Rapport - Deep Learning
Projet de reconnaissance d'images pour identifier le pays correspondant à un lieu

Benzemma Lina / Chemmama Sharon / Lebreton Louis
Master 2 MoSEF Data Science – Université Paris 1 Panthéon-Sorbonne

1 Contexte et Objectifs du Projet

1.1 Contexte

Le domaine de la reconnaissance d'images a connu des avancées considérables ces dernières années grâce à l'essor du deep learning. Cette évolution ouvre la voie à des applications de plus en plus sophistiquées, notamment dans le domaine de la géolocalisation. Notre projet s'inscrit dans cette dynamique en proposant une solution capable d'identifier le pays d'origine d'un lieu à partir d'une simple photographie.

Cette problématique présente plusieurs défis techniques intéressants. Contrairement à la reconnaissance d'objets courants, l'identification du pays associé à un lieu implique l'apprentissage de caractéristiques visuelles spécifiques à chaque pays : styles architecturaux, paysages naturels caractéristiques, monuments emblématiques.

1.2 Objectifs

L'objectif principal de ce projet est de développer un modèle de deep learning capable de déterminer le pays d'origine d'une image représentant un lieu caractéristique. Plus précisément, notre modèle vise à :

- Analyser une image d'entrée représentant un site, un monument ou un paysage
- Identifier les caractéristiques visuelles distinctives permettant de déterminer son origine géographique
- Produire en sortie une prédiction du pays associé à l'image, accompagnée de probabilités pour les différentes classes

Cette solution pourrait trouver des applications dans des domaines variés comme le tourisme ou encore la recherche d'images (organisation automatique de collections photographiques par pays).

2 Données utilisées

Pour ce projet, nous avons exploité le Google Landmarks Dataset v2 (GLDv2), l'une des plus vastes collections d'images de points d'intérêt disponibles publiquement. Ce dataset a été initialement créé pour des tâches de reconnaissance et de recherche de landmarks et contient environ 5 millions d'images associées à plus de 200 000 landmarks à travers le monde.

Le Google Landmarks Dataset v2 présente plusieurs caractéristiques qui le rendent particulièrement adapté à notre projet :

- Grande diversité géographique : Les images proviennent de nombreux pays et régions du monde, offrant une couverture globale diversifiée.
- Variété des types de landmarks : Le dataset inclut aussi bien des monuments construits par l'homme (bâtiments historiques, ponts, statues) que des points d'intérêt naturels (montagnes, lacs, formations géologiques).
- Métadonnées géographiques : Chaque image est associée à des informations sur sa localisation, incluant notamment le pays d'origine.
- Images réelles : Les photographies proviennent de sources variées et représentent des conditions réelles (différents angles, conditions d'éclairage, saisons), ce qui favorise la robustesse du modèle.

3 Preprocessing des données

Le projet repose sur une chaîne de prétraitement robuste qui transforme des images brutes en données exploitables par notre modèle de deep learning. Cette étape est cruciale car elle conditionne directement la qualité de l'apprentissage et les performances du modèle final.

La méthodologie de prétraitement que nous avons adoptée comprend plusieurs phases clés :
Jointure : Nous avons utilisé le dataset Google Landmarks v2, en combinant deux sources d'information : les images de points d'intérêt (landmarks) et leurs métadonnées géographiques. Cette fusion est réalisée par une jointure entre les tables d'images et de localisation, permettant d'associer chaque image à son pays d'origine. Cette étape a réduit notre ensemble de données initial de plus de 4 millions d'images à environ 1,2 million d'images possédant des métadonnées de localisation fiables.

Resizing : Les images sont redimensionnées à une taille standard de 224×224 pixels, ce qui présente plusieurs avantages :

- Uniformisation des dimensions pour le traitement par lots (batching)
- Réduction de la mémoire requise tout en préservant les caractéristiques visuelles essentielles
- Compatibilité avec les architectures de réseaux de neurones conventionnelles pré-entraînées

Normalisation : Les valeurs des pixels sont normalisées en utilisant les moyennes et écarts-types du dataset ImageNet. Cette normalisation est effectuée car elle :

- Accélère la convergence lors de l'entraînement
- Réduit les problèmes liés aux différences d'échelle entre les caractéristiques
- Facilite le transfert learning en alignant notre distribution de données sur celle des modèles pré-entraînés

Ensuite, nous avons transformé les images en tenseurs PyTorch.

