Gana Said

101329317

COMP 3105 – Fall 2025

Assignment 4

Model Architecture

I chose a CNN architecture, as CNNs are specifically designed for image data through local connectivity and parameter sharing. The small size (2 conv layers) was chosen to train quickly within the assignment constraints while still learning meaningful features.

So it is:

1. Fits the dataset size
2. Avoids unnecessary complexity

The architecture consists of two main parts:

1. **Feature Extraction Layers**
    - Conv2d(3, 32, kernel_size=3, padding=1) → ReLU → MaxPool2d(2)
    - Conv2d(32, 64, kernel_size=3, padding=1) → ReLU → MaxPool2d(2)
2. **Classifier Layers**
    - Flatten() → Linear(64 × 32 × 32, 256) → ReLU → Linear(256, 10)

All images are resized to 128×128 and normalized using the standard mean/std transform.

This structure keeps the model light while still giving it enough capacity to learn the dataset.

Training Strategy

The training setup is based on Maximum Likelihood Estimation using cross-entropy loss.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{10} y_{i,c}\,\log p_{i,c}$$

Hyperparameters:

- Optimizer: Adam (learning-rate=0.001) – provides stable learning
- Batch-size: 32
- Epochs: 10 (enough to learn without overfitting)

This choice provides stable convergence without taking too long.

Out-Domain Data Strategy

I intentionally trained only on in-domain images and completely ignored all out-domain data during training.

Why this design choice:

1. High in-domain accuracy:
   The model focuses entirely on learning from the distribution it is supposed to classify. Since it only sees in-domain examples, it learns their patterns well.
2. Poor out-domain accuracy:
   Since the model never encounters out-domain examples during training, it naturally performs badly on them.

The goal is to build a classifier that works well on the target domain but fails on everything else. By not training on out-domain data, the model remains specialized to in-domain images and does exactly that.

Implementation Details

- The dataset loader automatically reads class folders and assigns each class an index.
- Converts all images to RGB for consistency.
- Applies resizing (128×128) and normalization (mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]).
- The same transforms are used for both training and evaluation.

Experimental Results

- Training samples: 683 in-domain images
- Final accuracy on in-domain eval set: ~52%
- Training time: ~4 minutes
- Training was stable and did not show signs of overfitting.

These results show that a simple 2-layer CNN can learn useful features on the in-domain dataset while remaining completely untrained on out-domain data. This design successfully keeps the classifier specialized to in-domain images while preventing high accuracy on out-of-domain data.