

IS
ECOLE INTERNATIONAL

MASTER-INFORMATIQUE
SYSTÈME INTELLIGENT ET MULTIMÉDIA
RAPPORT-APPRENTISSAGE

Image and Time Series Classification
using CNN and Hybrid CNN-LSTM in
PyTorch

Étudiant :

Christine TOUMBA NGONGO
David LUTALA LUSHULI

Enseignant

CISSOKO MAMADOU BEN
HAMIDOU

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 2 |
| 2 | Description des datasets | 2 |
| 2.1 | CIFAR-10 | 2 |
| 2.2 | ECG5000 | 2 |
| 3 | Méthodologie | 2 |
| 3.1 | Prétraitement des données | 2 |
| 3.2 | Architectures des modèles | 2 |
| 3.3 | Configuration d'entraînement | 3 |
| 4 | Résultats | 3 |
| 4.1 | Classification d'images (CIFAR-10) | 3 |
| 4.1.1 | Avec Data Augmentation | 3 |
| 4.1.2 | Sans Data Augmentation | 6 |
| 4.1.3 | Analyse Visuelle des Courbes | 7 |
| 4.2 | Classification de séries temporelles (ECG5000) | 9 |
| 4.3 | Analyse des hyperparamètres | 9 |
| 5 | Conclusion | 10 |

Table des figures

| | | |
|---|---|---|
| 1 | les performances du modèle sur le jeu de test CIFAR-10, | 3 |
| 2 | les performances du modèle sur le jeu de test CIFAR-10 Sans Data Augmentation | 6 |
| 3 | Représentation du Courbe de perte | 7 |
| 4 | Représentation du Courbe de précision | 8 |

1 Introduction

Le développement rapide des technologies de deep learning a révolutionné le domaine de l'apprentissage automatique, notamment dans les tâches de classification d'images et de séries temporelles. Ce travail pratique a pour but d'appliquer et de comparer des modèles de réseaux neuronaux convolutifs (CNN) et hybrides CNN-LSTM sur deux types de données distincts : des images naturelles (CIFAR-10) et des signaux temporels (ECG5000). L'objectif est de mieux comprendre le comportement de ces modèles, d'observer l'impact de l'augmentation de données sur la performance, et d'évaluer l'apport des modèles récurrents pour les données dépendantes du temps.

2 Description des datasets

2.1 CIFAR-10

Le jeu de données CIFAR-10 est un benchmark classique en vision par ordinateur. Il contient 60 000 images en couleur de petite taille (32x32 pixels), réparties en 10 classes d'objets courants tels qu'avions, voitures, oiseaux, chats, chiens, grenouilles, chevaux, bateaux, camions et cerfs. Chaque classe est équilibrée avec 6 000 images.

2.2 ECG5000

Le jeu de données ECG5000 fait partie de la UCR Time Series Classification Archive. Il contient des signaux électrocardiographiques de 140 points temporels, classés en 5 catégories correspondant à différents types de battements cardiaques.

Nous avons choisi ECG5000 car il illustre bien l'application des réseaux neuronaux à des données temporelles réelles.

3 Méthodologie

3.1 Prétraitement des données

Pour CIFAR-10, nous avons appliqué des techniques d'augmentation de données afin de générer plus de variabilité à partir des images existantes : retournements horizontaux aléatoires, recadrages, variations de couleurs et rotations. Cela permet au modèle de mieux généraliser sur de nouvelles images.

Pour ECG5000, nous avons normalisé les séries temporelles pour qu'elles aient une moyenne nulle et une variance unité. Les signaux ont été remodelés pour être traités par des couches convolutionnelles 1D.

3.2 Architectures des modèles

— CNN pour CIFAR-10

Nous avons construit un CNN composé de trois blocs convolutionnels suivis de couches de pooling et de normalisation par batch. À la sortie, deux couches fully-connected permettent de produire les prédictions sur les 10 classes.

