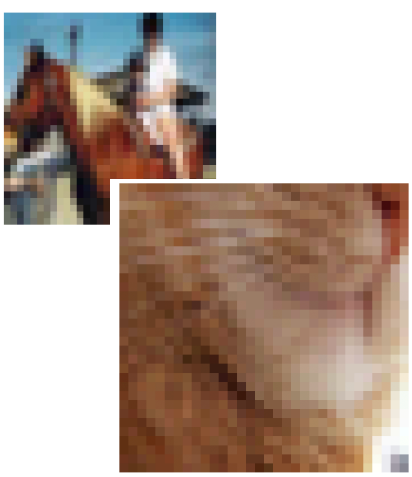
RAPPORT– MINI CHALLENGE DE CLASSIFICATION D'IMAGES EN COULEUR



1.Présentation du challenge

Le projet consiste à résoudre un problème de classification multi-classes à partir d'un jeu de données d'images en couleur de taille 32x32 pixels, réparties en 10 classes. Chaque image est codée sous la forme d'un vecteur de taille 3072 (3 canaux RGB × 32 × 32 pixels).
L'objectif est d'entraîner des modèles de classification sur un jeu d'apprentissage de 20000 images, puis de prédire les classes des 10000 images de test.

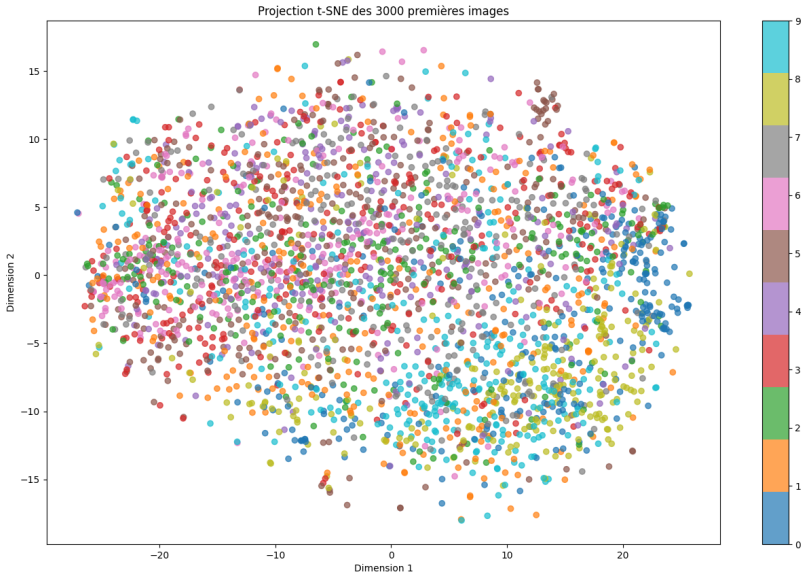
2. Observation des données

Dès le début, on constate que les données sont très **condensées** et réparties dans un espace de grande dimension, ce qui rend leur classification non triviale.

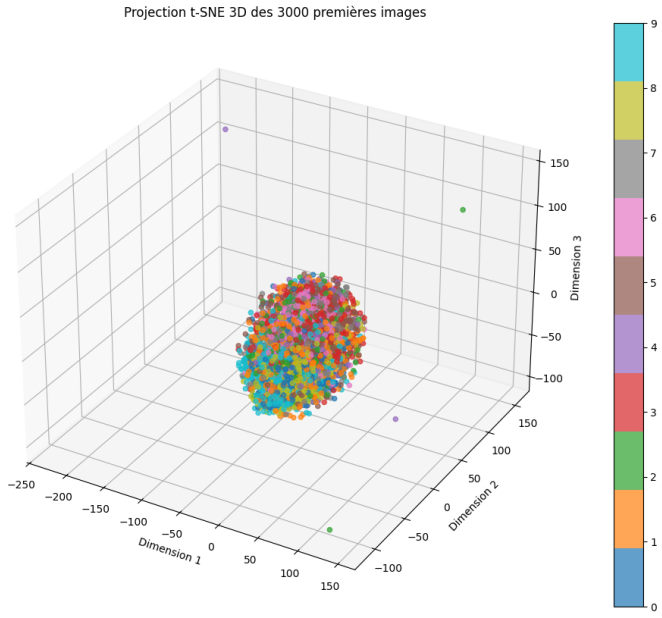
Pour illustrer cela, j'ai utilisé la méthode **t-SNE** afin de projeter les données dans un espace 2D puis 3D.

