

Université de Montréal

Laboratoire DIRO

Maîtrise en Informatique



IFT 6390A - Fondements de l'apprentissage machine

**Kaggle Compétition : Identification de
Maladies de la Rétine**

Réalisé par :

SALIOU GUEYE

LIDAW FABRICE LEDJAKI

Nom de l'équipe sur kaggle : Gainde

7 décembre 2024

Table des matières

1	Introduction	2
2	Prétraitement des Données	2
2.1	Prétraitements de base : Premier Jalon	2
2.2	Prétraitements Avancés : Deuxième Jalon	3
3	Algorithmes d'Apprentissage	4
3.1	Premier Jalon : Implémentations Manuelles	4
3.2	Deuxième Jalon : Modèles Avancés et Deep Learning	5
4	Discussion	6
5	Conclusion	7

1 Introduction

Dans ce projet, nous abordons un problème de classification d'images ophtalmologiques visant à détecter automatiquement quatre types de pathologies rétinienues à partir d'images de Tomographie par Cohérence Optique (OCT). L'OCT, une technique d'imagerie médicale non invasive, offre une résolution microscopique en coupe transversale de la rétine, et est largement utilisée pour le diagnostic et le suivi de diverses maladies oculaires.

Les quatres catégories considérées sont :

- 0 : **Néovascularisation choroïdienne (CNV)** – croissance anormale de vaisseaux sanguins pouvant altérer la vision.
- 1 : **Œdème maculaire diabétique (DME)** – épaissement rétinien lié au diabète, entraînant un gonflement de la macula.
- 2 : **Drusen** – dépôts lipidiques sous-rétiens, associés au vieillissement et à la DMLA.
- 3 : **Rétine saine** – absence de toute pathologie.

L'objectif est de concevoir un pipeline d'apprentissage automatique capable de classer une image OCT (28x28 pixels) dans l'une de ces quatre catégories. L'évaluation des performances se fera par le taux d'erreur de classification sur un jeu de test.

Le projet se déroule en deux phases. La première impose de surpasser une référence établie à l'aide d'une régression logistique simple, sans l'usage de bibliothèques complètes telles que *scikit-learn*. Dans ce cadre, nous avons mis en place nos propres classifieurs basés sur la régression logistique et le SVM. Dans la seconde phase, nous avons développé un modèle basé sur les réseaux de neurones convolutifs (CNN), afin d'optimiser notre performance sur la compétition Kaggle.

2 Prétraitement des Données

Le prétraitement des données a significativement influé sur les performances de nos modèles, tant lors du premier jalon que lors du second jalon. Les images utilisées sont des coupes OCT rétinienues de dimensions 28×28 , chacune appartenant à l'une des quatre classes pathologiques.

2.1 Prétraitements de base : Premier Jalon

Dans la première phase, nous avons travaillé avec des modèles relativement simples (régression logistique et SVM) codés à l'aide de *NumPy*. À partir de la structure initiale, plusieurs étapes ont été mises en œuvre :

- **Chargement et conversion** : Les données, fournies sous forme de fichiers *pickle*, ont été lues puis converties en tableaux *NumPy*, facilitant ainsi

les opérations de normalisation et de réduction dimensionnelle. Une première inspection a permis de vérifier les dimensions des jeux de données, qui comptent respectivement 77982 et 19495 d'images de taille 28×28 pour l'ensemble d'entraînement et de test,

