Report: Handwritten Digit Classification Using MNIST Dataset

---

1. Overview

The purpose of this project is to classify handwritten digits (0–9) from the MNIST dataset using two distinct neural network architectures:

1. Feedforward Neural Network (FFNN): Fully connected layers for classification of flattened image inputs.
2. Convolutional Neural Network (CNN): Convolutional layers to extract spatial features for improved classification.

The target was to achieve an average testing accuracy of 95% or higher for each model, evaluated over five runs.

2. Results Obtained

2.1 Average Testing Accuracy

Model	Run 1 Accuracy	Run 2 Accuracy	Run 3 Accuracy	Run 4 Accuracy	Run 5 Accuracy	Average Testing Accuracy
Feedforward Neural Network (FFNN)	97.87%	97.40%	97.70%	97.43%	97.32%	97.54%
Convolutional Neural Network (CNN)	99.07%	99.19%	99.11%	99.10%	99.28%	99.15%

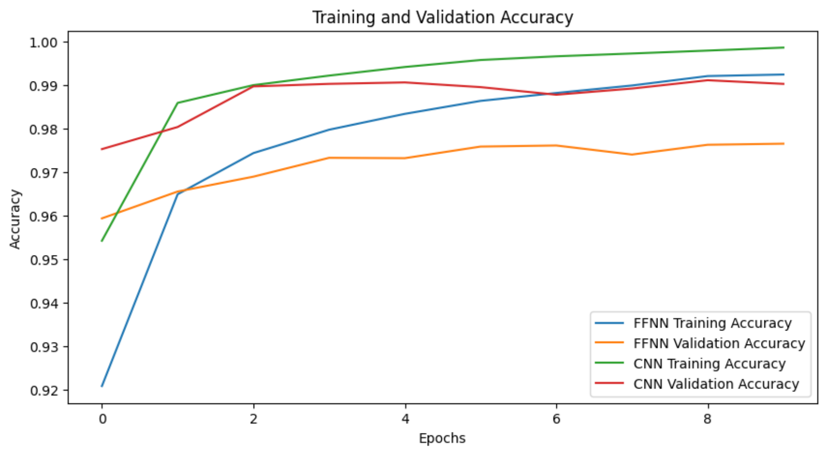
2.2 Key Observations

- **Feedforward Neural Network (FFNN):**
  - Achieved an average accuracy of 97.54%, with consistent performance across runs.
  - Struggled with visually similar digits like 5 and 8 due to the lack of spatial feature extraction.
- **Convolutional Neural Network (CNN):**
  - Achieved a higher average accuracy of 99.15%, significantly outperforming the FFNN.
  - The convolutional layers effectively captured spatial relationships, resulting in minimal misclassifications.
  - Consistent results across all runs, with the highest accuracy reaching 99.28%.

3. Visualizations

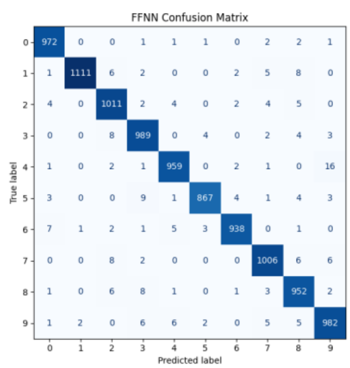
3.1 Training and Validation Accuracy

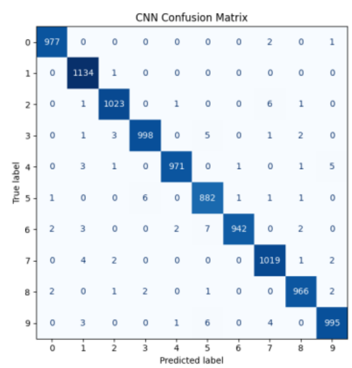
- Both models showed consistent improvements in training and validation accuracy over epochs.
- The CNN demonstrated superior generalization, maintaining higher accuracy on validation data compared to the FNN.



