

UNIVERSITE MOHAMMED V

*Département Physique*  
**Master Informatique et Télécommunications**

# Projet Deep Learning : Classification d'Images par Réseau de Neurones Convolutifs

**Encadrant**

Pr Mahmoudi

**Réalisé par**

**El Hlafi Aya**

**Atbir Rihab**

## Table of Contents

REFERENCES.....	2
INTRODUCTION.....	3
OBJECTIFS.....	4
DEEP LEARNING (APPLICATION SUR IMAGE).....	4
ENVIRONNEMENT DE DEVELOPPEMENT ET JEU DE DONNEES.....	4
CONCEPTION DE L'ARCHITECTURE DU MODELE.....	5
PHASE D'ENTRAINEMENT .....	5
EVALUATION ET RESULTATS .....	5
DEPLOIEMENT ET INTERFACE UTILISATEUR.....	6
CONCLUSION.....	8

## REFERENCES

1) **Kaggle**

<https://www.kaggle.com/datasets/gpiosenska/sports-classification?resource=download>

2) **jupyter:**

<https://jupyter.org/>

3) **streamlit :**

<https://streamlit.io/>

## INTRODUCTION

L'avènement du deep learning a marqué une transformation majeure dans le domaine de l'intelligence artificielle, en particulier pour le traitement et l'analyse d'images. Contrairement aux méthodes classiques reposant sur l'extraction manuelle de caractéristiques visuelles, le deep learning permet aux machines d'apprendre automatiquement des représentations complexes à partir des données. Dans le cadre de ce projet, nous nous sommes intéressés à la classification automatique d'images à l'aide de réseaux de neurones convolutifs (CNN), une architecture profondément adaptée à ce type de tâche. L'objectif principal est de concevoir un modèle capable de reconnaître correctement la catégorie à laquelle appartient une image donnée, tout en assurant une bonne généralisation sur des données inédites.

## 1. Objectifs :

- Développer une architecture CNN adaptée à la classification d'images.
- Optimiser l'entraînement pour obtenir de bonnes performances.
- Évaluer le modèle selon des métriques rigoureuses.
- Proposer une démonstration fonctionnelle de l'application.

## 2. Deep Learning (application sur les images) :

Le **deep learning**, ou apprentissage profond, a révolutionné le traitement d'images en permettant aux machines d'extraire automatiquement des caractéristiques complexes à partir de données visuelles brutes. Contrairement aux méthodes classiques de vision par ordinateur qui nécessitaient des étapes manuelles de détection de caractéristiques (comme les contours ou les formes), les réseaux de neurones convolutifs (CNN) utilisés en deep learning apprennent directement à identifier les motifs discriminants au sein des images. Ces architectures sont capables de reconnaître des objets, des visages, des scènes et bien d'autres éléments avec une précision souvent supérieure à celle de l'humain. Grâce à l'augmentation des puissances de calcul (GPU) et à la disponibilité de grands ensembles de données, le deep learning est devenu la norme pour des tâches telles que la classification, la détection d'objets, la segmentation sémantique ou encore la super-résolution d'images. Cette technologie est aujourd'hui largement utilisée dans des domaines variés comme la médecine (analyse d'IRM), la sécurité (reconnaissance faciale) ou encore l'automobile (véhicules autonomes).

## 3. Environnement de développement et jeu de données :

Le projet a été entièrement développé en langage Python, en s'appuyant sur des bibliothèques spécialisées telles que TensorFlow et Keras pour la conception et l'entraînement des réseaux de neurones. L'environnement de travail a été hébergé sur Google Colab, permettant une utilisation fluide de GPU pour accélérer les calculs. Le jeu de données exploité provient de la plateforme Kaggle. Il s'agit d'un ensemble d'images réparties en plusieurs classes (par exemple : animaux, objets ou vêtements), contenant plusieurs milliers d'exemples annotés.

Avant d'entraîner le modèle, une phase de prétraitement a été effectuée. Chaque image a été redimensionnée à une taille standard (128x128 pixels) et normalisée pour que les valeurs de pixels soient comprises entre 0 et 1. L'ensemble des étiquettes a été encodé de manière catégorielle, et des techniques d'augmentation de données telles que la rotation, le zoom et les inversions horizontales ont été appliquées afin d'enrichir l'ensemble d'apprentissage et d'améliorer la capacité du modèle à généraliser.