Chaque couche utilise la fonction d'activation ReLU pour introduire de la non-linéarité, et une couche Dropout est utilisée pour limiter le surapprentissage.

— CNN pour ECG5000

Pour les séries temporelles, nous avons conçu un CNN 1D adapté. Les signaux passent par plusieurs couches convolutionnelles 1D avec pooling, avant d'être aplatis puis classifiés.

— CNN-LSTM pour ECG5000

Nous avons ensuite enrichi le modèle en ajoutant un LSTM après les couches convolutionnelles. Ce module récurrent permet de capturer les dépendances temporelles à long terme, améliorant ainsi la compréhension de la dynamique du signal.

3.3 Configuration d'entraînement

Tous les modèles ont été entraînés avec la fonction de perte CrossEntropyLoss. L'optimiseur Adam a été utilisé pour son excellente capacité d'adaptation du taux d'apprentissage.

Nous avons introduit un scheduler pour ajuster dynamiquement le learning rate et utilisé une stratégie d'early stopping pour prévenir le surapprentissage.

4 Résultats

4.1 Classification d'images (CIFAR-10)

4.1.1 Avec Data Augmentation

Nous avons obtenu une accuracy de 83.74% et un F1-score de 83.66%. Les courbes de perte et de précision montrent une bonne stabilité entre l'entraînement et la validation, signe que notre modèle généralise bien.

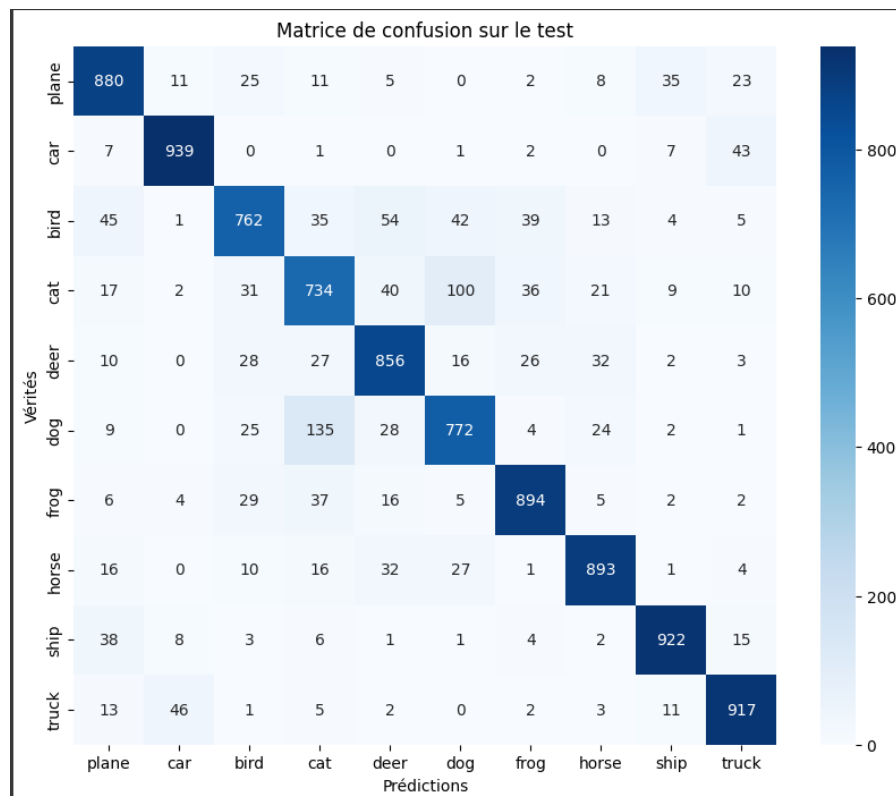


FIGURE 1 – les performances du modèle sur le jeu de test CIFAR-10,



Voici une analyse détaillée de la matrice de confusion obtenue après l'entraînement de notre modèle sur le dataset CIFAR-10. Cette matrice permet de visualiser les performances du modèle en identifiant les classes les plus souvent confondues.