- **Séparation des Données en Entraînement et Validation** : Les données d'entraînement ont été divisées en ensembles d'entraînement (80%) et de validation (20%) de manière aléatoire et reproductible.
- **Normalisation** : Après avoir calculé la moyenne et l'écart-type des pixels sur l'ensemble d'entraînement, nous avons appliqué une normalisation pixel par pixel aux données d'entraînement, de validation et de test. Ce centrage-réduction favorise une convergence plus rapide et stable des algorithmes d'optimisation.
- **Réduction de dimension** : Une décomposition de type Truncated SVD a été testée afin de compresser l'information visuelle en un espace de plus faible dimension, limitant ainsi le surapprentissage potentiel et améliorant la capacité discriminante. Ainsi, nous avons choisi de réduire la dimension de $28 \times 28 = 784$ après avoir aplati les images à 100.
- **Équilibrage partiel** : Pour remédier au déséquilibre entre classes, nous avons appliqué des augmentations simples de données (rotations, flips, décalages) aux classes minoritaires qui sont les classes 1 et 2, générées grâce au module **scipy.ndimage**. Cette approche a permis de mieux équilibrer la distribution des étiquettes et de réduire les biais des modèles.

Ces prétraitements ont rendu le jeu de données plus homogène et représentatif, contribuant déjà à une amélioration par rapport au score de référence initial.

2.2 Prétraitements Avancés : Deuxième Jalon

Pour la seconde phase, l'intégration de *PyTorch* et l'utilisation du module `torchvision.transforms` ont permis d'automatiser et d'enrichir nos transformations :

- **Normalisation intégrée** : Comme précédemment, la normalisation par soustraction de la moyenne et division par l'écart-type a été appliquée. L'intégration directe de cette étape au sein du pipeline *PyTorch* assure cohérence et simplicité.
- **Transformations géométriques aléatoires** : Comme au premier jalon, nous avons intensifié ces transformations pour les classes sous-représentées, créant davantage d'exemples artificiels afin de rééquilibrer la distribution des labels. Nous avons alors introduit des rotations aléatoires ($\pm 30^\circ$), des flips horizontaux et des transformations affines mineures (légers décalages, zooms) afin d'accroître la variabilité des échantillons. Cela renforce la robustesse du réseau convolutif face aux variations de position et d'orientation.

- **Random Erasing** : Cette technique supprime aléatoirement une petite portion de l'image, obligeant le réseau à s'appuyer sur des caractéristiques globales. Elle réduit ainsi le risque de surapprentissage et améliore la généralisation.

Ces méthodes plus avancées ont enrichi l'ensemble d'entraînement, permettant au réseau convolutif de mieux généraliser, d'atteindre une précision plus élevée sur l'ensemble de validation, et d'obtenir de meilleurs scores sur le classement *Kaggle*.

```
# Standardisation des données
mean = X_train.mean(axis=0)
std = X_train.std(axis=0)
X_train_augmented = (X_train_augmented - mean) / std
X_val = (X_val - mean) / std
X_test = (X_test - mean) / std
```

FIGURE 1 – Standardisation

3 Algorithmes d'Apprentissage

La phase d'apprentissage a consisté à explorer une variété de méthodes de classification afin d'identifier la pathologie rétinienne à partir d'images OCT. Les approches diffèrent selon les deux jalons du projet : lors du premier jalon, l'usage de bibliothèques dédiées était proscrit, nous forçant à implémenter manuellement nos algorithmes (à l'aide de *NumPy*), tandis que le second jalon a tiré parti d'outils avancés (notamment *PyTorch*) et de modèles plus puissants, tels que les réseaux de neurones convolutifs (CNN).

3.1 Premier Jalon : Implémentations Manuelles

Dans cette première phase, l'objectif était de surpasser un score de référence obtenu par une régression logistique rudimentaire, sans recourir à des bibliothèques telles que *scikit-learn*. Nous avons donc construit nos propres algorithmes à partir de zéro.

Régression Logistique Multiclasses (One-vs-All) : Notre point de départ fut une régression logistique multiclasses, en adoptant une approche One-vs-All (OvA) : pour chaque classe, un classifieur binaire sépare cette classe du reste. Les prédictions finales proviennent de la comparaison des probabilités estimées par chaque classifieur OvA. L'entraînement s'est fait via descente de gradient avec régularisation L2. Ce modèle a permis de dépasser la baseline initiale, notamment grâce à une normalisation adaptée et un équilibrage partiel des classes.

