Cahier des charges technique

*Projet d’intégration au site internet existant d’un système de reconnaissance d’images*

Ce document détaille les spécifications techniques du projet de reconnaissance d'images pour le site iNaturalist.org.

L'objectif principal de ce projet est de permettre aux utilisateurs de charger des photos de la vie sauvage et d'obtenir automatiquement des propositions d'espèces grâce à l'utilisation de l'intelligence artificielle.

# Spécifications techniques

## Exigence matérielle

Pour une question d’harmonisation des outils avec l’existant, la poursuite de l’utilisation d’AWS est préconisé.  
Une instance AWS type EC2 avec au minimum :  
- CPU : **4 cores**- RAM : **16 GB**  
- Stockage : **Amazon EBS, Elastic Block Store** (s’adapte selon l’usage, sachant que dans ce projet, l’image est supprimée en cas de non enregistrement de l’observation, et sinon elle sera stockée avec le service existant AMAZON S3)

## Exigence logicielle

*- Langage de programmation pour l’ IA* ***:* Python 3.11**  
*- Bibliothèques ou frameworks :*  
 **keras, tensorflow, pytorch, scikit-learn, numpy, plotly, seaborn et matplotlib**  
*- Langage pour le site web :* **HTML, CSS, JS**  
*- Autres technologies :* **Docker, Flask**  
*-* ***Prise en charge des formats d'image****:* JPEG, JPG, PNG, GIF, WEBP, BMP, TIFF, RAW, HEIC, TGA.

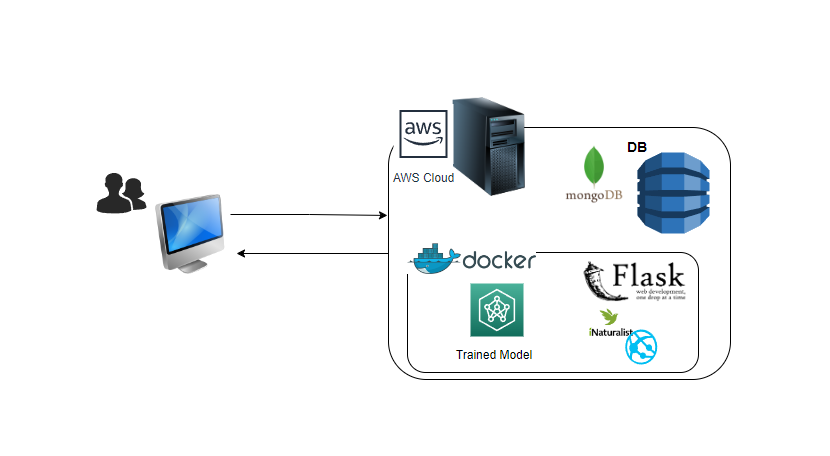
## Exigence en matière de performance et de précision de la reconnaissance d’images

Atteindre une précision minimale de 95% pour l'algorithme de reconnaissance.

# Base de Données

Nous poursuivons l’utilisation actuelle de MongoDB Atlas sous AWS comme méthode de stockage et de gestion des données.

# Architecture du Système



# Développement du modèle de reconnaissance

Afin de développer ce projet dans les meilleurs délais et avec les meilleurs résultats, nous **utiliserons un modèle pré entraîné orienté reconnaissance d’images (EfficientNet)**. Nous viendrons ajouter à ce modèle notre architecture de finalisation uniquement.

En effet, cette approche permet d'obtenir une meilleure précision avec moins de paramètres et de ressources par rapport à une architecture de réseau neuronal convolutif traditionnelle.

Nous testerons **plusieurs algorithmes et plusieurs hyperparamètres** afin d’établir un classement grâce à un score global. Ce score global prendra en compte 2 catégories :  
• l’efficacité « technique » du modèle (pondération 4/6) : loss, accuracy, precison, recall  
 Plus la valeur est élevée, plus le score est haut.  
• l’efficacité « énergétique » (pondération 2/6) :  
 Plus le temps de calcul est court, plus le score est élevé.

# Respect de la vie privée et sécurité des données

Tout comme le site internet, l’intégration de ce projet respectera le RGPD.  
Il sera demandé l’accord à l’utilisateur pour enregistrer de façon anonyme certaines informations techniques afin de mesurer des statistiques d’utilisation : la mesure de fréquentation (horodatage), le système d’exploitation, le navigateur, le pays d’origine, la latence, le temps de réponse.

Concernant la sécurité des données, nous continuerons d’utiliser les protocoles existants :

• Chiffrement des communications  
• Authentification des utilisateurs (uniquement pour l’enregistrement de l’observation)  
• Accès limité à l’API aux utilisateurs autorisés  
• Tous les protocoles de sécurité liés à AWS et MongoDB Atlas.

# Performances

• Temps de réponse attendu pour les requêtes d'identification : < 2 sec

• Scalabilité :  
- Utilisation d’équilibrage de charge et d’auto-scaling pour gérer les pics de trafic (inclus avec AWS)  
- Utilisation de l’architecture distribuée déjà présente (AWS et MongoDB Atlas) pour assurer une  
 haute disponibilité.

Optimisation du traitement d'image

# Tests et validation et déploiement

Afin de pouvoir tester le projet, nous développerons un « proof of concept » sous la forme d’une page web avec les fonctionnalités de prédiction.

Critères pour valider que le système répond aux exigences

Plan de Test

* Tests unitaires, d’intégration et de système
* Tests de robustesse face à des données erronées ou manquantes
* Tests d'intégration
* Tests de performance

# Documentation

Documentation remise :

Documentation du code

Guide d'utilisation pour les utilisateurs finaux

Guide de maintenance