

# Rapport de TP 3 : Réseaux de Neurones Convolutifs et Vision par Ordinateur

TIWA TIOTSAP AYMAR ARNOLD

2 janvier 2026

## Introduction

Ce Travail Pratique porte sur l'étude et la mise en œuvre des réseaux de neurones convolutifs (CNN), une classe de modèles de deep learning spécialisée dans le traitement de données spatiales comme les images. L'objectif est de comprendre les mécanismes fondamentaux tels que la convolution et le pooling, d'implémenter des architectures modernes comme les ResNets, et d'explorer des applications avancées telles que la segmentation, la détection d'objets et le transfert de style neuronal.

## 1 Partie 1 : Fondamentaux des CNN

### 1.1 Concepts Théoriques

#### 1.1.1 La Convolution (\*)

La convolution est l'opération de base d'un CNN. Elle consiste à faire glisser une fenêtre de poids appelée **filtre (kernel)** sur l'image d'entrée.

- **Le Filtre** : Il sert d'extracteur de caractéristiques (bords, textures, formes).
- **Le Stride (pas)** : Il définit le décalage du filtre à chaque étape. Un pas plus grand réduit la taille de la carte de caractéristiques en sortie.
- **Objectif principal** : L'objectif d'une couche convolutive est d'apprendre des hiérarchies de caractéristiques spatiales de manière locale et invariante par translation.

### 1.1.2 Le Pooling

Le pooling est une opération d'échantillonnage visant à réduire la dimension spatiale des données.

- **Max Pooling** : Conserve la valeur maximale dans chaque fenêtre. Il est particulièrement efficace pour extraire les caractéristiques les plus saillantes et apporter une robustesse aux petites distorsions.
- **Average Pooling** : Calcule la moyenne des valeurs de la fenêtre, lissant ainsi l'information.
- **Rôle** : Réduire le nombre de paramètres (calculs) et contrôler le sur-apprentissage tout en conservant les informations essentielles.

### 1.1.3 De l'Image à la Classification

Pour passer des cartes de caractéristiques 2D à une prédiction de classe, le réseau utilise des couches denses (Fully Connected) :

1. **Aplatissement (Flattening)** : Les cartes de caractéristiques multidimensionnelles sont converties en un vecteur unidimensionnel (flat vector).
2. **Couches Denses** : Ce vecteur sert d'entrée à un perceptron multicouche qui combine les caractéristiques extraites globalement pour décider de l'étiquette finale (classification).

## 2 Partie 2 : Implémentation du CNN Classique

### 2.1 Préparation des données CIFAR-10

Le dataset CIFAR-10 contient 10 classes d'images couleur de  $32 \times 32$ . Les données ont été normalisées dans l'intervalle  $[0, 1]$  et les étiquettes ont été converties en format One-Hot Encoding pour faciliter l'entraînement.

### 2.2 Architecture du Modèle

Nous avons implémenté un modèle séquentiel comprenant :

- Deux couches de convolution (32 et 64 filtres, taille  $3 \times 3$ , activation ReLU).
- Deux couches de Max Pooling ( $2 \times 2$ ).
- Une couche d'aplatissement (Flatten) suivie d'une couche dense (512 unités) et d'une sortie Softmax.

## 3 Partie 3 : Implémentation du ResNet

### 3.1 Bloc Résiduel

Pour résoudre le problème de disparition du gradient dans les réseaux profonds, nous avons implémenté des blocs résiduels. Ces derniers utilisent une connexion sauteuse (*skip connection*) qui ajoute l'entrée directement à la sortie du chemin de convolution.

### 3.2 Entraînement

Une mini-architecture ResNet a été entraînée sur CIFAR-10 pendant 10 époques. L'ajout de ces connexions permet une meilleure convergence par rapport aux modèles purement séquentiels de profondeur équivalente.

## 4 Partie 4 : Transfert de Style Neuronal

### 4.1 Principe

Le transfert de style sépare le contenu d'une image de la structure stylistique d'une autre en utilisant le modèle VGG16 pré-entraîné sur ImageNet.

- **Image de contenu** : Fournit la structure globale.
- **Image de style** : Fournit les textures et les couleurs artistiques.

### 4.2 Résultats

L'image générée combine les deux aspects. Le processus d'optimisation ajuste les pixels de l'image de sortie pour minimiser simultanément la perte de contenu et la perte de style.

## Conclusion

Ce TP a permis de comprendre les principes fondamentaux de la convolution et du pooling, de construire et d'évaluer un CNN, et de maîtriser des concepts avancés comme les ResNets et le transfert de style avec la bibliothèque Keras.

## **Annexes**

### **Lien Overleaf du rapport pdf**

Lien : <https://www.overleaf.com/read/ghswddyxthhm#5a6a6c>

### **Lien du repository GitHub**

Lien : [https://github.com/piochko/TP3\\_DL.git](https://github.com/piochko/TP3_DL.git)