

Rapport MLRF

theo.stosse

July 2023

1 Introduction

1.1 Presentation du dataset

Le jeu de données CIFAR-10 est une collection largement utilisée de 60 000 images 32x32 pixels en couleurs. Elle est divisée en 10 classes (automobile, avion, oiseau, chat, cerf, chien, grenouille, cheval, navire, camion), avec 6000 images par classe. La base de données est également divisée en un ensemble d'entraînement de 50 000 images et un ensemble de test de 10 000 images. Le jeu de données est donc parfaitement équilibré, ce qui rend les tâches de classification plus simple à effectuer.

En raison de sa taille relativement petite et de sa diversité de classes, CIFAR-10 est souvent utilisée comme référence pour évaluer les nouvelles méthodes de classification d'images. Ainsi, un nombre considérable de travaux ont été réalisés sur cette base de données, en particulier dans le domaine de l'apprentissage profond.

1.2 Etat de l'art

Avant l'avènement des réseaux de neurones profonds, des approches telles que les méthodes basées sur les descripteurs de texture, les classifieurs linéaires, le SVM et les Random Forests étaient couramment utilisées pour la classification des images CIFAR-10. Ces méthodes reposaient souvent sur l'extraction manuelle de caractéristiques, en utilisant des techniques telles que le SIFT (Scale-Invariant Feature Transform), le HOG (Histogram of Oriented Gradients), ou la LBP (Local Binary Patterns).

Cependant, avec l'essor des techniques d'apprentissage profond, les performances sur CIFAR-10 ont connu une amélioration significative. Des réseaux de neurones convolutifs comme ResNet ou DenseNet ont montré de bonnes performances sur ce jeu de données.

L'état de l'art actuel en matière de classification sur CIFAR-10 utilise une approche innovante qui combine l'utilisation de transformers avec une "classe de fond" introduite dans le modèle. Cette méthode, qui a permis d'obtenir des performances de pointe non seulement sur CIFAR-10, mais aussi sur d'autres bases de données comme STL-10, CIFAR-100, Oxford-102, Caltech-101, et CINIC-10,

améliore la généralisation des modèles tout en nécessitant moins de ressources de calcul par rapport à l'apprentissage multitâche traditionnel.

2 Méthode

La méthodologie de ce projet consiste en l'utilisation de trois techniques différentes d'extraction de caractéristiques, combinées à trois algorithmes de classification distincts.

2.1 Extraction de caractéristiques

2.1.1 Aplatissement simple

La technique la plus simple d'extraction de caractéristiques est l'aplatissement de l'image, qui convertit simplement l'image en une matrice unidimensionnelle. Supposons qu'une image I soit une matrice $m \times n$, où m et n sont respectivement le nombre de lignes et de colonnes de l'image, l'image aplatie sera un vecteur de taille mn .

2.1.2 Histogram of Oriented Gradients (HOG)

L'histogramme des gradients orientés (HOG) est une technique d'extraction de caractéristiques couramment utilisée en vision par ordinateur et en traitement d'images pour la détection d'objets. HOG compte les occurrences de directions de gradient dans des parties spécifiques d'une image. Cette méthode est implémentée en plusieurs étapes :

1. Calcul du gradient de l'image : la première étape consiste à calculer le gradient de l'image pour identifier la direction et l'intensité des bords. Les gradients peuvent être calculés en utilisant des filtres de Sobel dans les directions x et y .
2. Division de l'image en cellules : l'image est divisée en petites cellules connectées.
3. Calcul de l'histogramme de gradient pour chaque cellule : pour chaque cellule de l'image, on calcule un histogramme de gradient. Cela fournit une représentation de la distribution des intensités et des directions des gradients dans chaque cellule.
4. Normalisation : pour améliorer la précision, les histogrammes de chaque cellule sont normalisés en utilisant les histogrammes des cellules voisines.