On remarque que les classes sont très entremêlées, ce qui montre que les frontières entre classes sont complexes :

- En 2D :



- En 3D :



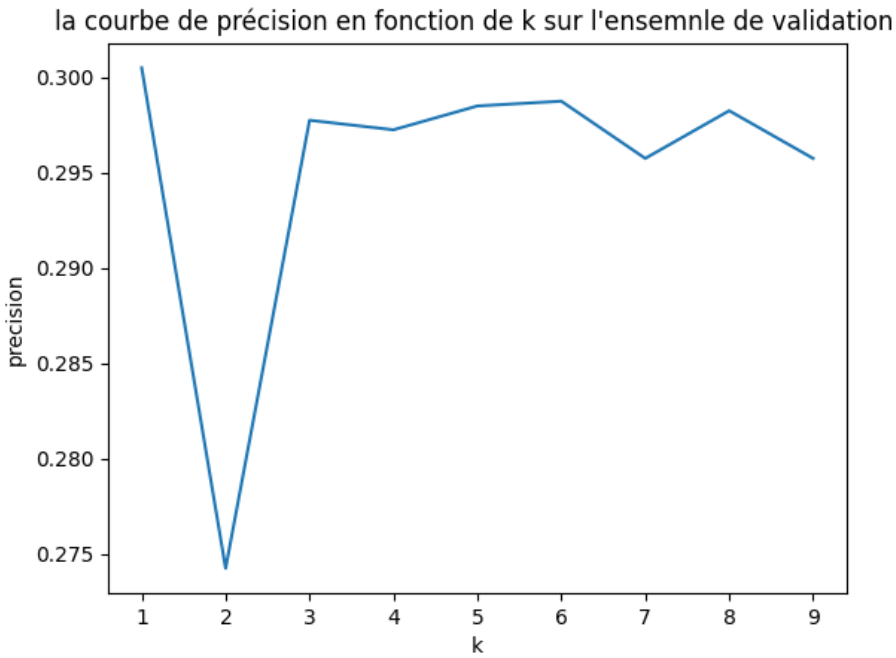
3. Méthodes de classification

3.1 k plus proches voisins (KNN)

J'ai testé plusieurs valeurs de k. Les résultats obtenus sont faibles :

- Précision avec k = 3 : **29.76%**
- Précision avec k = 6 : **30.38%**

Cela montre que KNN, bien que simple à implémenter, n'est pas performant sur ce jeu de données complexe. De plus, le temps de calcul est élevé.

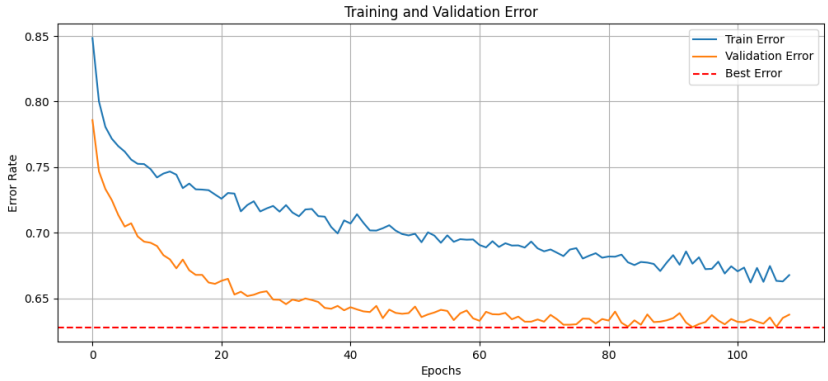


3.2 Régression logistique multivariée

Deux versions ont été testées :

