# 📘 Assignment: Comparing NN, CNN & ResNet for Image Classification

## 🎯 Learning Objectives

By the end of this assignment, students should be able to:

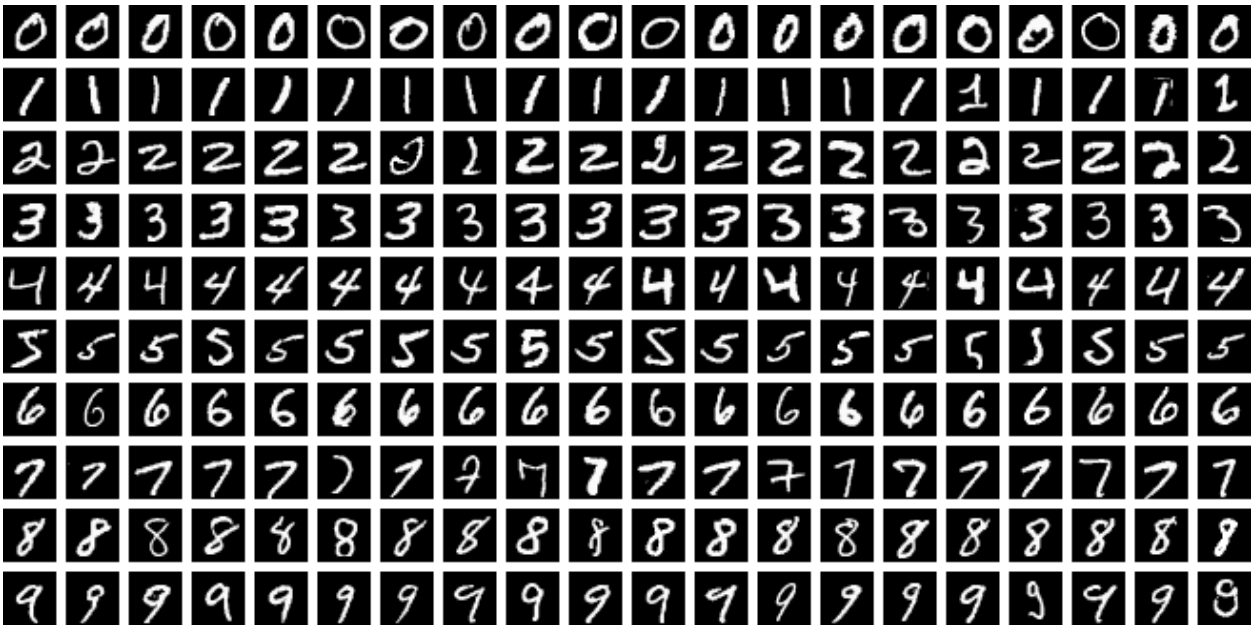
- Understand **why flattening images loses spatial information**
- Implement and compare **Fully Connected NN vs CNN vs ResNet**
- Analyze **accuracy, loss, confusion matrix, and failure cases**
- Apply CV models to a **real-world dataset (traffic lights)**

