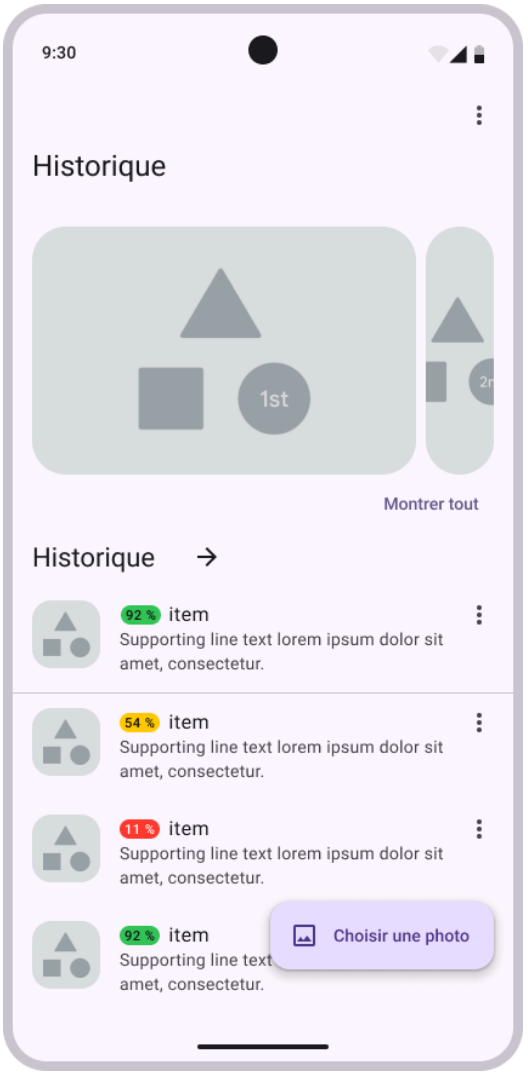
Rapport 18/10/2024

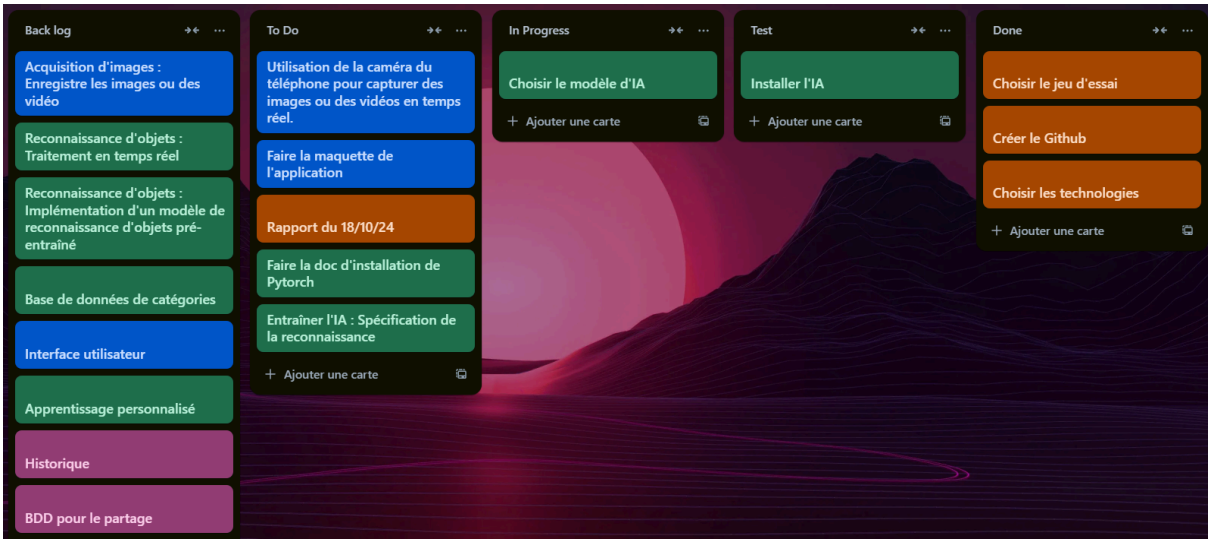
1. Début de la maquette

- La première version visuelle/schématique du projet à été faite, à savoir l'historique des scans réalisés
- Image :



2. Réalisation du Trello

- Le tableau Trello a été créé avec les prochaines tâches à réaliser.
- Image

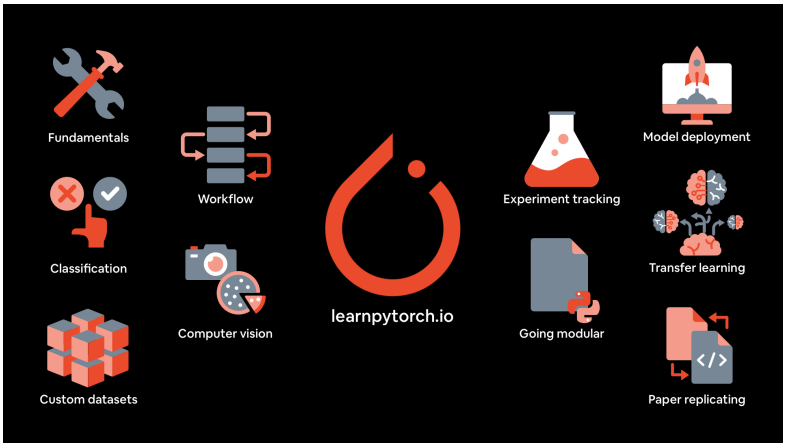


3. Organisation de la répartition des tâches

- Les tâches ont été réparties entre les membres de l'équipe. (Peut-être modifié à l'avenir).
- Front, intégration : Illia, Yoni, Jonathan
- IA : Rodney, Dorian

4. Installation de PyTorch

- PyTorch a été installé permettant de doter l'ordinateur d'une vision, à savoir l'accès à la caméra pour effectuer les scans.



5. Installation du framework de l'IA

- Le framework d'IA lié à PyTorch a été installé.

6. Test du modèle ConvNet

- Le modèle **ConvNet** inclus dans le tutoriel PyTorch a été testé.

7. Choix du modèle EfficientDet

Le modèle d'IA **EfficientDet** a été choisi pour la suite du projet.

En effet, en réalisant un comparatif, nous avons trouvé que ce modèle est adapté pour un environnement mobile, avec des ressources limitées. Il représente un bon compromis entre performance et précision de reconnaissance.

Model	Type	Pros	Cons
YOLO	One-stage	• Very fast processing speed • Simple architecture • Good for real-time applications	• Can be less accurate than two-stage detectors • May struggle with small objects
EfficientDet	One-stage	• Excellent balance of accuracy and speed • Efficient backbone network • Good for mobile and resource-constrained environments	• May still lag slightly behind top two-stage detectors in accuracy • Can struggle with very small objects
RetinaNet	One-stage	• High accuracy • Addresses class imbalance well	• Can be slower than YOLO • More complex architecture
Faster R-CNN	Two-stage	• High accuracy • Good for detecting small objects	• Slower than one-stage detectors • More complex training process
Mask R-CNN	Two-stage	• High accuracy for instance segmentation • Provides pixel-level masks for objects	• Even slower than Faster R-CNN • More computationally expensive

(Source : https://dagshub.com/blog/best-object-detection-models/)

Il convient maintenant de trouver un repo, ayant pré-entraîné ce modèle, afin de l'utiliser pour notre thématique, les boissons.