- Sans early stopping → **33.97 %**
- Avec early stopping → **36.3 %**

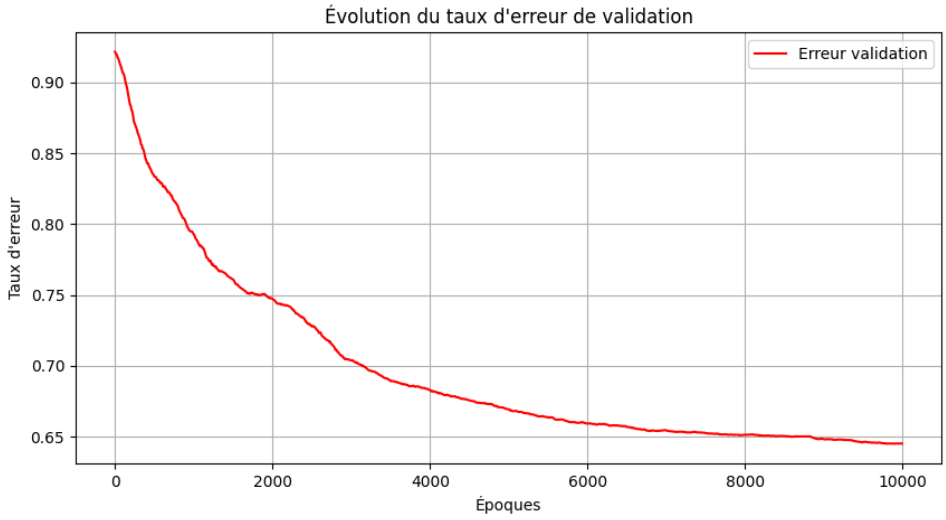
On observe une **légère amélioration** avec early stopping, qui permet d'éviter le sur-apprentissage en arrêtant l'entraînement dès que les performances sur les données de validation cessent de s'améliorer.



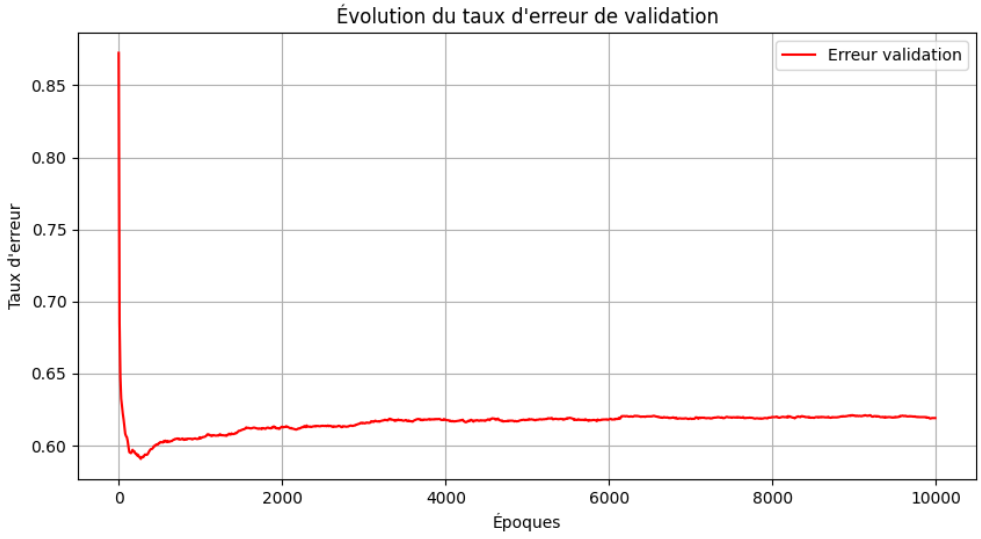
En comparant les courbes d'erreur de validation pour SGD et Adam, on peut clairement constater que l'optimiseur Adam permet une convergence plus rapide et plus stable que SGD. La courbe d'Adam descend plus rapidement et atteint un taux d'erreur plus bas, tandis que celle de SGD stagne autour de 0.6. Cela montre que le choix de l'optimiseur a un impact considérable sur les performances du modèle.

