

Compte Rendu Projet IA

SkinScanAI

Soukarieh Ali



1. Introduction

Dans ce projet, nous avons développé un modèle d'intelligence artificielle appelé **SkinScan** pour la classification des lésions cutanées. L'objectif de **SkinScan** est d'analyser des images de lésions cutanées pour déterminer si elles présentent un potentiel cancéreux. Ce modèle ne remplace pas un diagnostic médical professionnel mais vise à fournir des informations préliminaires utiles pour apaiser les inquiétudes ou encourager une consultation médicale si nécessaire.

2. Jeu de Données

Pour entraîner notre modèle d'IA, nous avons utilisé PyTorch et le dataset HAM10000, qui offre une collection complète d'images de lésions cutanées. Cependant, ce dataset présentait une limitation importante : il manquait d'images de peau saine. Pour remédier à cela, nous avons complété le dataset avec des images supplémentaires de peau saine provenant du dataset ISIC. Cet enrichissement était crucial pour garantir que le modèle puisse distinguer une peau normale des différents types de tumeurs, réduisant ainsi les faux positifs.

3. Modèles

Nous avons expérimenté avec plusieurs modèles pré entraînés, notamment :

- **ResNet50**
- **AlexNet**
- **DenseNet121**

Après des tests rigoureux, **DenseNet121** s'est avéré être le modèle le plus prometteur. Son architecture a démontré des performances supérieures pour gérer la complexité de notre jeu de données, et nous l'avons adopté pour des développements et ajustements ultérieurs.

4. Fonction de Perte

Nous avons utilisé la fonction de perte **CrossEntropyLoss** pour les raisons suivantes :

- **Adaptée à la classification binaire** : CrossEntropyLoss est idéale pour comparer les probabilités prédites avec les labels vérité terrain (0 ou 1).
- **Pénalisation des erreurs** : Elle impose une pénalité importante pour les prédictions incorrectes, notamment sur des labels catégoriques (ex. Vrai/Faux).

Avantages pour notre projet :

- Amélioration de la précision des probabilités prédites : le modèle a été encouragé à attribuer des valeurs proches de la vérité (ex. 0.9 au lieu de 0.4 pour une lésion cancéreuse).
 - Les erreurs significatives ont influencé plus efficacement les ajustements des poids pendant l'entraînement, améliorant ainsi les performances.
-

5. Fonction d'Optimisation

Nous avons choisi l'optimiseur **Adam** pour ses nombreux avantages :

- Combine le **Momentum** (pour réduire les oscillations) et **RMSProp** (ajustement dynamique du taux d'apprentissage).
- Permet une convergence rapide et stable vers un minimum de perte.

Bénéfices pour notre projet :

- Offre d'excellents résultats sans nécessiter de réglage manuel du taux d'apprentissage, contrairement à SGD.
 - A permis une convergence efficace et stable vers une perte faible, ce qui était crucial pour le succès de notre modèle.
-

6. Interface Graphique

Pour rendre le modèle accessible, nous avons développé une interface graphique conviviale :

- **Backend** : Un serveur Flask a été mis en place pour interagir avec le modèle entraîné. Le serveur traite les images soumises par les utilisateurs et retourne les prédictions.
- **Frontend** : Une application Next.js a été conçue pour offrir une expérience utilisateur fluide, permettant aux utilisateurs de télécharger des images et de recevoir des résultats en temps réel.

Cette architecture à deux couches garantit une communication fluide entre le modèle et l'utilisateur final, rendant le système pratique pour le déploiement.

7. Conclusion

Notre projet a permis de développer un modèle robuste et précis pour la classification des lésions cutanées en utilisant des techniques avancées d'apprentissage profond. En résolvant les limitations du jeu de données, en optimisant l'architecture du modèle et en créant une interface utilisateur intuitive, nous avons réussi à concevoir une solution efficace et conviviale. Les travaux futurs pourraient explorer une extension supplémentaire des jeux de données et des améliorations du modèle pour accroître la polyvalence et la fiabilité du système.