Pour chaque classe, nous avons examiner les confusions fréquentes avec d'autres classes, ce qui nous a aidé à comprendre les limites du modèle et à envisager des améliorations.

TABLE 1 – Analyse détaillée de la matrice de confusion par classe

| Classe | Précision | Erreurs notables | Observations |
|--------|-----------------------------|---|---|
| plane | 880 correctement classifiés | <ul style="list-style-type: none"> — 38 classés comme “ship” — 35 classés comme “ship” — 25 classés comme “bird” — 23 classés comme “truck” | Confusion modérée avec “ship”, possiblement due à des similitudes visuelles . |
| car | 939 correctement classifiés | <ul style="list-style-type: none"> — 43 classés comme “truck” — 7 classés comme “plane” et “ship” | Bonne classification, avec une confusion prévisible avec “truck” tous deux véhicules terrestres. |
| bird | 762 correctement classifiés | <ul style="list-style-type: none"> — 54 classés comme “deer” — 45 classés comme “plane” — 42 classés comme “dog” — 39 classés comme “frog” | Performance plus faible, avec des confusions variées suggérant que les caractéristiques des oiseaux sont plus difficiles à distinguer. |
| cat | 734 correctement classifiés | <ul style="list-style-type: none"> — 100 classés comme “dog” — 40 classés comme “deer” — 36 classés comme “frog” — 31 classés comme “bird” | Performance moyenne avec forte confusion avec “dog”, ce qui est logique (animaux domestiques à quatre pattes). |
| deer | 856 correctement classifiés | <ul style="list-style-type: none"> — 32 classés comme “horse” — 28 classés comme “bird” — 27 classés comme “cat” — 26 classés comme “frog” | Bonne classification, avec quelques confusions avec d’autres animaux, particulièrement “horse”. |
| dog | 772 correctement classifiés | <ul style="list-style-type: none"> — 135 classés comme “cat” — 28 classés comme “deer” — 25 classés comme “bird” — 24 classés comme “horse” | La confusion majeure est avec “cat”, ce qui est prévisible. Cette classe présente le plus grand nombre d’erreurs vers une autre classe spécifique . |
| frog | 894 correctement classifiés | <ul style="list-style-type: none"> — 37 classés comme “cat” — 29 classés comme “bird” — 16 classés comme “deer” | Très bonne classification, avec des erreurs relativement dispersées. |

TABLE 2 – Observations générales sur la matrice de confusion

Observations générales

1. Les classes avec les meilleures performances sont “ship” (922), “truck” (917) et “car” (939).
2. Les classes avec les moins bonnes performances sont “cat” (734) et “bird” (762).
3. Les confusions les plus importantes sont :
 - Dog → Cat (135)
 - Cat → Dog (100)
 - Bird → Deer (54)
 - Car → Truck (43)

4.1.2 Sans Data Augmentation

L’accuracy tombe à 76.66% avec un F1-score de 76.50%. Le modèle tend à surapprendre les données d’entraînement, ce qui montre l’importance de l’augmentation de données.

Conclusion intermédiaire : L’augmentation de données est essentielle pour améliorer la robustesse du modèle sur de nouvelles images.