Voici les courbes correspondantes :

- SGD (sans early stopping) :



- Adam (sans early stopping) :



3.3 Réseau de neurones fully connected (MLP)

➤ Modèle avec beaucoup d'époques :

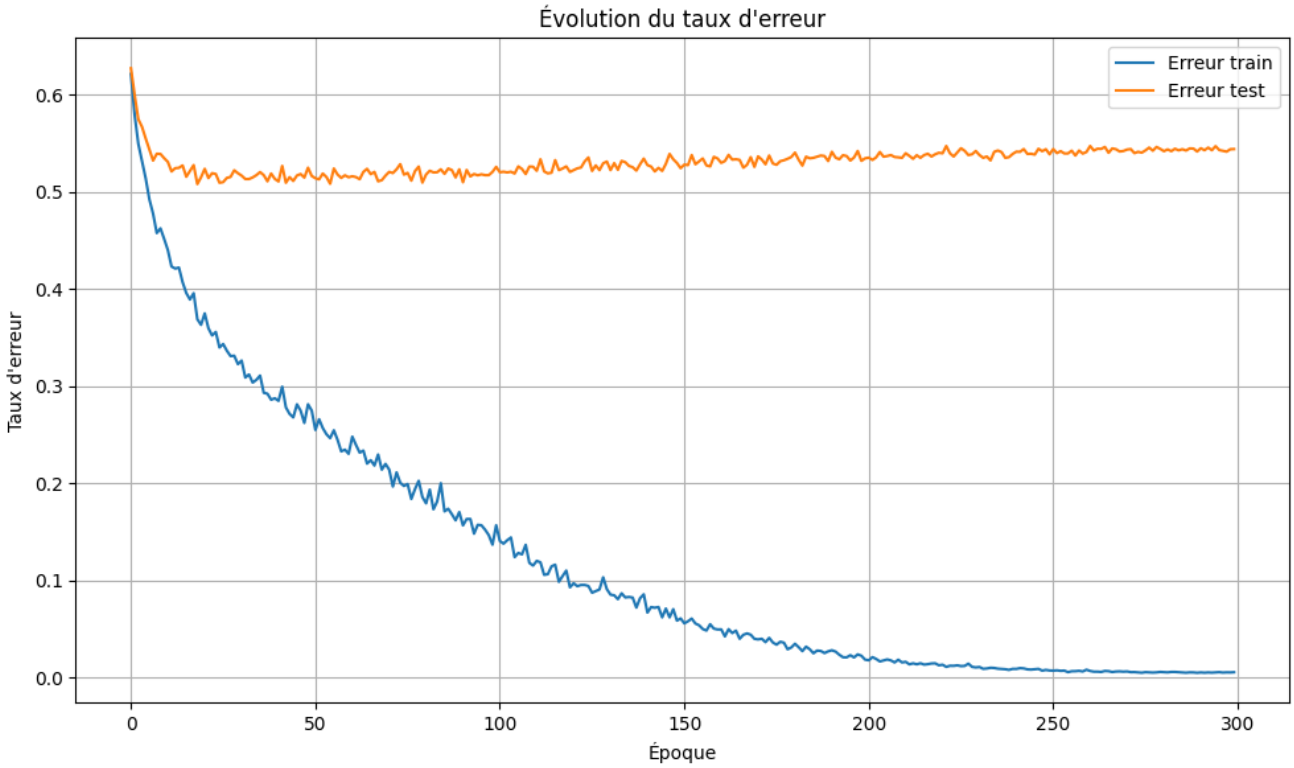
- Résultat très faible : **11.68 %**
- Montre que le surapprentissage ou un mauvais choix d'hyperparamètres peut dégrader fortement les performances.