2.1.3 Opérateur Sobel

L'opérateur Sobel est utilisé en traitement d'image, notamment pour la détection des contours. L'opérateur Sobel calcule le gradient de l'intensité de l'image à

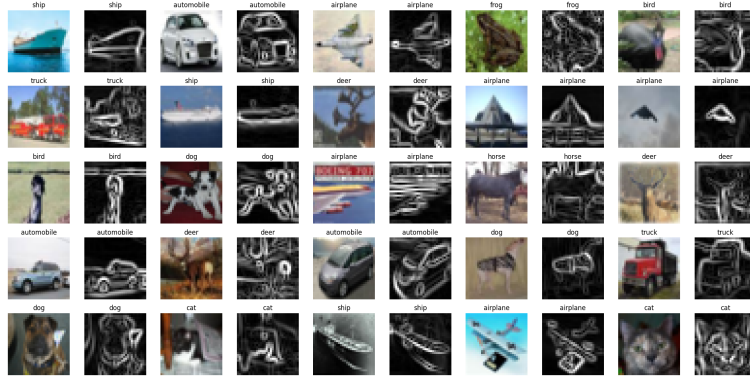


Figure 1: L'opérateur Sobel appliqué a différentes images du dataset (originale à droite, transformée à gauche)

chaque point, donnant la direction et l'intensité du contour. Les filtres de Sobel, un pour chaque direction x et y , sont appliqués à l'image pour produire des images de gradient séparées.

2.2 Classificateurs

2.2.1 Régression logistique

La régression logistique est un algorithme de classification statistique. Pour une instance donnée x_i , la probabilité que cette instance appartienne à une certaine classe est donnée par :

$$p(y = 1|x_i) = \frac{1}{1 + e^{-(w^T x_i + b)}} \quad (1)$$

où w est le vecteur de poids, b est le biais et e est la base du logarithme naturel.

2.2.2 Forêts aléatoires

L'algorithme des forêts aléatoires est un algorithme d'apprentissage ensembliste qui consiste à entraîner un grand nombre d'arbres de décision indépendants sur des sous-ensembles aléatoires des données d'entraînement, puis à moyenner leurs prédictions.

Chaque arbre est formé en sélectionnant aléatoirement un sous-ensemble de caractéristiques à chaque nœud, puis en choisissant la meilleure coupure. La prédiction finale est donnée par le mode (classification) ou la moyenne (régression) des prédictions de tous les arbres.

2.2.3 K-Nearest Neighbors (KNN)

L'algorithme des K-Nearest Neighbors est un algorithme d'apprentissage non paramétrique basé sur la distance. Pour une nouvelle instance, KNN identifie les k instances les plus proches dans l'ensemble d'apprentissage et attribue à la nouvelle instance la classe la plus courante parmi ces k instances. La distance entre deux instances x_i et x_j est généralement calculée en utilisant la distance euclidienne :

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^n (x_{il} - x_{jl})^2} \quad (2)$$

où n est le nombre de caractéristiques.

2.3 Choix des hyperparamètres

Le choix des hyperparamètres appropriés pour les algorithmes de machine learning est une étape cruciale pour obtenir des performances optimales. Dans ce projet, une recherche exhaustive d'hyperparamètres, également connue sous le nom de *Grid Search*, a été utilisée pour identifier les meilleurs hyperparamètres pour les algorithmes de classification.

L'ensemble des hyperparamètres considérés était relativement réduit pour maintenir un temps de calcul raisonnable. Pour chaque combinaison d'hyperparamètres, les performances ont été évaluées à l'aide de la validation croisée en 2 folds.

La validation croisée est une technique de validation de modèle qui divise l'ensemble de données en un certain nombre de "folds" ou de sous-ensembles. Le modèle est ensuite entraîné sur tous les sous-ensembles à l'exception d'un, qui est utilisé comme ensemble de test. Ce processus est répété pour chaque sous-ensemble, de sorte que chaque instance de l'ensemble de données est utilisée à la fois pour l'entraînement et le test. Cela permet d'obtenir une estimation plus robuste de la performance du modèle, car elle ne dépend pas d'une division particulière des données en ensembles d'entraînement et de test.

Dans ce cas, une validation croisée en 2 folds a été utilisée, ce qui signifie que l'ensemble de données a été divisé en deux sous-ensembles de taille égale. Chaque sous-ensemble a été utilisé une fois comme ensemble d'entraînement et une fois comme ensemble de test.

La combinaison d'hyperparamètres qui a donné les meilleures performances en validation croisée a été choisie comme ensemble d'hyperparamètres optimal pour chaque algorithme. Cela garantit que les hyperparamètres choisis sont susceptibles de généraliser le mieux possible à de nouvelles données, tout en évitant le surapprentissage sur l'ensemble d'entraînement.

Vous trouverez tous les hyperparamètres trouvés dans le notebook ou en exécutant les scripts individuels présents dans le dépôt git.

3 Resultats

Toutes les visualisations des dits résultats sont présentes en annexe.

