First Experiment Report: Numerosity-Based Categorization

Introduction

This document summarizes the first experiment in numerosity-based categorization, where a CNN was trained to classify dot patterns into three categories: "Few," "Medium," and "Many." This experiment serves as a baseline for evaluating performance on synthetic data and exploring generalization capabilities.

Objective

The goal was to:

- Develop a neural network capable of categorizing grayscale dot patterns based on numerosity.
- Test the performance using a structured dataset and analyze generalization capabilities.

Experiment Setup

- Dataset: Synthetic grayscale dot patterns (128x128 pixels), categorized as:
  - Few: 1–5 dots.
  - Medium: 6–15 dots.
  - Many: >16 dots.
  - Variations included random dot sizes, positions and density.
- Model: A custom CNN with three residual blocks and a fully connected layer for classification.
- Training:
  - Optimizer: Adam.
  - Learning Rate: 0.001.
  - Loss Funtion: Cross-Entropy.
  - Batch size = 32
  - Early Stopping: Enabled (patience = 3).
  - Number of Epochs: 10
- Hardware: NVIDIA L4 GPU on Google Colab.

Implementation

The model was implemented using PyTorch and trained over 10 epochs with early stopping enabled. The dataset consisted of 5000 samples split into 80% training and 20% testing subsets.

Results

Training loss per Epoch:

Epoch [1/10], Loss: 1.2221

Epoch [2/10], Loss: 0.2356

Epoch [3/10], Loss: 0.2034

Epoch [4/10], Loss: 0.1555

Epoch [5/10], Loss: 0.1355

Epoch [6/10], Loss: 0.0930

Epoch [7/10], Loss: 0.0759

Epoch [8/10], Loss: 0.0533

Epoch [9/10], Loss: 0.0332

Epoch [10/10], Loss: 0.0394

Test Accuracy:

The model achieved an accuracy of 91.70% on the test data, demonstrating strong generalization to unseen dot patterns.

Observations

- The steady decrease in training loss indicates effective learning of numerosity patterns.
- The test accuracy of 91.70% suggests that the model is generalizing well and effectively distinguishing between the three categories.
- Early stopping helped prevent overfitting, though the loss reduction near the end suggests possible model fine-tuning for further improvement.

Next Steps

- Reproducibility: Save raw results, parameters and training metrics in structured files for further analysis and reproducibility.
- Visualization:
    - Generate confusion matrices and misclassification visualizations to identify areas for improvement.
- Generalization: Extend the experiment to other modalities (e.g., shapes, sequences) to test cross-modality learning.
- Scaling: Add regularization techniques (e.g., dropout) and expand the dataset with more complex patterns, such as occluded dots or clustered distributions.