➤ Modèle avec 3000 époques + early stopping :

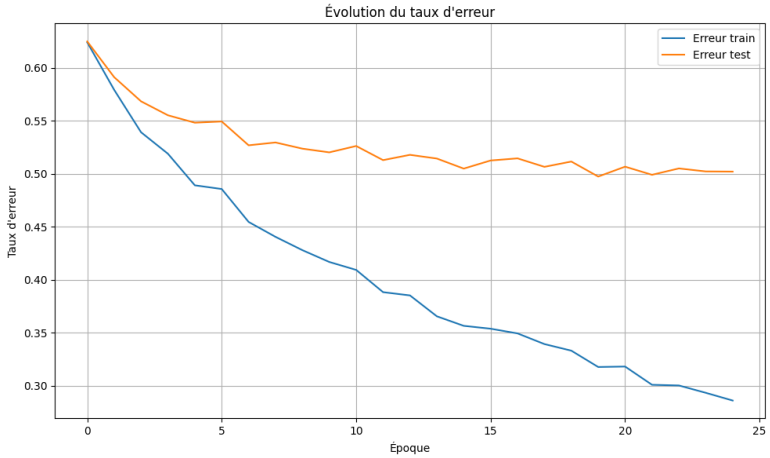
- Résultat : **24.94 %**
- Meilleur mais reste insuffisant.

➤Version rapide avec BatchNorm, Dropout, petit réseau :

- h1=128, h2=64, **3 couches**, **ReLU**, **dropout**, **30 époques → 49.13 %**
- Même architecture, **300 époques**, sans early stopping **→ 46.32 % :**
  comme on remarque dans la courbe en dessous erreur test remonte lorsque on est au alentour de 50 **époques,** c'est pourquoi le early shopping est une bonne solution



- h1=256, h2=64 avec early stopping → **49.19 %**



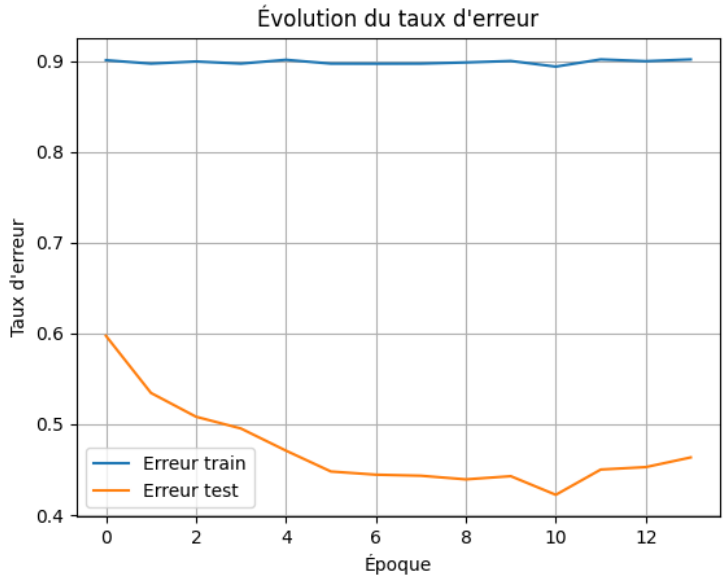
- h1=256, h2=128, h3=64 avec early stopping → **48.4 %**

On remarque que le choix des dimensions, du dropout, de l'early stopping et du nombre d'époques a un vrai impact sur la performance. L'usage de BatchNorm + Dropout améliore significativement les résultats.

3.4 Réseau de neurones convolutionnel (CNN)

Enfin, un CNN a été implémenté. C'est cette méthode qui a donné le **meilleur résultat** :

•Précision obtenue : 53.46 %



Ce résultat confirme que les CNN sont plus adaptés à la classification d'images, car ils exploitent la structure spatiale des données (contrairement au réseaux fully connected ou KNN).

4. Conclusion

Ce challenge m'a permis d'explorer plusieurs méthodes de classification sur un jeu de données image complexe. Voici un récapitulatif des performances :

Méthode	Précision (%)
KNN (k=3)	29.76
KNN (k=6)	30.38
Régression logistique	33.97
Régression logistique (ES)	36.30
MLP simple	11.68
MLP (3000 epochs + ES)	24.94
MLP rapide (30 epochs)	49.13
MLP rapide (300 epochs)	46.32
MLP ES (256, 64)	49.19
MLP ES (256,128,64)	48.40
CNN	53.46

Le CNN est de loin la meilleure méthode pour ce challenge. Les autres méthodes nécessitent un réglage fin des hyperparamètres et restent limitées par rapport à la capacité d'abstraction des CNN.