Compte rendu N°2 : 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 (24/10/25) :

Durant cette nouvelle étape du projet, l'équipe a poursuivi le développement de l'application Flutter et la mise en place du module d'intelligence artificielle. L'objectif principal de cette semaine était de renforcer la partie interface utilisateur et d'amorcer la connexion entre Flutter et le back-end Python dédié à la détection d'objets.

II) Partie application (Said et Thomas) :

1) Conception et amélioration de l'interface utilisateur (UI)

Conception d'une interface claire, moderne et cohérente avec la thématique du projet.

- L'écran d'accueil présente un dégradé bleu-violet évoquant le monde de l'éducation et des nouvelles technologies.
- Trois boutons principaux sont proposés : « Détection en temps réel », « Analyser une image enregistrée » et « Voir l'historique ».
- L'aspect visuel a été soigné : boutons arrondis, centrés et fluides pour une meilleure expérience utilisateur.

2) Implémentation de la caméra

- Intégration de la bibliothèque camera de Flutter pour accéder au flux vidéo en temps réel.
- L'écran Camera_Screen permet d'activer la détection en direct, de changer de caméra (avant/arrière) et de capturer une photo avec un effet de flash simulé.
- Le flux caméra est désormais affiché en plein écran, sans bordures, et offre une expérience fluide.

3) Implémentation de la galerie

- Ajout de la page Gallery_Screen permettant de sélectionner une image locale via la bibliothèque image_picker.
- L'image choisie s'affiche dans une carte arrondie avec effet d'ombre.
- Un bouton « Analyser » a été ajouté : il affiche pour l'instant un message simulé (« Analyse simulée en cours... »), en attendant la liaison avec l'IA.

4) Implémentation de l'historique

- Création de la page History_Screen, qui affiche les images capturées sous forme de grille visuelle.
- L'utilisateur peut cliquer sur une image pour l'afficher en plein écran.
- À terme, cette page sera reliée à la base de données pour afficher automatiquement les captures enregistrées.

III) Partie Intelligence artificielle (Lilian) :

Prise en charge la partie IA du projet, consacrée à la détection d'objets via la caméra.

- Mise en place d'un module Python dans le dossier ml_model.
- Configuration d'un environnement virtuel et installation des bibliothèques principales : TensorFlow, Flask, OpenCV et NumPy.
- Après résolution de conflits Git et intégration du projet Flutter, étude de l'architecture afin de préparer la communication entre le front-end (Flutter) et le back-end (Flask).
- Début du développement d'une API Flask (app.py) permettant de :
    - Charger un modèle TensorFlow Lite (.tflite).
    - Capturer une image depuis la caméra.
    - Renvoyer la liste des objets détectés au client Flutter sous format JSON.

Le modèle .tflite n'a pas encore été généré. Les prochaines étapes concernent :

- L'entraînement du modèle via train_model.py.
- Sa conversion en format mobile (convert_to_tflite.py).
- Les premiers tests de détection en temps réel dans l'application Flutter.

IV) Prochaines étapes :

- Intégration du modèle TensorFlow Lite dans Flutter.
- Liaison complète entre le front-end et le back-end.
- Entraînement et conversion du modèle IA.
- Tests de reconnaissance en temps réel.
- Intégration de la base de données pour l'historique.