

Le pipeline de la détection de maladies

La mise en œuvre technique se fera en 4 étapes principales, en utilisant Google Earth Engine (GEE) et des outils de data science classiques.

1. Préparation de l'environnement de travail

- **Google Earth Engine (GEE)** : C'est la plateforme centrale. C'est là que nous accéderons à toutes les images satellites (Sentinel-2, par exemple) et où nous utiliserons le modèle **AlphaEarth** pour générer les vecteurs. Nous pouvons y travailler soit directement dans l'éditeur de code (qui utilise JavaScript) ou via son API avec **Python** (ce qui est idéal pour la science des données).
- **Notre environnement de développement** : Un ordinateur avec Python et les librairies classiques de data science installées (Pandas, Scikit-learn, etc.). C'est ici que nous ferons le "gros" du travail d'analyse et d'entraînement du modèle.

2. Fusion des données (Le liant du projet)

Cette étape est cruciale : il s'agit de fusionner nos données de terrain (le fichier CSV) avec les données satellites d'AlphaEarth.

1. **Extraction des vecteurs AlphaEarth** : Dans GEE, nous écrivons un script pour interroger les images satellites. Pour chaque point GPS de notre fichier CSV, nous demandons au modèle AlphaEarth de nous fournir les vecteurs de 64 dimensions correspondant à cette localisation et à cette date précise.
2. **Création du jeu de données final** : nous exportons ces vecteurs hors de GEE (par exemple, dans un fichier CSV ou JSON). Chaque ligne du fichier contiendra les 64 dimensions du vecteur AlphaEarth et sera étiquetée avec la colonne `Stade_Maladie` de votre fichier de vérité terrain. Notre fichier final aura le format suivant :

| Dimension 1 | Dimension 2 | ... | Dimension 64 | Stade_Maladie |
|-------------|-------------|-----|--------------|-----------------------|
| 0.25 | -1.5 | ... | 0.98 | 2 (Maladie débutante) |
| -0.12 | 0.78 | ... | -0.45 | 0 (Sain) |

3. Entraînement du modèle d'IA

Nous utiliserons ce jeu de données pour entraîner notre modèle.

1. **Préparation** : nous divisons le jeu de données en deux : une partie pour l'entraînement (par exemple, 80 %) et une partie pour le test (20 %).
2. **Choix du modèle** : Pour une première preuve de concept, un modèle simple comme un **Random Forest Classifier** est un excellent choix. Il est puissant, rapide et ses résultats sont faciles à interpréter. Un modèle plus complexe comme un réseau neuronal pourrait être envisagé plus tard.

3. **Entraînement** : Le modèle est nourri avec nos données d'entraînement. Il va apprendre à reconnaître les schémas dans les 64 dimensions des vecteurs qui correspondent aux étiquettes "sain" ou "malade".
4. **Évaluation** : Nous testons la performance du modèle sur la partie "test" des données. Nous vérifions son taux de précision, sa capacité à détecter les maladies sans faire de fausses alertes, et sa performance générale.

4. Déploiement et visualisation

Une fois le modèle entraîné, nous pouvons l'utiliser pour faire des prédictions sur de nouveaux champs.

1. **Inférence** : Nous repassons par GEE pour obtenir les vecteurs AlphaEarth des parcelles que nous voulons surveiller, sans aucune donnée de vérité terrain cette fois.
2. **Prédiction** : Notre modèle IA analyse ces nouveaux vecteurs et attribue une probabilité de maladie à chaque pixel de 10x10 mètres.
3. **Visualisation** : Nous utilisons ces scores de probabilité pour générer la **carte thermique** dont nous avons parlé. Les zones à forte probabilité seront affichées en rouge, et les zones à faible probabilité en vert. Cette carte sera notre produit final.

Le cœur du projet est là : la capacité à faire le lien entre le monde réel (notre agronome sur le terrain) et le monde numérique (les vecteurs d'AlphaEarth). Si nous réussissons cette fusion, la suite est une application standard des méthodes de science des données.