

Anomaly Detection with MVTec Dataset

Overview

The code is designed to train two anomaly detection models (EfficientAd and PatchCore) on the MVTec dataset. It is structured with a main execution block and several helper functions to facilitate training, evaluation, and inference.

Data Preparation

- The implementation uses the MVTec dataset, focusing specifically on the **leather** category.
- The image size is set to **256x256 pixels** to standardize input dimensions.
- A **batch size of 1** is used during training to optimize memory usage.

Models

- **EfficientAd**: A recent and efficient anomaly detection model known for its performance and speed.
- **PatchCore**: Another widely used anomaly detection model, recognized for its effectiveness in handling various types of anomalies.

Training Process

- The **Engine** class from the `anomalib` library is utilized for managing the training process.
- Each model is trained for **5 epochs** (specified by `max_epochs=5`).
- After training, the models are saved in a specified results directory for future use and inference.

Evaluation

- The model is evaluated on the test set, utilizing features from the Engine class.
- While the calculation of ROC AUC score and Average Precision is part of the evaluation process, this functionality is not explicitly implemented in the current code.

Inference

- The code provides functionality to save the trained model for later inference and deployment.

Code Structure

- **Data Preprocessing**: Handled by the MVTec data module, which loads and preprocesses images for training.
- **Model Initialization**: Both EfficientAd and PatchCore models are instantiated for training.
- **Training Loop**: The main training loop is managed by the Engine class, which handles the forward and backward passes and optimization steps.
- **Model Saving**: Trained model weights are saved to disk for later use.