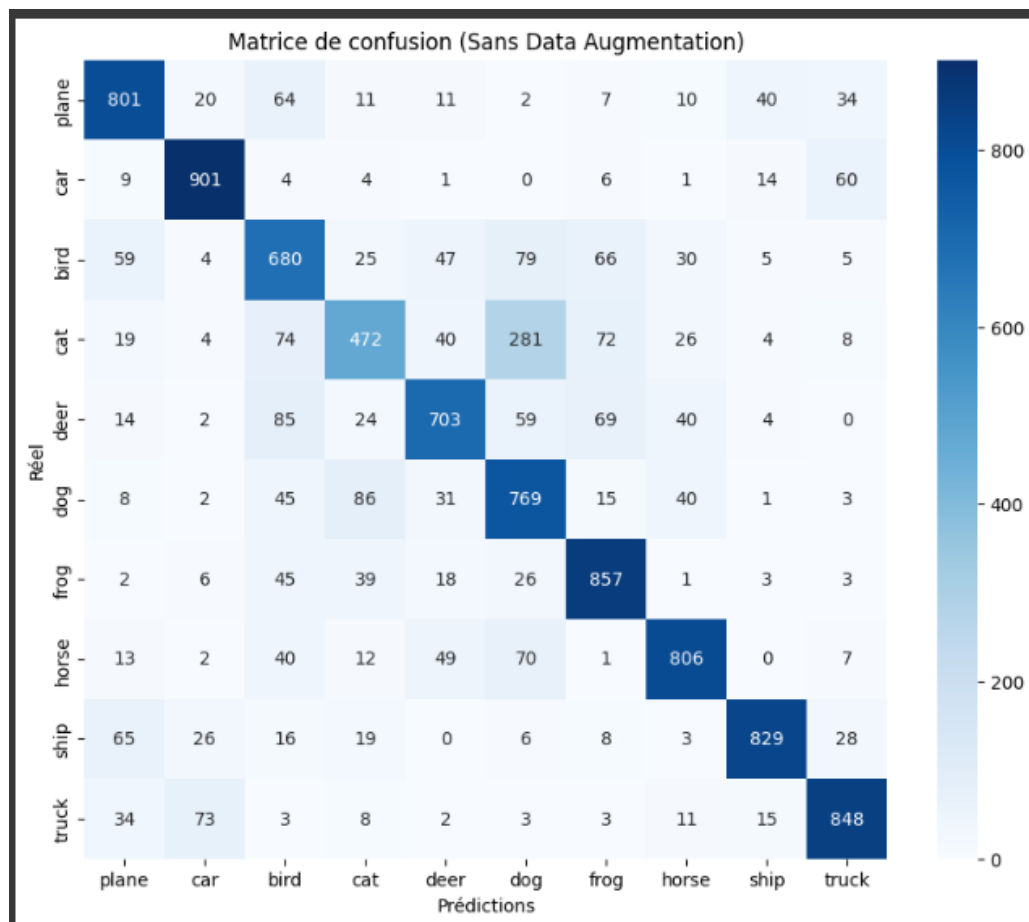


FIGURE 2 – les performances du modèle sur le jeu de test CIFAR-10 Sans Data Augmentation

Les performances du modèle varient considérablement entre les classes. Le modèle performe généralement mieux sur les objets inanimés (véhicules) que sur les animaux :

Meilleures performances : car (901), frog (857), truck (848) et ship (829) Performances les plus faibles : cat (472) et bird (680)

Le problème majeur se situe au niveau de la classe "cat" qui est fréquemment confondue avec "dog" (281 erreurs), ce qui représente près de 38% des échantillons de cette classe. Cette confusion est également présente dans l'autre sens mais de façon moins prononcée (dog → cat : 86 erreurs). D'autres confusions notables incluent :

- Deer → Bird (85)
- Car → Truck (bidirectionnel : 60 et 73)
- Bird → Dog (79)
- Ship → Plane (65)

Ces résultats suggèrent que le modèle a du mal à distinguer certaines caractéristiques similaires entre animaux, en particulier entre chats et chiens. L'absence d'augmentation de données limite probablement la capacité de généralisation du modèle, surtout pour les classes ayant des caractéristiques visuelles variables comme les animaux. Pour améliorer les performances, nous devrions envisager l'augmentation de données particulièrement pour les classes "cat" et "bird", ou explorer des architectures de réseau plus adaptées à la distinction des caractéristiques fines entre espèces animales similaires.