Cette étape de transformation et leur organisation en TensorDataset présente des avantages significatifs par rapport à d'autres approches de prétraitement :

- Efficacité computationnelle : Les tenseurs PyTorch sont optimisés pour les calculs vectoriels et matriciels sur GPU, ce qui accélère considérablement les opérations de prétraitement à grande échelle. La conversion préalable des images en tenseurs évite les opérations répétitives de chargement et transformation pendant l'entraînement.
- Gestion de la mémoire : En utilisant des tenseurs, nous pouvons contrôler précisément le type et la précision des données (ex. float32), optimisant ainsi l'utilisation de la mémoire sans compromettre la qualité de l'information visuelle nécessaire aux modèles de deep learning.

- Cohérence du pipeline : L'utilisation d'un format unifié (tenseurs) tout au long du processus, du prétraitement à l'inférence, élimine les problèmes d'incompatibilité et simplifie le déploiement du modèle.

Enfin, nous avons chargé les données via DataLoader. Passer par cette méthode plutôt que par des méthodes traditionnelles offre plusieurs avantages critiques pour notre projet :

- Traitement par lots (batching) : Le DataLoader organise automatiquement les données en lots de taille configurable, permettant d'optimiser l'utilisation du GPU et d'accélérer l'entraînement. Dans notre implémentation, nous utilisons une taille de lot de 32 images, offrant un bon équilibre entre vitesse d'entraînement et précision des gradients.
- Parallélisation : Le DataLoader peut charger les données en utilisant plusieurs workers en parallèle, réduisant considérablement les temps d'attente ce qui est particulièrement important lorsque l'on travaille avec de grandes quantités d'images.
- Mélange aléatoire (shuffling) : La capacité à mélanger automatiquement les données entre les époques améliore la généralisation du modèle en évitant les biais liés à l'ordre des exemples.
- Intégration fluide : Le DataLoader s'intègre parfaitement avec le reste de l'écosystème PyTorch.

4 Modèles

Nous avons comparé plusieurs architectures de réseaux de neurones convolutifs et de transformers pré-entraînés sur ImageNet, dans une approche de transfer learning. Pour chaque modèle, seule la dernière couche a été remplacée afin d'adapter le réseau à notre tâche de classification de pays, tout en conservant les poids pré-entraînés sur les couches précédentes.

Les architectures évaluées incluent ResNet50, reconnu pour sa robustesse grâce aux connexions résiduelles, EfficientNet-B3, qui offre un bon compromis entre précision et coût, ConvNeXt-Tiny, un CNN modernisé inspiré des Transformers, Swin Transformer, basé sur une attention glissante à l'échelle locale, DenseNet-201, dont les connexions denses favorisent le partage d'information, et enfin Vision Transformer (ViT), qui traite les images comme des séquences de patches via l'auto-attention.

4.1 Modèle final : EfficientNet-B0 et évaluation

Parmi les différentes architectures testées, nous avons retenu **EfficientNet-B0** comme base de notre modèle final. Ce choix repose sur son excellent compromis entre performance et légèreté : bien que relativement compact, il est capable de capturer des motifs visuels complexes présents dans les paysages, monuments et scènes urbaines.

Nous avons conservé les poids pré-entraînés sur ImageNet afin de tirer parti de connaissances visuelles génériques déjà acquises (formes, textures, structures), et n'avons modifié que la couche finale pour l'adapter à notre tâche de classification multi-classe. Cette stratégie de *transfer learning* permet non seulement de réduire le temps d'apprentissage, mais aussi d'améliorer la généralisation, notamment pour les classes peu représentées.

Afin de renforcer la robustesse du modèle et de limiter le surapprentissage, nous avons intégré une procédure d'*early stopping*, interrompant automatiquement l'entraînement lorsque les performances sur le jeu de validation cessent de s'améliorer. Ce mécanisme a permis de conserver un modèle stable, en évitant une dégradation de la capacité de généralisation.