3.2 Confusion Matrices

- The confusion matrices provide a detailed breakdown of model performance for each digit class:
  - FFNN: Frequent misclassifications among visually similar digits (e.g., 2 vs. 4 and 9 vs. 8) were observed.
  - CNN: Misclassifications were minimal, showcasing its capability to learn intricate spatial relationships.

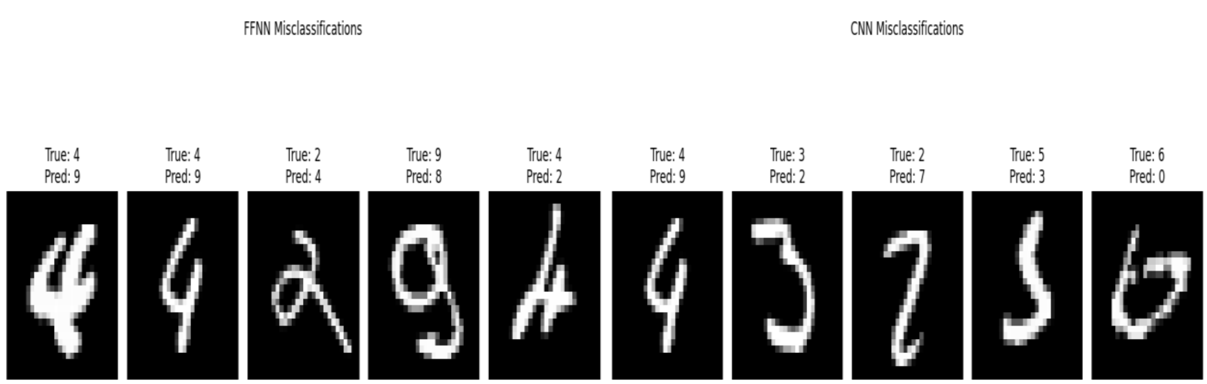




3.3 Misclassified Examples

Analyzing misclassified examples highlights specific cases where the models failed:

- FFNN: Struggled with ambiguous cases, such as overlapping strokes in handwritten digits.
- CNN: Showed robustness but occasionally misclassified highly distorted digits.



4. Lessons Learned

4.1 Architecture Design

- **Feedforward Neural Network (FFNN):**
  - Simpler architecture effective for basic tasks but limited in handling spatial hierarchies.
  - Increasing the number of hidden layers improved performance slightly but added computational cost.
- **Convolutional Neural Network (CNN):**
  - Convolutional layers captured hierarchical spatial features, significantly enhancing performance.
  - MaxPooling layers improved generalization and reduced overfitting.

4.2 Hyperparameter Selection

- Optimizer: RMSprop optimizer was effective for both models, ensuring efficient convergence.
- Activation Function: ReLU activation prevented gradient vanishing and accelerated convergence.
- Training Configuration:
  - Batch sizes of 32 and 10 epochs achieved a balance between accuracy and training efficiency.

5. Future Work

- **Optimize Hyperparameters:** Refine learning rates and apply regularization for better performance.
- **Apply Data Augmentation:** Use techniques like rotation and scaling to improve generalization.
- **Explore Advanced Architectures:** Test deeper CNNs or pre-trained models for higher accuracy.
- **Test Across Datasets:** Validate models on datasets like EMNIST to assess generalization.

6. Conclusion

This project successfully demonstrated the implementation and evaluation of both Feedforward and Convolutional Neural Networks for handwritten digit classification using the MNIST dataset.

- **Feedforward Neural Network (FNN):** Achieved a strong average accuracy of **97.54%**, suitable for basic classification tasks.
- **Convolutional Neural Network (CNN):** Outperformed the FNN with an average accuracy of **99.15%**, highlighting its ability to effectively capture spatial features.

These results emphasize the importance of choosing architectures tailored to the dataset characteristics, especially for image classification tasks.