4.1.3 Analyse Visuelle des Courbes

a. Modèle Avec Data Augmentation :

1. Courbe de perte :

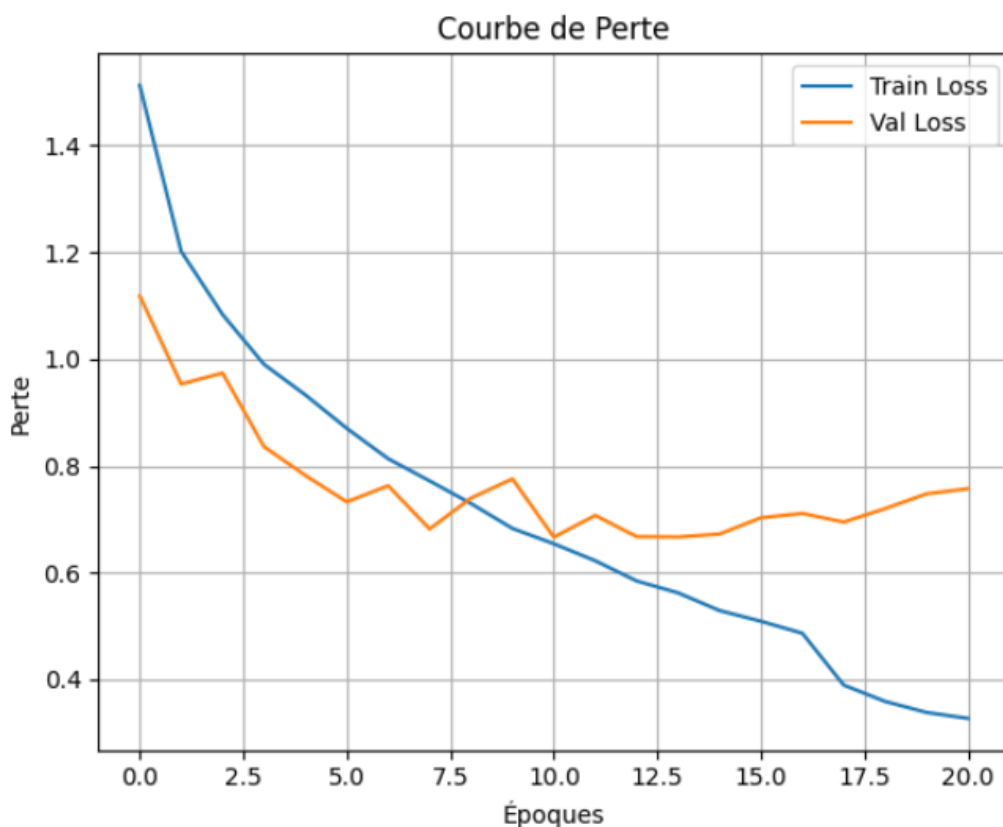


FIGURE 3 – Représentation du Courbe de perte

Baisse progressive et stable de la loss d'entraînement.

La perte de validation suit bien celle de l'entraînement jusqu'à la saturation où le earling stopping est déclenché donc le temps a convergé (vers 71 époques).

2. Courbe de précision :

Hausse constante et régulière de la précision, train et validation, jusqu'à atteindre un plateau élevé (84%-86%).

b. Modèle Sans Data Augmentation :

1. Courbe de perte : Baisse plus rapide de la perte d'entraînement, mais la perte de validation stagne et diverge légèrement, ce qui suggère une tendance de surapprentissage. c'est pourquoi le earling stopping est déclenché plutôt pour éviter le surapprentissage (vers 21 epoques)

