuelle.

La reconnaissance d’objet ou la classification d’image réalisée directement sur un téléphone peut se heurter à des facteurs limitants comme l’architecture matériel ou la résolution de la caméra. De plus, le deep learning pouvant être relativement gourmand en consommation d’énergie, il sera nécessaire de réaliser des tests de performances en incluant des modèles et marques de smartphones différents (Android/iOS, différentes gammes de prix).

### Livrable

Le rapport final contiendra diverses informations concernant les tests de performances réalisés et indiquera les différents résultats obtenus.

## WP5 - Gestion du projet, documentations et présentation (50h)

### Contenu

Différents documents doivent être réaliser pendant ce temps. Les voici :

* Cahier des charges
* Rapport intermédiaire comprenant l’état de l’art des technologies disponibles ainsi que celles retenues
* Documentation de l’implémentation du modèle ainsi que celle de l’application
* Affiche et résumé publiable
* Préparation à la soutenance

### Livrable

Cette section sera réalisée en parallèle de toutes les autres et constituera le rapport final de ce travail de bachelor. Il parlera de chaque étape en mentionnant l’approche utilisée, les tests effectués, les problèmes rencontrés ainsi que les solutions exploitées pour les surmonter.

Chacune des sections qu’il contiendra sera agrémentée d’illustrations et de tableaux pour permettre aux lecteurs une compréhension la plus complète et limpide possible.

Un journal de bord sera également transmis afin de détailler le temps passé à chacune des tâches et de mieux comprendre où les avances et éventuels retards ont été pris.

# Organisation

L’organisation globale peut être consultée via le digramme de Gantt ci-dessous.

Une image contenant texte, ligne, Tracé, Parallèle

Description générée automatiquement

Figure 1 - Diagramme de Gantt de l'organisation générale du projet