

Compte rendu N°10 : 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 (19/12/25) :

Cette semaine, nous avons continué la préparation des données pour l'entraînement du modèle.

Prendre des photos d'objets scolaires nous-même étant très long, nous avons cherché des plateformes permettant de trouver de nombreuses banques d'images d'objets scolaires, étant en plus de cela déjà annotées pour entraîner notre modèle. Nous avons également annoté les images que nous avons prises nous-même la semaine dernière. La branche « alex_branch » a également été fusionnée vers la branche « main », et les anciens fichiers relatifs au serveur Flask ont été supprimés. De plus, certaines corrections mineures ont été appliquées à l'application.

II) Partie Intelligence Artificielle (Alexandre, Said) :

Cette semaine, nos efforts se sont concentrés sur l'étape critique de la constitution du jeu de données (dataset). Nous avons identifié un goulot d'étranglement dans notre processus initial : la prise de photos manuelle d'objets scolaires s'est avérée trop chronophage et ne garantissait pas une diversité suffisante des conditions d'éclairage et d'arrière-plan.

Pour pallier ce problème et accélérer le processus, nous avons pivoté vers l'agrégation de données existantes :

- Nous avons exploré des plateformes spécialisées et sélectionné Roboflow Universe, qui offre des banques d'images d'objets scolaires open-source.
- L'intégration de ces datasets externes nous permet de bénéficier d'images déjà annotées (bounding boxes), ce qui représente un gain de temps considérable pour la phase d'apprentissage supervisé.

En parallèle, nous avons finalisé le traitement de nos propres données :

- Les images capturées la semaine précédente ont été intégralement labellisées pour être compatibles avec le format attendu par notre modèle.
- Nous travaillons désormais à l'homogénéisation des images collectées (internes) et des images sourcées (externes) pour constituer un dataset robuste et varié.

III) Partie Développement et Architecture (Alexandre, Thomas) :

En dehors des aspects liés au modèle, nous avons travaillé sur la maintenance et l'évolution de l'architecture technique du projet. L'objectif était d'assainir la base de code et de corriger certains problèmes d'affichage.

- La branche de fonctionnalité « alex_branch » a été fusionnée avec succès vers la branche de production « main ». Cette fusion marque l'intégration des dernières fonctionnalités développées isolément au cœur de l'application stable. Cela a d'ailleurs permis de corriger les défauts de la précédente méthode d'affichage du cadre de détection d'objets de par la fusion entre la méthode de la branche « alex_branch » et la méthode de la branche « main ».
- Afin de réduire la dette technique, nous avons procédé à la suppression des fichiers obsolètes liés à l'ancien serveur Flask. Cette action allège le projet et évite toute confusion avec la nouvelle architecture backend mise en place.
- Plusieurs correctifs mineurs ont été déployés sur l'application. Ces modifications visent principalement à améliorer la stabilité globale et à corriger des bugs d'affichage remontés lors des tests précédents, mais aussi à supprimer le bandeau « DEBUG » dans le coin supérieur droit de l'écran. Des correctifs sont prévus sur la méthode paint de la classe ObjectPainter du fichier « camera_screen.dart ».

IV) Prochaines étapes :

- Apporter les correctifs nécessaires à la méthode paint de la classe ObjectPainter (fichier camera_screen.dart).
- Vérifier le build iOS
- Reprendre les anciens fichiers Dart, concernant l'interface utilisateur, provenant du main, ils sont plus complets et ont vu un apport de diverses améliorations
- Entraîner le modèle d'IA
- Rendre fonctionnelle la fonctionnalité de reconnaissance d'images provenant de la galerie
- Implémenter l'historique des détections
- Améliorer les messages d'erreur pour l'utilisateur