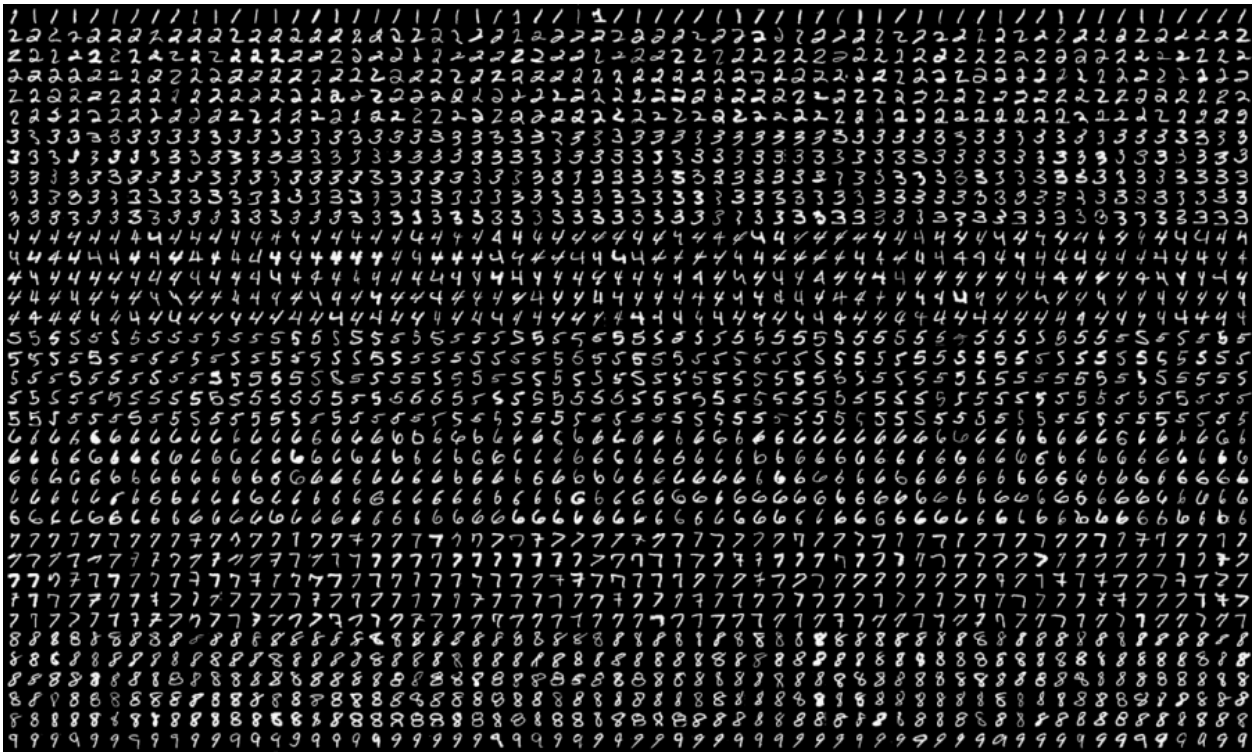
## ⚠️ Academic Integrity

- Plagiarism is strictly prohibited
- You may use libraries **only where allowed**
- All manual implementations must be your own

---

## 🔢 PART 1: MNIST Digit Classification





## Dataset

- MNIST: https://www.kaggle.com/datasets/hojjatk/mnist-dataset

---

## Task 1: Fully Connected Neural Network (Baseline)

- Flatten image (28×28 → 784)
- Simple NN (Dense + ReLU + Softmax)
- Train and evaluate

**Report**

- Accuracy
- Confusion matrix
- 5 misclassified samples

---

## Task 2: CNN-Based Model (example YOLOn11-cls)

- Use convolution + pooling layers
- Train and evaluate

**Report**

- Accuracy
- Comparison with NN
- Confusion matrix

---

## Task 3: ResNet (Use library)

- Use ResNet
- Resize input if required
- Train using transfer learning(post training/fine-tuning)

**Report**

- Accuracy
- Why ResNet performs better

---

## MNIST Comparison Table (Mandatory)

| Model | Train Acc | Val Acc | Notes |
|---|---|---|---|
| NN (Flatten) | | | |
| CNN | | | |
| ResNet | | | |

# 🚦 PART 2: Traffic Light Color Classification

## Dataset (Use any one)

- https://www.kaggle.com/datasets/wjybuqi/traffic-light-detection-dataset/data

## Task 4: CNN on Traffic Lights (Example YOLOn11-cls)

- Resize images
- Train CNN classifier

## Task 5: ResNet on Traffic Lights (use library)

- Use ResNet18/34 (transfer learning)

## Final Comparison Table (Mandatory)

| Dataset | Model | Accuracy | Key Observations |
|---|---|---|---|
| MNIST | NN | | |
| MNIST | CNN | | |
| MNIST | ResNet | | |
| Traffic Light | CNN | | |
| Traffic Light | ResNet | | |

---

# 📄 Submission Format

You must submit a zip file containing:

## 1️⃣ Jupyter Notebook ( `.ipynb` )

- All code cells must be:
  - Executed
  - Outputs visible
- Proper section headings
- No broken cells

## 2️⃣ Report (separate PDF)

The report must include:

- Explanation of each model
- Tables of results
- Graphs and interpretations
- Your **methodology and thought process**

---

# 📝 Evaluation Criteria

| Component | Weight |
|---|---|
| Correct implementation | 40% |
| Experimental comparison | 30% |

| Component | Weight |
| --- | --- |
| Report clarity & explanation | 30% |