

Compte rendu N°6 : SAE 5.01 – Développement Avancé

Sujet : Création d'une application de reconnaissance par IA d'objets du monde réel en temps réel

Thème : Matériel Scolaire (stylos, cahiers, règle, etc...)

I) Avancement (21/11/25) :

Cette semaine, le travail a porté sur la fonctionnalité de détection d'objets, ainsi que sur la mise à jour complète de l'infrastructure iOS/Flutter générée automatiquement par l'environnement.

La page de détection et la caméra ont été mises à jour pour pointer vers une nouvelle adresse IP locale (anciennement 192.168.1.186 et 10.11.123.248, désormais 172.20.10.9).

Cela a permis de rétablir la communication avec le serveur Flask pour les fonctionnalités de prédiction.

De nombreux fichiers générés automatiquement par Flutter ont été régénérés, notamment pour iOS (Podspec, configuration XCConfig, environnement d'exécution)

Le système de détection est fonctionnel, mais le code n'est pas encore totalement propre.

II) Partie Intelligence Artificielle (Lilian) :

Le travail IA de cette semaine c'est concentré sur la mise en place de la détection fonctionnelle, l'ajustement des endpoints réseau et la synchronisation entre l'application Flutter et le serveur Python.

Travaux réalisés :

- Mise à jour des endpoints réseau :
 - detect_page.dart : 192.168.1.186 → 172.20.10.9
 - camera_screen.dart : 10.11.123.248 → 172.20.10.9
 - ➔ Garanti que la prédiction est envoyée vers l'adresse correcte du serveur Flask
- Stabilisation de la communication Flutter/Flask :
 - Validation de l'envoi des images en JPEG (camera) et upload (gallery)
- Régénération automatique de l'environnement IOS :

- Ajout des fichiers générés par Flutter (Podspec, XCConfig, Script environnement Shell). Ces fichiers permettent de stabiliser le build IOS lors des prochaines compilations.

Ces actions ont permis de rendre la détection fonctionnelle, même si des refactoring restent nécessaires pour rendre le code plus propre et lisible.

III) Prochaines étapes :

- Finaliser le nettoyage du code Dart (refactor organisation + suppression des reliquats)
- Améliorer la gestion des images (optimisation du traitement + formats YUV/BGRA)
- Sécuriser le changement d'adresse IP pour éviter de modifier le code à chaque test
- Vérifier le build iOS complet et préparer les tests sur appareil
- Améliorer les messages d'erreur pour l'utilisateur
- Nettoyer les fichiers générés Flutter pour éviter les commits inutiles