3.1 Régression logistique

L'analyse des rapports de classification fournis pour le jeu de données CIFAR-10 révèle plusieurs points intéressants. En utilisant une régression logistique avec un simple aplatissement de l'image comme technique d'extraction de caractéristiques, nous obtenons une précision pondérée et une moyenne macro de 0,41, avec une précision globale de 0,42. Les classes individuelles montrent une variation considérable dans leur performance, avec des scores F1 allant de 0,26 à 0,54. Les classes 3 et 2 ont le plus bas score F1, ce qui suggère que le modèle a du mal à classer correctement ces classes.

En utilisant la technique d'extraction de caractéristiques HOG avec la régression logistique, il y a une amélioration notable de la performance. La précision globale augmente à 0,47 et les scores F1 pour chaque classe montrent également une amélioration, en particulier pour la classe 6 qui obtient un score F1 de 0,55. Cette technique semble donc plus efficace pour extraire des caractéristiques pertinentes des images CIFAR-10 et aider le modèle de régression logistique à mieux généraliser.

Enfin, l'utilisation de l'opérateur Sobel pour l'extraction des caractéristiques avec la régression logistique a conduit à une dégradation de la performance du modèle. La précision globale tombe à 0,32 et les scores F1 pour les classes individuelles sont également inférieurs, en particulier pour la classe 3 qui obtient un score F1 de 0,11. Cette technique d'extraction de caractéristiques ne semble pas adaptée à ce problème de classification sur le jeu de données CIFAR-10, du moins lorsqu'elle est utilisée en conjonction avec la régression logistique.

3.2 Random Forests

L'analyse des rapports de classification obtenus à partir de l'algorithme Random Forest révèle des résultats mixtes. Lorsque nous utilisons un simple aplatissement de l'image comme technique d'extraction de caractéristiques, la précision pondérée et la moyenne macro atteignent seulement 0,38, avec une précision globale de 0,31. Il est intéressant de noter que le rappel varie considérablement entre les classes, avec des scores allant de 0,16 (classes 1 et 9) à 0,62 (classe 8). Cette variation indique que le modèle a du mal à maintenir une performance cohérente sur toutes les classes.

L'application de la technique d'extraction de caractéristiques HOG en conjonction avec la forêt aléatoire améliore la performance, en élevant la précision globale à 0,47. Les scores F1 pour chaque classe montrent également une amélioration, en particulier pour la classe 6 qui atteint un score F1 de 0,55. Il semble donc que la technique HOG soit plus efficace pour extraire des caractéristiques pertinentes des images CIFAR-10 qui aident la forêt aléatoire à mieux classer.

Enfin, l'utilisation de l'opérateur Sobel pour l'extraction de caractéristiques avec la forêt aléatoire conduit à une performance similaire à celle de l'aplatissement simple de l'image, avec une précision globale de 0,32. Cependant, la distribution des scores F1 est plus uniforme entre les classes, avec des scores allant de 0,11 (classe 3) à 0,42 (classe 0). Cela suggère que l'opérateur Sobel, malgré une performance globale plus faible, pourrait offrir une performance plus équilibrée sur les différentes classes.

3.3 KNN

L'analyse des rapports de classification obtenus à partir de l'algorithme K-Nearest Neighbors (KNN) donne des résultats variés. Lorsqu'un simple aplatissement de l'image est utilisé comme technique d'extraction de caractéristiques, la précision pondérée et la moyenne macro sont seulement de 0,38, avec une précision globale de 0,31. On observe que le rappel varie beaucoup entre les classes, avec des scores allant de 0,16 (classes 1 et 9) à 0,62 (classe 8). Cela indique que le modèle a du mal à maintenir une performance cohérente sur toutes les classes.

Lorsqu'on utilise la technique d'extraction de caractéristiques HOG avec KNN, la performance s'améliore, augmentant la précision globale à 0,48. Les scores F1 pour chaque classe montrent également une amélioration, en particulier pour la classe 6 qui atteint un score F1 de 0,53. Il semble donc que la technique HOG soit plus efficace pour extraire des caractéristiques pertinentes des images CIFAR-10 qui aident KNN à mieux classer les images.

Enfin, l'utilisation de l'opérateur Sobel pour l'extraction de caractéristiques avec KNN conduit à une performance globale inférieure, avec une précision de seulement 0,24. Cependant, le score F1 pour la classe 4 a augmenté à 0,26, ce qui suggère que l'opérateur Sobel peut être plus efficace pour certaines classes spécifiques, même si la performance globale est plus faible.