Le modèle a atteint une précision de **61 % sur les données d'entraînement**, contre environ **34 % sur le jeu de validation**. Cet écart important reflète un phénomène de **surapprentissage** : bien que le modèle apprenne efficacement les exemples vus, il peine à généraliser à de nouvelles

images. Cela souligne la nécessité d'ajuster le pipeline, notamment via une meilleure gestion des classes rares ou par des techniques d'augmentation de données plus poussées.

5 Piste exploré : traitement avancé et gestion des classes rares

5.1 Modifications apportées par rapport à B0

Le backbone EfficientNet-B0 a été remplacé par EfficientNet-B2, en conservant les poids pré-entraînés sur ImageNet.

- Les premiers blocs du réseau ont été gelés (jusqu'au 4e) pour limiter le surapprentissage et accélérer la convergence.
- Un scheduler ReduceLROnPlateau a été ajouté pour ajuster dynamiquement le taux d'apprentissage selon la performance sur le set de validation.

Avec ce nouveau modèle, nous avons atteint 70% d'accuracy sur l'entraînement après seulement 8 époques. Cependant, la précision en validation est restée stable autour de 34%, avec une perte qui recommence à augmenter dès la 4e époque. Ces résultats témoignent d'un surapprentissage rapide, en partie lié à l'absence d'augmentation des données ou de traitement spécifique des classes rares.

Cette observation nous a conduit à améliorer notre pipeline d'apprentissage en nous concentrant sur la gestion du déséquilibre entre les classes.

5.2 Prétraitement différencié selon la fréquence des classes

Après analyse de la répartition des pays dans le dataset, nous avons défini comme classes rares celles comptant moins de 1000 images. Pour mieux les prendre en compte, nous avons mis en place un prétraitement différencié :

- Standard Transform (inchangé pour les classes fréquentes) : redimensionnement à 160×160, normalisation selon ImageNet, et conversion en tenseurs.
- Rare Transform : même traitement de base, mais enrichi d'augmentations de données (recadrage aléatoire, rotation, flipping horizontal, jittering) pour générer une plus grande diversité visuelle.

6 Interface TrioVision : une plateforme interactive de géolocalisation par l'image

Pour rendre notre système accessible et démontrer son utilité pratique, nous avons développé une interface interactive utilisant Streamlit.

La plateforme **TrioVision** repose sur une interface développée avec **Streamlit**, conçue pour offrir une expérience utilisateur fluide, moderne et interactive. Elle permet de visualiser les prédictions d'un modèle de deep learning appliqué à la géolocalisation automatique d'images.

6.1 Fonctionnalités principales

- **Chargement dynamique du modèle** : import d'un modèle EfficientNet pré-entraîné, capable de prédire les 5 à 15 pays les plus probables associés à une image.
- **Téléversement d'image** : interface simple et intuitive permettant à l'utilisateur d'importer sa propre image.

- **Prédiction du pays** : classification directe avec affichage des probabilités, permettant d'interpréter la confiance du modèle.

6.2 Carte interactive

- **Affichage de la position géographique** du pays prédit.
- **Marqueurs animés et regroupement (clustering)** des prédictions alternatives.
- **Couches personnalisables** (vue satellite, terrain, océan...) avec outils intégrés : dessin, zoom, plein écran.

6.3 Visualisations avancées

- **Diagramme radar** pour comparer la confiance entre plusieurs pays.
- **Graphique à barres** représentant la distribution des probabilités.
- **Heatmap de similarité** entre les prédictions.
- **Jauges de confiance par pays**, pour une lecture rapide et visuelle.

6.4 Interface et design

- **Thème CSS personnalisé**, inspiré des interfaces web modernes (typographie Poppins, palettes harmonisées, ombrages dynamiques).
- **Animations Lottie** pour enrichir l'interactivité (étapes de chargement, analyse en cours, succès de la prédiction).
- **Composants interactifs avec retours visuels fluides** : loaders animés, onglets dynamiques, badges de confiance, tableaux stylisés.

6.5 Résultat

L'application **TrioVision** combine intelligence artificielle et design interactif pour offrir une solution complète de reconnaissance géographique à partir d'images. Elle rend les résultats du modèle à la fois compréhensibles visuellement et accessibles, tout en conservant une interface rapide, performante et agréable à utiliser.