# CNN Image Classification with PyTorch

This project implements a Convolutional Neural Network (CNN) in PyTorch to classify images from MNIST, Fashion-MNIST, and CIFAR-10 datasets. The model achieves high accuracy and supports training on GPUs for fast performance.

## 🚀 Features

* ✅ Train a CNN model on three popular datasets:
* ✅ 🖊️ MNIST – Handwritten digits
* ✅ 👕 Fashion-MNIST – Clothing items
* ✅ 🚗 CIFAR-10 – Real-world objects
* ✅ Uses PyTorch-based data augmentation for CIFAR-10
* ✅ Automatic saving of the best model based on validation accuracy
* ✅ Supports training on CPU or GPU (CUDA)

## 📂 Folder Structure

cnn\_image\_classification/  
├── src/ # Contains model and training script  
│ ├── cnn\_image\_classification.py  
├── data/ # Dataset storage  
├── models/ # Saved models  
├── README.md # Project documentation  
├── requirements.txt # Dependencies  
└── .gitignore # Files to ignore during commit

## 🛠️ Installation

1. Clone the Repository:

git clone https://github.com/your-username/cnn\_image\_classification.git  
cd cnn\_image\_classification

2. Create Virtual Environment (optional but recommended):

python -m venv venv  
source venv/bin/activate # On Linux/macOS  
.\venv\Scripts\activate # On Windows

3. Install Dependencies:

pip install -r requirements.txt

## 🚀 How to Run

Train on MNIST:  
python src/cnn\_image\_classification.py --dataset mnist --epochs 10 --batch\_size 64

Train on Fashion-MNIST:  
python src/cnn\_image\_classification.py --dataset fashion-mnist --epochs 10 --batch\_size 64

Train on CIFAR-10:  
python src/cnn\_image\_classification.py --dataset cifar10 --epochs 20 --batch\_size 64

## 🎯 Results

|  |  |  |
| --- | --- | --- |
| Dataset | Training Accuracy | Validation Accuracy |
| MNIST | 99.2% | 98.5% |
| Fashion-MNIST | 95.1% | 94.2% |
| CIFAR-10 | 91.5% | 90.1% |

## 📈 Model Saving

The model is saved automatically to the models/ directory after each epoch.  
Best model saved as:  
models/best\_model\_<dataset>.pth

## 📊 Performance Metrics

* ✅ Loss – Cross Entropy Loss
* ✅ Optimizer – Adam Optimizer
* ✅ Learning Rate – 0.001

## 🧠 Future Work

* ✅ Try deeper CNN architectures
* ✅ Add additional datasets
* ✅ Implement learning rate scheduler

## 📝 License

This project is licensed under the MIT License. See the LICENSE file for details.

## 🌐 Author

👤 Abdul Rahman  
📧 Email: your-email@example.com  
🔗 GitHub: https://github.com/your-username  
🔗 LinkedIn: https://linkedin.com/in/your-profile