Compte Rendu 11 : SAE 501

Durant cette semaine, dans le cadre de notre projet, plusieurs avancées significatives ont été réalisées. Voici un résumé des éléments déjà implémentés :

Nous avons réussi, avec Yolov8 et la dépendance Flutter Vision, à utiliser la caméra pour réaliser de la détection d'objets. Notre vue caméra peut aujourd'hui détecter des objets en temps réel, prendre des photos tout en affichant les résultats, mais également effectuer des reconnaissances sur des photos dans la galerie de l'utilisateur et afficher les résultats.

En plus de quelques modifications de l'esthétique de nos pages, nous avons ajouté des pages permettant de se connecter et de s'inscrire dans l'API. Actuellement, les formulaires existent et sont stylisés, mais ils ne font pas encore de requêtes vers notre API.

Concernant l'API, nous avons créé une API permettant de recevoir des images sous forme encodée en base64, de les télécharger et de les enregistrer sur le serveur. Cette API inclut également la génération d'un fichier ".txt" contenant des informations associées à chaque image.

Une route API nommée `/api/upload-image` a été mise en place pour recevoir des requêtes POST contenant une image encodée en base64 et ses métadonnées comme le tag associé à l'image et sa position sur l'image. L'image est décodée, puis enregistrée sur le serveur dans un répertoire dédié (`public/uploads/images/`).

Parallèlement, un fichier `.txt` est créé dans un répertoire « jumeau » qui est `public/uploads/labels/`. Ce fichier contient des informations telles que le tag associé à l'image et sa position sur l'image. Une confirmation est ensuite retournée à l'utilisateur avec un statut HTTP approprié, comme 200 pour un succès ou 400/500 en cas d'erreur.

L'objectif de la deuxième route API était de renvoyer l'historique des images envoyées par un utilisateur. Une route nommée `/api/user/{id}/history` a été créée pour accepter des requêtes GET.

Pour traiter la requête, l'utilisateur est identifié grâce à son `user_id`. Une recherche est effectuée dans la base de données. Chaque entrée dans l'historique inclut l'id de l'image, son chemin dans le serveur, la date et l'heure de l'envoie.

Ces deux API répondent aux besoins fonctionnels exprimés, tout en offrant une base solide pour la gestion des images et leur historique d'envoi.

Membres du groupe :
- Renaud Zell
- Thomas Ducret
- Julien Maaroufi
- Steven Lefebvre