

Projet 1: Interface Web pour la Classification d'Images avec Hyperparamètres Personnalisés

Objectif :

Ce projet combine à la fois le développement web dynamique, la gestion de la base de données, et l'intégration d'un système externe de traitement des images, offrant ainsi une expérience complète et enrichissante.

Description :

Ce projet consiste à développer une interface web qui permet à un utilisateur de sélectionner un répertoire d'images, de spécifier les hyperparamètres pour un système de classification ou de reconnaissance d'images, et d'exécuter le traitement. Les données saisies seront enregistrées dans une base de données, et le traitement des images sera effectué via un système appelé par PHP. L'interface affichera les résultats de la classification.

Fonctionnalités principales :

1. Sélection du répertoire d'images :

- L'utilisateur peut naviguer et sélectionner un répertoire contenant les images à traiter.
- Option d'affichage des images sélectionnées sous forme de miniatures.

2. Saisie des hyperparamètres :

- Interface pour saisir des hyperparamètres (par exemple : taux d'apprentissage, nombre d'époques, taille du lot, patience, monitor, optimiser, model_name, activation_function, validation_split, test_split, etc.).
 - ✓ Taux d'apprentissage [10^{-4} , 1]
 - ✓ Nombre d'époques : [10, 50]
 - ✓ Patience : [3, 5, 7]
 - ✓ Monitor : [val_loss, val_accuracy]
 - ✓ Optimiser : [Adam, SGD]
 - ✓ model_name : ...
 - ✓ activation_function : Sigmoid, ReLU, Tanh, Softmax
 - ✓ validation_split : [0.1, 0.2]
 - ✓ test_split : [0.1, 0.2]
- Validation des entrées pour s'assurer que les valeurs saisies sont correctes.

3. Enregistrement dans la base de données :

- Les informations saisies par l'utilisateur (chemin du répertoire d'images, hyperparamètres, etc.) sont enregistrées dans une base de données MySQL.
- Historique des traitements précédents pour permettre de revoir ou de rejouer des configurations passées.

4. Appel du système de traitement des images :

- Une fois les hyperparamètres saisis et validés, le système appelle un script de traitement (par exemple, un script Python) via PHP.
- Gestion de l'état du traitement (en cours, terminé, erreurs éventuelles).

5. Affichage des résultats :

- Les résultats de la classification ou de la reconnaissance sont affichés dans l'interface (par exemple : prédictions de classe pour chaque image, graphiques de performance).
- Option de télécharger les résultats ou de les enregistrer pour une consultation ultérieure.

6. Gestion des utilisateurs (optionnel) :

- Système d'authentification pour permettre aux utilisateurs de sauvegarder leurs configurations et résultats.

Architecture orientée objet :

- **Classe ImageDirectory** : Gère la sélection et l'affichage des images dans un répertoire.
- **Classe Hyperparameters** : Représente et valide les hyperparamètres saisis par l'utilisateur.
- **Classe Database** : Gère l'enregistrement des données dans la base de données et la récupération des configurations passées.
- **Classe ImageProcessing** : Interface avec le script externe (par exemple, Python) pour exécuter le traitement des images.
- **Classe ResultDisplay** : Gère l'affichage des résultats dans l'interface utilisateur.

Technologies utilisées :

- **PHP orienté objet** pour la logique backend.
- **MySQL** pour la base de données.
- **HTML/CSS/JavaScript** pour l'interface utilisateur.
- **Python** (ou un autre langage) pour le script de traitement des images, appelé depuis PHP.
- **AJAX** pour les appels asynchrones entre l'interface et le backend.
- **Bootstrap** pour le design responsif.