2. Courbe de précision :

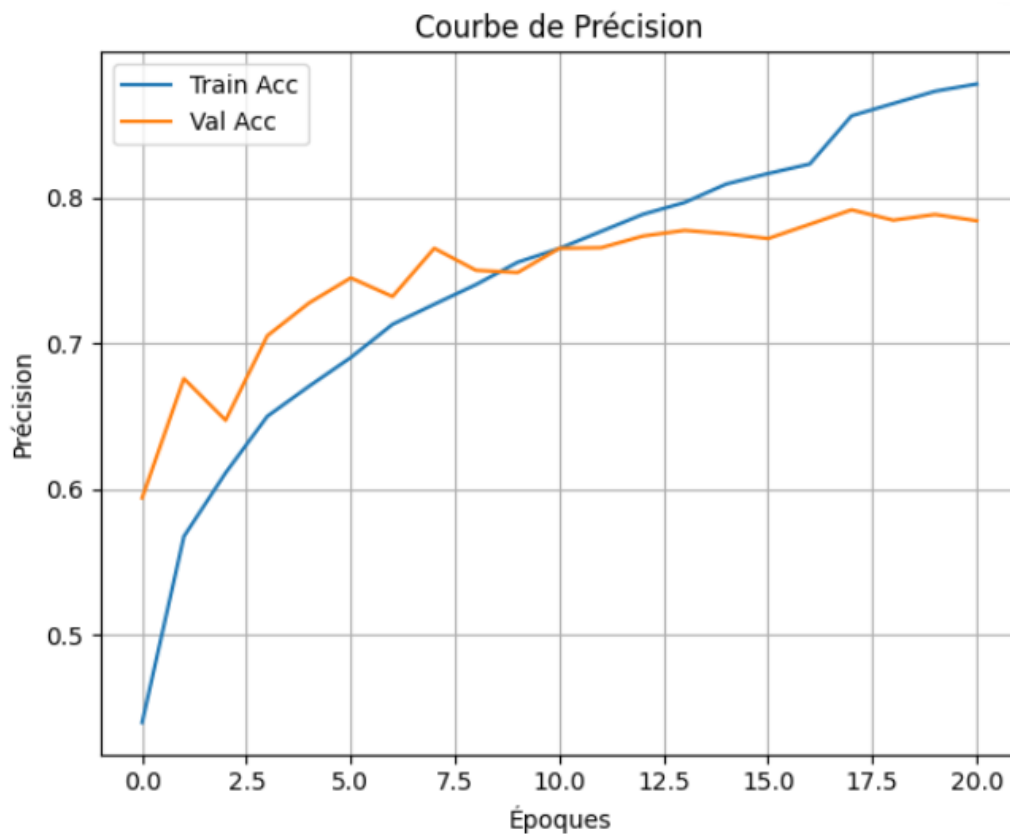


FIGURE 4 – Représentation du Courbe de précision

La précision de validation atteint un plateau plus bas (78-79%) avec un écart croissant avec celle de l'entraînement

4.2 Classification de séries temporelles (ECG5000)

- CNN

Le CNN seul atteint une accuracy de 93.00% avec un F1-score de 91.76%. Il détecte correctement la majorité des motifs présents dans les signaux.

- CNN-LSTM

L'ajout du LSTM amène une légère amélioration avec une accuracy de 93.56% et un F1-score de 92.12%. Le modèle hybride capture mieux les relations temporelles entre les motifs.

4.3 Analyse des hyperparamètres

Nous avons testé différents learning rates, batch sizes et structures de réseaux :

- Learning Rate : 0.001 s'est avéré le plus stable.
- Batch Size : 64 a donné un bon compromis entre qualité et temps d'entraînement.
- Nombre de couches : Trois couches convolutionnelles offrent un bon équilibre entre complexité et performance.
- Hidden Units du LSTM : 64 neurones ont été suffisants pour modéliser efficacement les séquences.

5 Conclusion

Ce travail nous a permis de mettre en pratique des concepts essentiels du deep learning appliqués à des données réelles.

Nous avons constaté que :

1. L'augmentation de données est cruciale pour la classification d'images.
2. Les CNN sont très performants sur des données temporelles simples.
3. L'ajout d'un LSTM améliore la capture des dépendances temporelles.

Perspectives futures :

- Expérimenter des architectures plus profondes comme ResNet ou DenseNet.
- Tester des techniques d'augmentation avancées (CutMix, MixUp).
- Appliquer nos modèles à des données plus complexes comme l'audio ou la vidéo.