# 📸 Deep Learning: Image Classification using CNN and Transfer Learning

---

## 📌 1. General Description

In this project, we first build a Convolutional Neural Network (CNN) model from scratch to classify images from the CIFAR-10 dataset into predefined categories. Then, we implement a transfer learning approach using the pre-trained MobileNet model. Finally, we compare the performance of the custom CNN and the transfer learning model based on evaluation metrics and analysis.

---

## 📊 2. Project Overview

### 🔍 What does this project do?

- Performs image classification on the CIFAR-10 dataset, which contains 60,000 32x32 color images across 10 different classes (e.g., airplanes, cars, birds, cats, etc.).

- Builds a CNN model from scratch to learn and classify these images.

- Implements transfer learning using the MobileNet pre-trained model to improve classification accuracy.

- Compares and evaluates the performance of both the custom CNN and the MobileNet transfer learning model using key metrics.

### 🎯 What problem does it solve?

- Automates the classification of images into meaningful categories, useful in many fields like computer vision, robotics, and content organization.

- Provides hands-on experience to understand differences between training CNNs from scratch and leveraging pre-trained models.

### 🌍 Potential impact / practical application

- \*\*Educational use:\*\* Offers practical learning in deep learning and transfer learning techniques.

- \*\*Industry relevance:\*\* Demonstrates how transfer learning accelerates model training and improves accuracy on moderate-sized datasets.

- \*\*Research foundation:\*\* Serves as a base for further image classification experiments.

---

## 📁 3. Dataset Description

- Dataset: CIFAR-10

- Source: [https://www.cs.toronto.edu/~kriz/cifar.html](https://www.cs.toronto.edu/~kriz/cifar.html)

- Contains 60,000 color images sized 32x32 pixels, divided into 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck).

- Split into 50,000 training images and 10,000 test images.

- Well-balanced dataset commonly used for benchmarking image classification algorithms.

---

## 🎯 4. Research Goal / ML Objective

- Develop deep learning models to classify CIFAR-10 images into 10 categories.

- Compare a custom CNN trained from scratch with a transfer learning approach using MobileNet.

- Evaluate models using accuracy and loss metrics.

---

## ⚙️ 5. Steps Taken

### 1. Data Preprocessing

- Loaded the CIFAR-10 dataset using standard libraries.

- Normalized image pixel values for better training stability.

- Applied data augmentation to improve generalization.

### 2. Model Building

- Designed a CNN architecture from scratch with convolutional, pooling, and dense layers.

- Implemented transfer learning using MobileNet pre-trained model, fine-tuning top layers.

### 3. Training

- Trained both models using appropriate optimizers and learning rates.

- Used early stopping.

### 4. Evaluation

- Compared models on test accuracy and loss.

- Visualized training history and confusion matrices.

---

## 🔍 6. Key Findings

- The MobileNet transfer learning model outperformed the CNN trained from scratch, achieving higher accuracy with less training time.

- The custom CNN successfully learned image features but required more epochs and parameter tuning.

- Transfer learning showed to be an efficient approach for CIFAR-10.

- Visualizations indicated better generalization for the MobileNet model.

---

## 🧪 7. How to Reproduce the Project

- Python version: 3.11.5

- Main libraries used:

```python

import tensorflow as tf

from tensorflow.keras import datasets, layers, models

import matplotlib.pyplot as plt

import numpy as np

- Files to run:

- `Project\_3\_(CNN\_LN).ipynb` — notebook with CNN and MobileNet transfer learning implementations

- `Project\_3 Presentation.pdf` — project presentation slides

- `requirements.txt` — project dependencies

---

## 🚀 8. Next Steps / Improvements

- Experiment with other pre-trained models and architectures.

- Conduct hyperparameter tuning for improved performance.

- Apply advanced data augmentation and regularization methods.

- Explore deploying the trained model in an application.

---

## 🗂️ 9. Repository Structure

| File/Folder | Description |

|-----------------------------|----------------------------------------------|

| `Project\_3\_(CNN\_LN).ipynb` | Notebook containing CNN and MobileNet models |

| `Project\_3 Presentation.pdf` | Slide presentation of project findings |

| `requirements.txt` | Python dependencies required to run project |