#### **4. Conception de l'architecture du modèle :**

Le cœur du système repose sur une architecture de réseau de neurones convolutifs, qui constitue aujourd'hui la norme pour les tâches de reconnaissance d'images. Cette architecture est composée de plusieurs couches de convolution suivies de couches de pooling, qui permettent de capturer les motifs visuels à différents niveaux de complexité tout en réduisant la dimensionnalité des données. Ces couches sont suivies de couches entièrement connectées (fully connected) qui interprètent les caractéristiques extraites et produisent la prédiction finale.

Le modèle que nous avons conçu comprend trois blocs convolutionnels avec activation ReLU et des couches de max pooling, suivis d'un aplatissement (flatten) et de deux couches denses. Une couche dropout est ajoutée avant la sortie pour réduire le risque de surapprentissage. Enfin, la couche de sortie utilise une fonction d'activation softmax qui permet de classer l'image parmi les différentes catégories disponibles. L'entraînement est guidé par la fonction de perte « categorical crossentropy » et optimisé à l'aide de l'algorithme Adam, reconnu pour sa rapidité de convergence.

#### **5. Phase d'entraînement :**

L'entraînement du modèle s'est déroulé sur plusieurs époques, généralement entre 20 et 30, avec une taille de lot (batch size) de 32. Le jeu de données a été divisé en trois sous-ensembles : un ensemble d'apprentissage, un ensemble de validation et un ensemble de test, dans les proportions 70%, 15% et 15% respectivement. Cette séparation permet de suivre l'évolution de la performance du modèle tout en évitant le surapprentissage.

Pendant l'entraînement, la perte sur l'ensemble de validation a été surveillée à l'aide d'un mécanisme d'arrêt anticipé (early stopping), ce qui permet d'interrompre l'apprentissage si le modèle cesse de s'améliorer. Des visualisations des courbes de perte et de précision montrent une bonne stabilité, avec une convergence rapide dès les premières époques. Le modèle atteint un bon compromis entre performance et temps d'apprentissage, ce qui le rend exploitable dans des contextes réels.

#### **6. Évaluation et résultats :**

Une fois le modèle entraîné, il a été testé sur un ensemble de données non vues afin d'évaluer sa capacité à généraliser. Les résultats sont encourageants, avec une précision moyenne dépassant les 90%, un rappel équilibré et un F1-score qui reflète une bonne gestion des faux positifs et faux négatifs. L'analyse de la matrice de confusion permet de mettre en évidence les classes qui posent le plus de difficultés au modèle, souvent à cause de similitudes visuelles fortes entre elles.

Nous avons également comparé notre modèle à des architectures plus avancées comme MobileNetV2 ou ResNet50 pré-entraînées sur ImageNet. Ces modèles atteignent une performance légèrement supérieure, mais au prix d'une complexité accrue et d'un temps d'apprentissage plus long. Ainsi, notre architecture plus simple reste compétitive tout en étant légère.

## **7. Déploiement et interface utilisateur**

Afin de rendre le projet accessible et interactif, une interface web a été développée avec Streamlit. Cette interface permet à l'utilisateur de charger une image locale, d'exécuter la prédiction du modèle et de visualiser le résultat de manière intuitive. L'utilisateur reçoit en retour l'étiquette prédite ainsi qu'un pourcentage de confiance. En option, nous avons inclus une visualisation des activations internes du réseau, permettant de comprendre quelles zones de l'image ont influencé la décision du modèle.

L'interface a été hébergée sur Streamlit Cloud, ce qui permet un accès à distance sans nécessiter d'installation complexe du côté de l'utilisateur. Ce type de déploiement permet d'intégrer rapidement des modèles de deep learning dans des environnements industriels ou éducatifs.





## CONCLUSION

Ce projet a permis de mettre en œuvre de manière concrète les principes du deep learning appliqués à la classification d'images. Le modèle conçu a démontré des performances solides tout en restant simple à mettre en œuvre. Grâce à un prétraitement efficace, une architecture bien pensée et un suivi rigoureux des métriques d'évaluation, nous avons atteint un taux de précision satisfaisant.

Pour l'avenir, plusieurs axes d'amélioration sont envisageables. Nous pourrions expérimenter des techniques de transfert d'apprentissage (transfer learning) plus avancées, utiliser des données plus complexes et augmenter la robustesse du modèle en l'exposant à des exemples plus variés. De plus, une intégration sur mobile avec des versions quantifiées du modèle pourrait permettre son déploiement sur des appareils à faible puissance.