

## **Compte rendu N°5 : 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 (14/10/25) :**

Cette semaine, le travail a porté sur la stabilisation finale du code Dart, la mise à jour de la configuration réseau et la validation du build iOS.

Après la correction des structures cassées dans le code Dart et la migration de la librairie image vers la version 4, nous avons pu compiler l'application sans aucune erreur et préparer l'environnement pour les tests sur appareil mobile.

Le serveur Flask fonctionne correctement avec le modèle YOLOv8 chargé, et la communication réseau a été ajustée suite au changement d'adresse IP.

Le build iOS a été effectué avec succès (flutter build ios --no-codesign) : 559 lignes, 0 erreurs Dart, 0 erreurs natives et 20 pods installés.

### **II) Partie Intelligence Artificielle (Lilian) :**

Cette semaine, l'accent a été mis sur l'optimisation du code et sur la librairie image.

Plusieurs ajustements ont été réalisés :

- Correction du code Dart : réécriture complète du code cassé (1047 lignes -> 559 lignes), résolution de toutes les erreurs Dart.
- Migration Image Library v4 : adaptation des appels API et encodage JPEG fonctionnel.
- Configuration réseau : mise à jour de l'adresse IP du serveur Flask (ancien : 192.168.1.186 :5001 -> nouveau 10.11.123.248:5001)
- Serveur Flask : modèle YOLOv8 chargé et opérationnel, serveur accessible sur la nouvelle IP, prêt à recevoir les requêtes http depuis l'application.

Ces étapes ont permis de résoudre les derniers blocages liés au code, à la librairie image et au réseau, préparant un environnement stable pour les tests live.

### III) **Prochaines étapes :**

- Implémenter l'envoi des images en temps réel et gérer la réponse du serveur
- Corriger la conversion d'images (support complet BGRA + YUV)
- Finaliser la détection dans GalleryScreen (upload -> prédiction -> affichage)
- Ajouter un affichage clair des objets détectés (overlay ou liste)
- Vérifier la communication réseau (IP, port, disponibilité du serveur)
- Améliorer les messages d'erreur pour l'utilisateur