📝 Project Report – Neural Network on MNIST Dataset

1. **Introduction**

- Objective: Build and evaluate a neural network to classify handwritten digits (0–9) using the MNIST dataset.
- Tools: TensorFlow/Keras, scikit-learn, and Keras Tuner for hyperparameter optimization.
- Dataset: MNIST — 70,000 grayscale images (28×28 pixels).

---

2. **Data Preparation**

- **Loading Data:** Used `keras.datasets.mnist.load_data()`.
- **Conversion:** Converted to `float32` and added channel dimension → `(n, 28, 28, 1)`.
- **Split:**
  - Training: 51,000
  - Validation: 9,000
  - Test: 10,000
- **Visualization:** Displayed sample digits with their labels.
- **Preprocessing:**
  - Normalized pixel values.
  - Applied data augmentation for robustness.

---

3. **Model Architecture**

- Implemented a **Convolutional Neural Network (CNN)** with:
  - Convolution layers (`Conv2D`)
  - Pooling layers (`MaxPooling2D`)
  - Dropout layers for regularization
  - Fully connected dense layers
  - Softmax output with 10 classes

---

4. **Hyperparameter Tuning**

- Used **Keras Tuner** (RandomSearch) to optimize:
  - Number of filters in Conv2D
  - Kernel sizes
  - Learning rate
  - Dense layer units
- Best hyperparameters were selected based on validation accuracy.

---

5. **Model Training**

- Optimizer: Adam
- Loss Function: Sparse Categorical Crossentropy
- Metrics: Accuracy
- Training strategy:
  - Early stopping
  - Model checkpoints

---

6. **Evaluation**

- **Classification Metrics:**
  - Accuracy, Precision, Recall, F1-score (via `classification_report`)
- **Confusion Matrix:** Showed digit-wise misclassifications.
- **Results:**
  - Training Accuracy: ~99%
  - Validation Accuracy: ~98%
  - Test Accuracy: ~98%

---

7. **Insights & Discussion**

- The CNN achieved strong generalization with minimal overfitting.
- Data augmentation improved robustness.
- Hyperparameter tuning yielded better architecture vs. default setup.
- Misclassifications were mostly between visually similar digits (e.g., 4 vs 9, 3 vs 5).

---

8. **Conclusion**

- Successfully built a high-accuracy digit recognition model.
- Validated the effectiveness of CNNs in image classification tasks.
- Future improvements could include:
  - Testing deeper architectures (e.g., ResNet, EfficientNet).
  - Trying advanced optimizers.
  - Deploying the model in a real-time digit recognition application.