3.4 Analyses

3.4.1 Effet de la technique d'extraction de caractéristiques

Le choix de la technique d'extraction de caractéristiques a une influence significative sur la performance des modèles. L'aplatissement simple de l'image s'est révélé être la technique la moins efficace pour tous les trois modèles. La technique d'extraction de caractéristiques HOG s'est avérée la plus efficace en termes d'amélioration de la performance des modèles, avec une augmentation notable de la précision et des scores F1 dans la plupart des classes. L'opérateur Sobel, bien qu'il ait montré une amélioration des performances avec Random Forest, a conduit à une diminution des performances avec K-Nearest Neighbors et n'a pas montré d'amélioration significative avec la Régression Logistique.

3.4.2 Performance des modèles de classification

La Régression Logistique a montré une performance modeste avec une simple technique d'aplatissement de l'image, mais a mieux performé avec HOG et a eu une performance légèrement inférieure avec Sobel. Random Forest a montré des performances similaires à la Régression Logistique, avec une amélioration notable en utilisant HOG et une légère amélioration avec Sobel. K-Nearest Neighbors, en revanche, a affiché une amélioration substantielle des performances avec HOG mais une chute significative avec l'utilisation de l'opérateur Sobel.

3.4.3 Variation de la performance entre les classes

Un aspect notable était la variation significative de la performance entre les différentes classes pour chaque modèle et chaque technique d'extraction de caractéristiques. Cela pourrait indiquer que certains modèles et techniques d'extraction de caractéristiques sont plus adaptés à la classification de certaines classes que d'autres.

En conclusion, ces résultats soulignent l'importance de la sélection de la technique d'extraction de caractéristiques appropriée en fonction du modèle de classification utilisé et des caractéristiques spécifiques des données. Ils soulignent également la nécessité d'effectuer des analyses de performances à l'échelle de la classe pour comprendre plus en détail où et comment les modèles réussissent et échouent. Enfin, ils mettent en évidence le besoin de recherches et d'expérimentations supplémentaires pour identifier les combinaisons optimales de modèles de classification et de techniques d'extraction de caractéristiques pour des jeux de données spécifiques tels que CIFAR-10.

4 Conclusion

5 Conclusion

Le projet présenté dans ce rapport a exploré différentes méthodes d'extraction de caractéristiques et de classification pour la classification d'images sur le jeu de données CIFAR-10. Bien que nous ayons réussi à construire des modèles fonctionnels avec différentes combinaisons de techniques, nos résultats restent loin de l'état de l'art en matière de performance de classification sur ce jeu de données.

Cependant, il est important de noter que notre approche n'a pas impliqué l'utilisation de techniques de deep learning, qui sont actuellement les plus efficaces pour la classification d'images et ont atteint les performances les plus élevées sur des jeux de données tels que CIFAR-10. Étant donné que les techniques que nous avons utilisées, comme la régression logistique, les forêts aléatoires et le K-NN, ainsi que les méthodes d'extraction de caractéristiques comme l'aplatissement d'image, HOG et l'opérateur Sobel, sont plus simples et ne bénéficient pas de la capacité du deep learning à apprendre des représentations

hiérarchiques des données, il n'est pas surprenant que nos performances soient inférieures à celles obtenues avec les méthodes de deep learning.

Néanmoins, cette étude offre une perspective intéressante sur les défis de la classification d'images sans deep learning et offre une base solide pour l'exploration de techniques plus avancées et potentiellement plus efficaces. Il convient également de noter que, malgré la performance inférieure en termes de précision de classification, les méthodes utilisées dans cette étude ont l'avantage d'être plus interprétables que les méthodes de deep learning, ce qui peut être un atout précieux dans certaines applications où l'explicabilité est une nécessité. Enfin, les leçons tirées de ce projet peuvent être utiles pour la conception de nouvelles approches et méthodes pour relever les défis posés par la classification d'images.

6 Annexe

Le dépôt git contenant le code est disponible ici : <https://github.com/Wiky1/MLRF>.

Toutes les images présentées ici, sont aussi disponibles au format .jpg sur le dépôt git.