SVM (Support Vector Machine) Manuel : Nous avons également développé un SVM basique, toujours sans bibliothèques externes, en testant notamment :

- Un SVM linéaire, entraîné par descente de gradient et régularisé.
- Un SVM avec noyau polynomial, introduisant une non-linéarité pour mieux séparer les classes complexes.

L'ajustement des hyperparamètres (paramètre C, degré du noyau) s'est avéré délicat, et l'entraînement plus coûteux. Les performances furent globalement similaires ou légèrement meilleures que la régression logistique, au prix d'un investissement en temps de calcul et d'ingénierie plus élevé.

Bilan du Premier Jalon

Les algorithmes implémentés manuellement (régression logistique et SVM) ont permis de dépasser la baseline. La régression logistique, renforcée par des pré-traitements adéquats, a fourni des résultats corrects, tandis que le SVM a offert de légères améliorations. Cependant, l'ensemble restait limité par la complexité des méthodes et le temps de calcul.

3.2 Deuxième Jalon : Modèles Avancés et Deep Learning

Avec l'autorisation d'utiliser des frameworks plus complets lors du deuxième jalon, nous avons opté pour un réseau de neurones convolutif (CNN), outil très efficace en classification d'images.

Réseau de Neurones Convolutif (CNN) Les CNN sont conçus pour extraire automatiquement les caractéristiques discriminantes des images. Notre architecture a comporté :

- Plusieurs couches de convolution suivies de normalisation par lot (Batch-Norm) et d'activation ReLU.
- Des couches de pooling (MaxPool) réduisant la dimension spatiale et concentrant l'information.
- Une partie fully-connected avec Dropout pour la classification finale.

L'entraînement s'est effectué grâce à *PyTorch*, en utilisant l'optimiseur Adam et une stratégie de *Cosine Annealing* pour ajuster le taux d'apprentissage. L'*Early Stopping* a permis de prévenir le surapprentissage. De plus, la fonction de perte (CrossEntropyLoss) a été pondérée par des poids de classe pour gérer le déséquilibre des étiquettes.

Améliorations et Ajustements Plusieurs raffinements ont été introduits :

- **Data Augmentation sophistiquée :** rotations, flips, *Random Erasing*, transformations affines, afin d'augmenter la diversité des données.

- **Régularisation** : Dropout et pénalités L2 pour limiter le surapprentissage.
- **Tuning d'hyperparamètres** : Ajustement du taux d'apprentissage, du nombre de filtres et de la profondeur du réseau en se basant sur les performances de validation.

Cette approche a nettement amélioré la généralisation et la précision du modèle, surpassant largement les scores du premier jalon.

Bilan du Deuxième Jalon

La mise en place d'un CNN a marqué une avancée considérable par rapport aux méthodes manuelles. L'utilisation d'outils de deep learning a facilité l'expérimentation, permis un entraînement plus rapide et abouti à un modèle plus performant et robuste. Le passage des approches linéaires et manuelles à un réseau convolutif complet illustre la montée en complexité et en efficacité rendue possible par les frameworks modernes.

En somme, la progression des algorithmes entre les deux jalons témoigne du gain apporté par les méthodes avancées et les architectures spécialisées en vision par ordinateur. Les modèles simples du premier jalon ont offert une base de compréhension et des points de comparaison, tandis que le CNN du second jalon, couplé à un prétraitement adéquat, s'est révélé nettement plus performant, démontrant la valeur ajoutée de l'apprentissage profond pour la classification d'images médicales.

