Hybrid CNN-Transformer Waste Classifier

Report - Assignment 2

Daniele Gotti - 1079011

December 2025

1 Introduction

In this project, I developed a multi-class image classifier to categorize waste objects into nine distinct classes (e.g., Plastic, Paper, Metal). The core challenge was to implement a hybrid architecture combining a Convolutional Neural Network (CNN) backbone for feature extraction with a Transformer Encoder for global context processing. Given a training dataset of images, I employed cross-validation strategies to optimize the model's hyperparameters and ensure generalization. The final model was trained on the full dataset and uploaded to HuggingFace for deployment.

2 Methodology

2.1 Data Preprocessing and Augmentation

The dataset consists of images divided into 9 classes. To enable hyperparameter tuning, I split the provided data into a training set and a validation set (typically utilizing an 80/20 split). All images were resized to 224×224 pixels and normalized using ImageNet statistics. To combat overfitting, a significant challenge observed in early experiments, I implemented an aggressive data augmentation pipeline for the training phase, which included:

- Random horizontal flips (p=0.5);
- Random rotations (up to 15 degrees);
- Color jittering (brightness and contrast).

2.2 Hybrid Model Architecture

I implemented a class named `ResnetTransformer` that fuses a CNN with a Transformer:

1. Backbone: I used a ResNet-18, removing the last two layers. This outputs a feature map of shape $512 \times 7 \times 7$.
2. Tokenization: the spatial features are flattened into 49 tokens (7×7). A linear projection layer maps these from 512 dimensions to a customizable embedding dimension.
3. Transformer: I prepended a learnable CLS token and added positional embeddings. These inputs are processed by a standard PyTorch `TransformerEncoder`.
4. Classification: the final logits are computed by passing the output of the CLS token through a LayerNorm followed by a Linear head.

3 Hyperparameter Tuning and Analysis

I conducted a systematic series of experiments (labeled Experiment 0 through 6, followed by 3 Final Tests) to identify the optimal configuration. The accompanying notebook documents the code and plots for each step.

3.1 Baseline and Complexity Analysis

I established a baseline (Exp 0) using a frozen backbone, 128 embedding dimension, and 1 transformer layer.

- Baseline results: the model reached a plateau around 70% accuracy with signs of overfitting.
- Increasing capacity (Exp 1-3): I attempted to increase the model capacity by doubling the layers (to 2 or 4) and the embedding dimension (to 256). Contrary to expectations, increasing complexity worsened overfitting, widening the gap between training and validation performance without improving validation accuracy.
- Attention heads (Exp 4): increasing the number of heads from 4 to 8 had a negligible impact on performance.

3.2 Impact of Regularization and Fine-Tuning

Since architectural changes alone did not solve the overfitting, I shifted focus to data and training dynamics.

- Data augmentation (Exp 5): introducing the augmentation pipeline described in Section 2.1 successfully eliminated overfitting, causing the training and validation curves to overlap. However, the accuracy remained limited by the frozen backbone.
- Unfreezing the backbone (Exp 6): unfreezing the ResNet-18 weights allowed the CNN to learn features specific to the waste dataset. This was the most critical improvement, significantly boosting accuracy.

3.3 Refining the Training Process

The final phase focused on balancing stability and learning speed.

- Learning Rate: I tested lower learning rates ($1e-5$ vs $5e-5$). While $1e-5$ produced extremely stable curves, it led to underfitting within the epoch limit. A value of $3e-5$ provided the best balance.
- Epochs: I extended training to 20 epochs to allow the fine-tuned model to converge.

4 Selection of the Optimal Model

Based on the analysis, I selected the configuration from "Final Test 3" as the optimal model. Although it showed a slight gap between training and validation curves compared to the most conservative tests, it achieved the highest absolute validation accuracy (∼87%), maximizing predictive power.

The winning configuration is:

- Backbone: ResNet-18 (unfrozen).
- Transformer: 1 layer, 8 heads, 256 embedding dimension.

- Data augmentation: active (flip, rotation, jitter).
- Optimizer: Adam with learning rate $3e-5$ (and exponential scheduler).
- Training epochs: 20.

5 Conclusion

I successfully implemented a hybrid CNN-Transformer classifier for waste detection. My analysis demonstrated that for this specific dataset, simply increasing the transformer's size was ineffective. Instead, the combination of aggressive data augmentation and fine-tuning the convolutional backbone proved to be the decisive factors in reaching high performance.

For the final deployment, I retrained the optimal model on the complete dataset to maximize its knowledge base. The final model weights were saved and uploaded to the HuggingFace repository `daniele-gotti/Waste_Classifier`.

6 Contributions

I completed this project individually and was responsible for all aspects of the work, including code implementation, experimental analysis, and report writing.

7 AI Usage Acknowledgment

I acknowledge that no AI software was used for generating the implementation code or for writing the core content of this report. I used an AI-based tool solely for the purpose of refining English grammar and ensuring a scientific writing style.