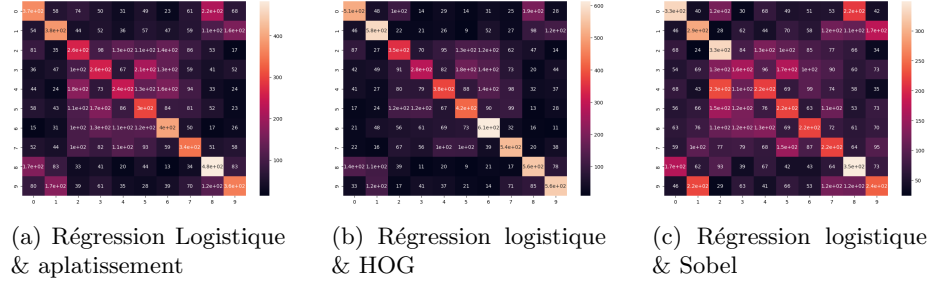


Figure 2: Matrices de confusion des classificateurs utilisant la régression logistique

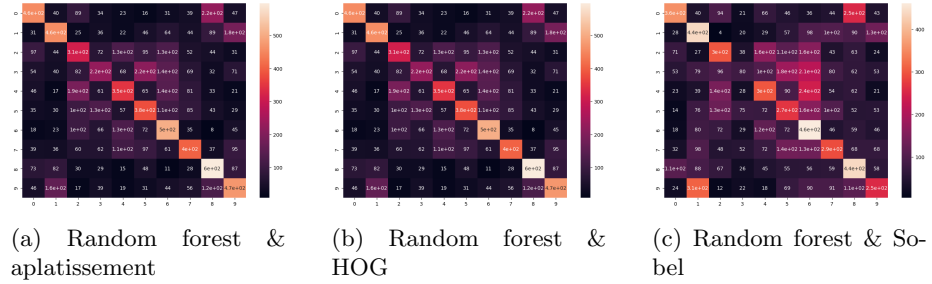


Figure 3: Matrices de confusion des classificateurs utilisant les Random forest

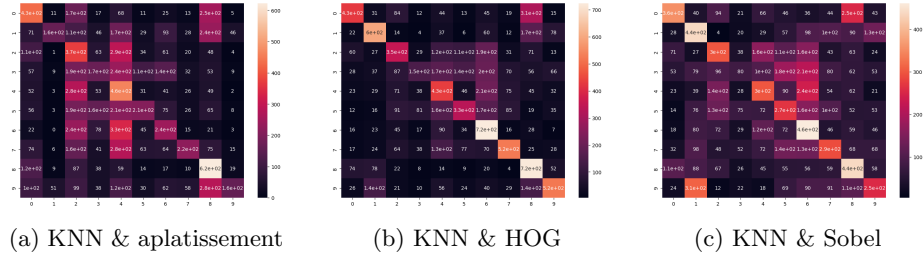


Figure 4: Matrices de confusion des classificateurs utilisant les KNN

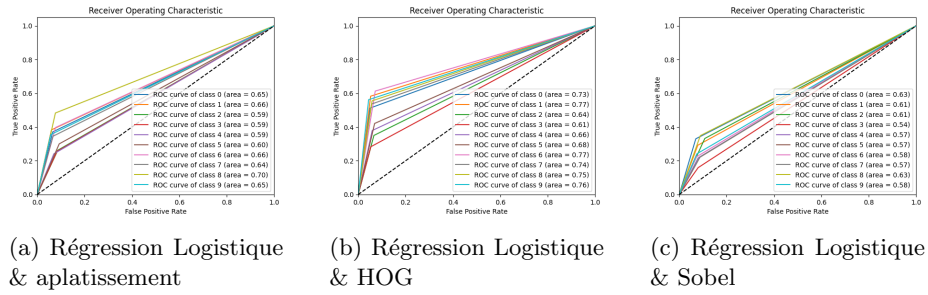


Figure 5: Courbes ROC des classificateurs utilisant la régression logistique

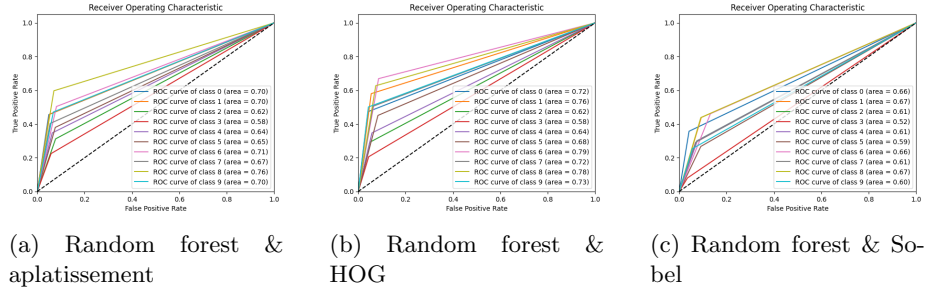


