

Présentation Finale – Détecteur de Plantes Malades

1. Objectifs Initiaux du Projet - Atteindre une précision 85% pour la détection des maladies. - Développer un modèle performant basé sur un CNN (ResNet9). - Mettre en place une API capable d'effectuer un diagnostic en temps réel. - Déployer une solution portable grâce à Docker.
2. Résultats Obtenus - Précision du modèle supérieure à 85%. - API fonctionnelle avec endpoint /predict permettant de diagnostiquer à partir d'une image. - Conteneur Docker stable et portable. - Solution complète : modèle API déploiement.
3. Modèle IA – Résumé - Architecture : ResNet9, adaptée à la classification d'images. - Entraîné sur 38 classes issues du dataset PlantVillage. - Production de prédictions rapides et fiables.
4. API – Résumé - Endpoint /predict pour l'analyse d'image. - Chargement du modèle au démarrage pour des performances optimales. - Sortie structurée (JSON) contenant classe et probabilité.
5. Déploiement & Dockerisation - Image Docker intégrant Python 3.12, FastAPI, PyTorch CPU. - Déploiement reproductible sur n'importe quel serveur. - Commandes simples : build run.
6. Objectifs vs Résultats - Précision cible : Atteinte. - Rapidité du diagnostic : API très réactive. - Portabilité : Docker opérationnel. - Utilisabilité : Workflow complet prêt pour démonstration.
7. Limites & Pistes d'Amélioration - Manque d'interprétabilité : ajout futur de Grad-CAM. - Amélioration de la calibration des probabilités. - Besoin de données réelles du terrain pour fiabiliser. - Création future d'une interface mobile.
8. Conclusion Le projet couvre toute la chaîne IA : entraînement du modèle, construction d'une API, déploiement et exécution. La solution est fonctionnelle, portable et constitue une base solide pour une application destinée aux agriculteurs.