System Architecture and Data Flow

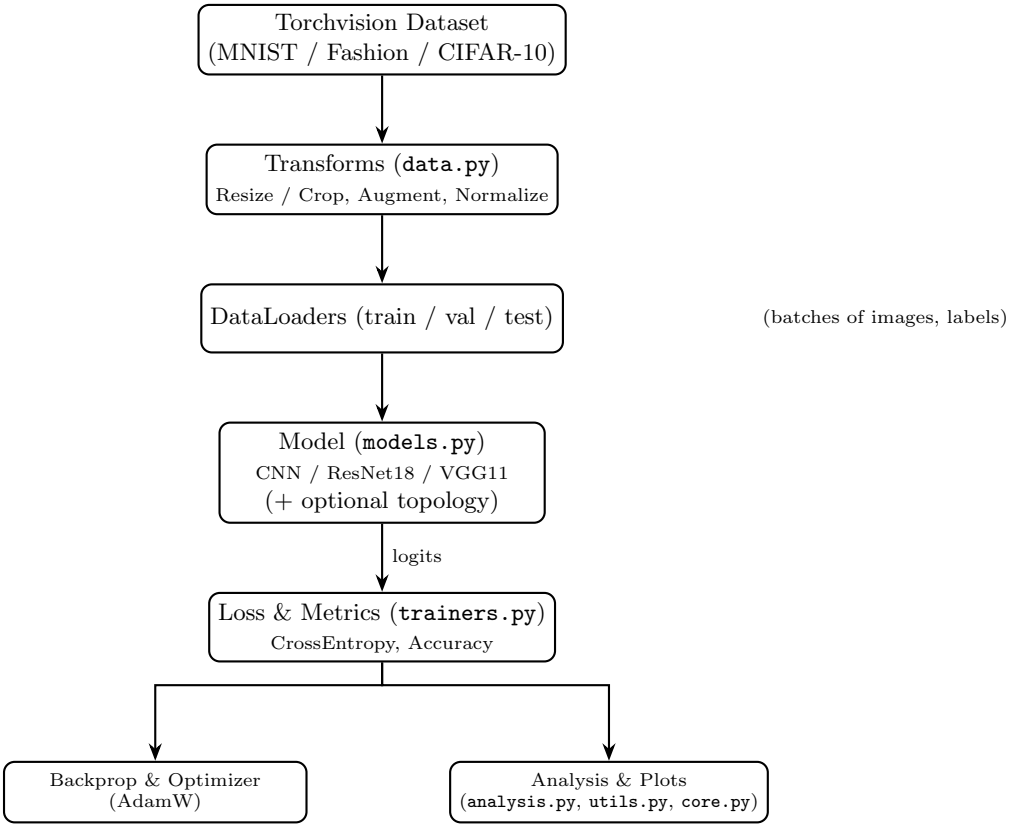
Ravindra Kumar Saini
Missouri State University

November 7, 2025

This document describes the architecture and data flow of the experimental image classification framework. It includes three core diagrams: (1) the high-level training pipeline, (2) the topology-enhanced model flow, and (3) the evaluation and robustness pipeline.

1. High-Level Pipeline

From raw datasets to training, optimization, and analysis.

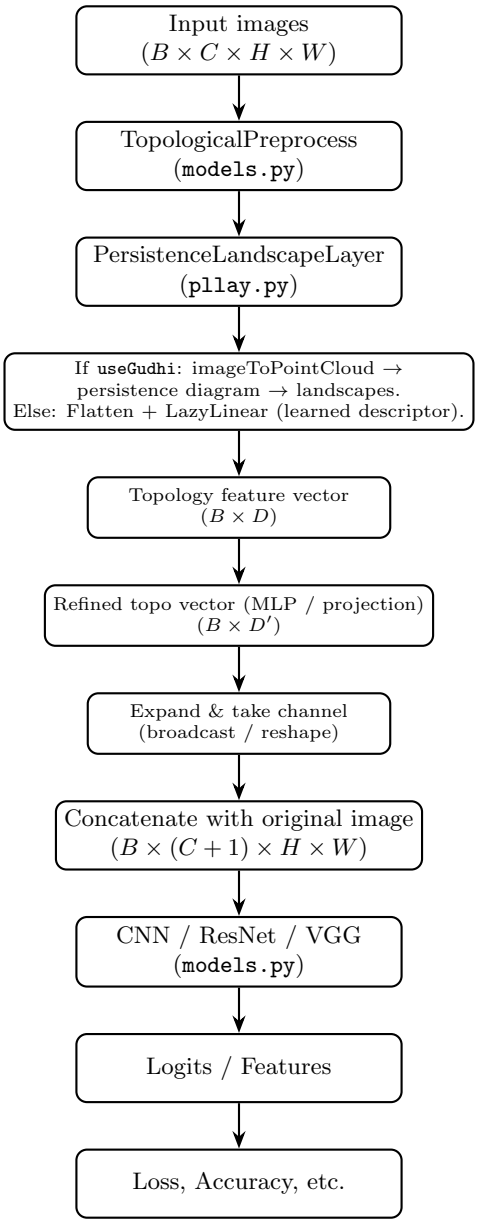


In this pipeline:

- Datasets are loaded from `torchvision` and preprocessed in `data.py`.
- `DataLoader` objects feed batches into the model defined in `models.py`.
- `trainers.py` computes loss and accuracy and performs backpropagation using AdamW.
- `analysis.py`, `utils.py`, and `core.py` log results and create figures such as learning curves and confusion matrices.

2. Topology-Enhanced Model Flow

Process showing how topological features (PLLay) are integrated into the model architecture.



In this topology-enhanced flow:

- `TopologicalPreprocess` calls `PersistenceLandscapeLayer` to compute topological descriptors.
- When `useGudhi` is enabled, `pllay.py` converts images to point clouds, computes persistence diagrams via Gudhi, and derives persistence landscapes.
- Otherwise, a learned global descriptor (Flatten + LazyLinear) is used as a fast approximation.
- The resulting topology vector is projected, reshaped, and inserted as an additional channel before passing into the backbone network.

3. Evaluation and Robustness Flow

In the evaluation and robustness stage:

- `evaluate.py` reloads the trained model checkpoint and the test split.
- `data.py` can add Gaussian noise via `applyEvalNoise` to test robustness.
- `trainers.py` computes loss and accuracy on noisy or clean data.
- `utils.py` and `analysis.py` generate confusion matrices, robustness curves, and save all metrics as JSON and plots.

Using saved checkpoints for evaluation, noise robustness, and visualization.