Figure 6: Courbes ROC des classificateurs utilisant les Random forests

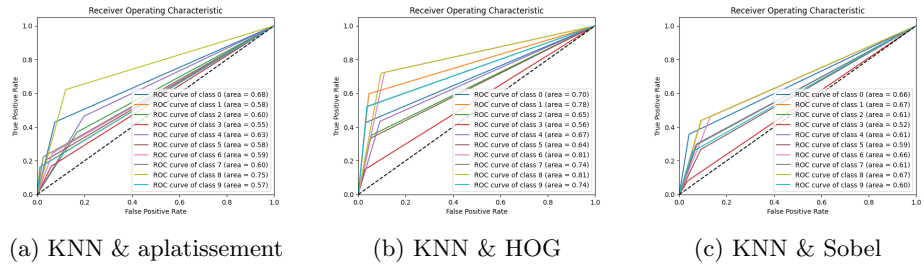


Figure 7: Courbes ROC des classificateurs utilisant les KNN

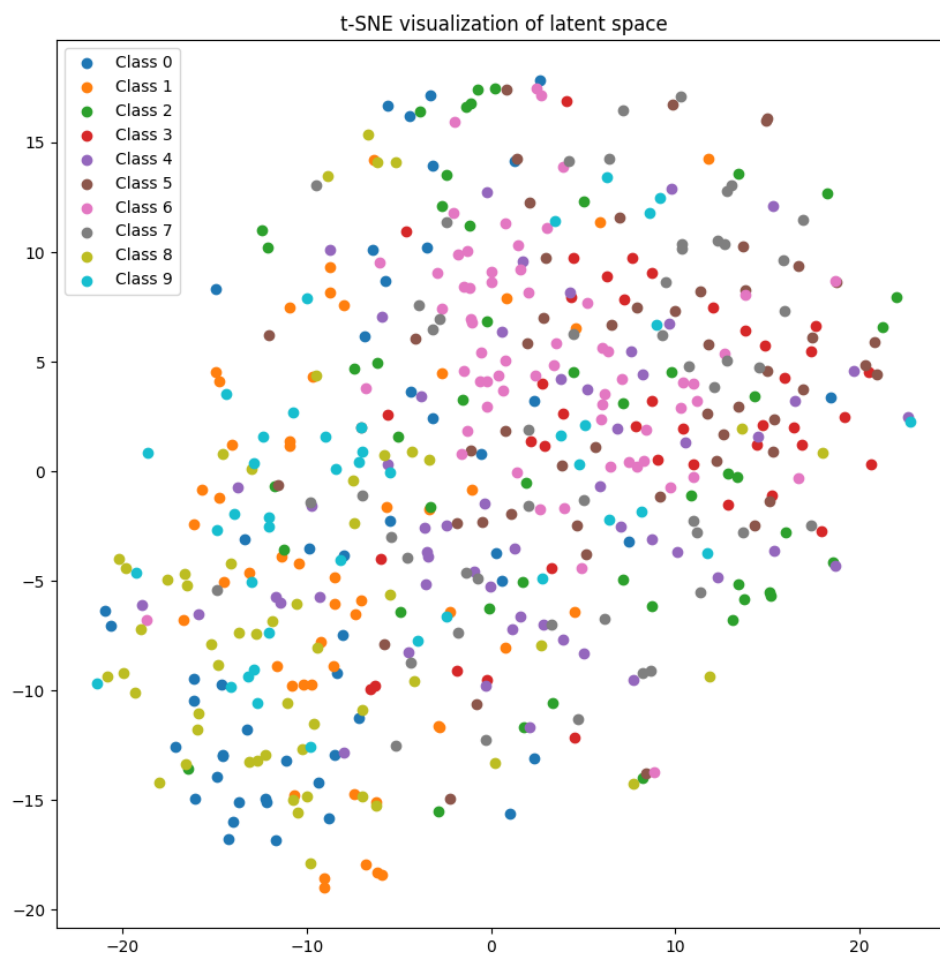


Figure 8: L'interprétation de la représentation latente d'HOG obtenue via t-SNE s'est révélée difficile en raison du nombre élevé de classes. Les regroupements de données dans l'espace de la t-SNE sont devenus confus et superposés, rendant difficile la distinction entre les classes, même si certaines structures semblent ressortir.