4 Discussion

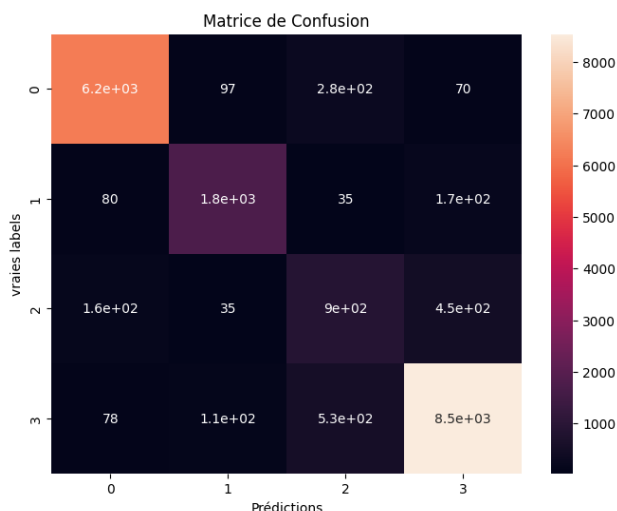


FIGURE 2 – Matrice de confusion des prédictions du modèle CNN sur l'ensemble validation

Cette matrice de confusion nous a permis de faire une analyse détaillée sur les métriques. Vous trouverez ci-dessus les scores en fonction des classes :

TABLE 1 – Analyse approfondie des métriques par classe

Classe	Précision	Rappel	F1-score
0	0.95	0.93	0.94
1	0.88	0.86	0.87
2	0.52	0.58	0.55
3	0.92	0.92	0.92

Les classes 0, 1, et 3 montrent des scores de F1 bien supérieurs (autour de 87-94%). Cela indique que ces classes sont mieux représentées et mieux définies dans l'espace des caractéristiques.

La précision (52%) et le rappel (58%) de la classe 2 qui est très en minoritaire sont significativement inférieurs à ceux des autres classes. Cela reflète une difficulté du modèle à distinguer correctement les instances de cette classe, malgré les efforts d'augmentation de données. Le F1-score est également faible (55%), ce qui souligne un déséquilibre dans les performances.

Ainsi, cela s'explique du fait que l'augmentation des données pour la classe 2 ne semble pas avoir donné des résultats satisfaisants. Cela pourrait être dû à plusieurs facteurs :

- **Qualité des données augmentées** : Si les données générées ne reflètent pas bien la distribution réelle, le modèle pourrait ne pas apprendre des informations pertinentes.
- **Overlap entre classes** : Si les caractéristiques de la classe 2 se chevauchent fortement avec celles d'autres classes, le modèle pourrait avoir du mal à différencier les instances, même avec plus de données.
- **Complexité de la classe** : Si la classe 2 présente des caractéristiques intrinsèquement complexes ou des variations importantes, l'augmentation ne suffira pas sans des ajustements plus profonds (comme des transformations avancées ou des approches architecturales spécifiques).

5 Conclusion

Ce projet met en lumière l'importance des choix méthodologiques dans la classification des pathologies rétinienues à partir d'images OCT. Il compare des approches classiques (régression logistique, SVM) sans bibliothèques et un réseau de neurones convolutif (CNN) basé sur PyTorch. Le CNN, grâce à l'augmentation de données et la régularisation, s'est montré plus performant, notamment dans la gestion des subtilités visuelles et du déséquilibre des classes.

Les limites incluent la faible résolution des images, le besoin de données plus variées, et le réglage chronophage des hyperparamètres, en plus du défi d'interprétabilité en milieu clinique. Les prétraitements avancés (normalisation, transformations, Random Erasing) et la régularisation (Dropout, L2) ont néanmoins amélioré la généralisation du modèle.

Pour aller plus loin, des architectures avancées (ResNet, DenseNet), le transfert d'apprentissage, davantage de données, et des approches d'explicabilité sont préconisés. Ce travail illustre ainsi le potentiel du deep learning dans la classification OCT, tout en ouvrant des perspectives pour améliorer la robustesse, la qualité, et l'interprétation des modèles, dans le but de soutenir le diagnostic clinique.

Références

- [1] Documentation fournie pour le cours IFT 6390A.
- [2] Huang, G., Liu, Z., Van der Maaten, L., & Weinberger, K. Q. (2017). *Densely Connected Convolutional Networks*. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700–4708.
- [3] PyTorch Documentation
- [4] Torchvision Transforms
- [5] Stack Overflow
- [6] ChatGPT