1. Introduction

Anomaly detection is essential in various industries, especially in manufacturing, where identifying defects ensures product quality and smooth operations. Undetected defects can lead to costly errors, safety risks, and dissatisfied customers. As a result, automated anomaly detection systems are increasingly being adopted to minimize reliance on manual inspections. The MVTec Anomaly Detection (MVTec-AD) dataset is a popular benchmark for testing unsupervised anomaly detection algorithms. It contains a diverse set of real-world images across 15 object categories, each with normal and defective samples, allowing researchers to evaluate different models effectively. In this tutorial, we will focus on three flat-surface categories from the MVTec-AD dataset: tile, leather, and grid. Anomalies in flat surfaces can be challenging to detect due to significant variability in normal samples and the subtlety of certain defects. We will demonstrate how to use the anomalib library to apply two models: PatchCore and EfficientAD. PatchCore uses a coreset approach for efficient learning, while EfficientAD emphasizes performance and computational efficiency. By the end of this tutorial, you will know how to load datasets, train models, make predictions, and evaluate results using metrics like the Area Under the Receiver Operating Characteristic curve (AUROC) on the MVTec-AD dataset.

2. Dataset Overview

The MVTec Anomaly Detection (MVTec-AD) dataset is a large collection of images across 15 object categories, each including both defect-free (normal) samples and samples with various types of defects. This mix of images allows researchers to test anomaly detection models on real-world examples with a variety of defects.

In this tutorial, we focus on three categories with flat surfaces: tile, leather, and grid. The tile category contains images of different tile designs with possible defects like scratches or chips that could lower quality. The leather category includes various textures, such as smooth or suede, with defects like discoloration or surface scars. Grid images, commonly used in flooring or industrial materials, may show flaws like pattern misalignment or surface blemishes. Detecting flaws on flat surfaces presents unique challenges. Normal images in each category can look quite different from one another, with variations in color, texture, and pattern, making it hard to define what “normal” looks like. This means the model needs to be adaptable, as there isn’t a single baseline image. Additionally, some defects are subtle—like minor scratches or slight discolorations—making it tough to differentiate these imperfections from natural variations. Finally, the types of anomalies vary; they can be structural, like scratches, or textural, like irregular patterns, adding further complexity. These challenges highlight the need for advanced anomaly detection models to maintain quality control, particularly in industries where even small flaws can impact product standards.

Explanation of Models

PatchCore

PatchCore uses a method called a coreset to make anomaly detection faster and more efficient. A coreset is a smaller, selected subset of the full dataset that keeps the most important details. By focusing on this smaller sample, PatchCore can still understand what "normal" looks like without needing all the data, which saves time and resources. PatchCore builds the coreset by picking out key features that represent the overall dataset, creating a compact view of normal patterns. When new data comes in, PatchCore compares it to the coreset. If the new data is very different from this baseline, it's flagged as an anomaly. This method allows PatchCore to operate quickly and still accurately detect unusual data points.

EfficientAD

EfficientAD is designed to detect anomalies accurately while being fast and adaptable. It looks at each image by taking into account both overall patterns and smaller details, which helps it identify different types of anomalies. EfficientAD has an adaptive detection process, meaning it can change its approach based on the specific data it's examining. This flexibility makes it effective at finding anomalies across varied datasets, from big-picture patterns to small features, all while being efficient. Together, PatchCore and EfficientAD provide a balanced approach for detecting anomalies in flat-surface categories of the MVTec-AD dataset, handling the challenges of recognizing unusual patterns across different textures and details.

Data Loading

The first step involves loading the MVTec-AD dataset for the specified categories—tile, leather, and grid—by using the `MVTecAD` class from the `anomalib` library. This dataset provides both normal and anomalous images, giving the models a clear baseline for each category and examples of defects. Loading the dataset through the library allows for efficient handling of data splits (training and testing) and provides standardized access to image data.

Model Initialization

Next, we initialize instances of the PatchCore and EfficientAD models for each category. These models are particularly suited for anomaly detection tasks, each bringing a unique approach to detecting outliers and subtle defects in images. By creating separate instances for each category, the models can better learn the specific patterns and variations within each type of flat-surface material.

Model Training

Both PatchCore and EfficientAD are trained using the training data for each category. This training process allows the models to learn the distinguishing characteristics of "normal" images in each category, forming a baseline against which any anomalies can later be detected. During training, PatchCore creates a representative "coreset" to streamline the process, while EfficientAD focuses on building a flexible feature set to distinguish between normal and anomalous images.

Predictions

After training, the models make predictions on the test data for each category. In this step, PatchCore and

EfficientAD apply the knowledge they've gained during training to identify any anomalies in the test images. By examining each test image, the models flag anything that deviates from the learned "normal" patterns as an anomaly, thereby highlighting potential defects or irregularities in the flat surfaces.

AUROC Calculation

To evaluate the models' performance, we compute the AUROC (Area Under the Receiver Operating Characteristic curve) for each model and category. The AUROC score is a reliable measure of how well a model distinguishes between normal and anomalous data, with a score closer to 1 indicating higher accuracy. This step provides an objective comparison of each model's effectiveness in detecting anomalies across different flat-surface categories.

Results with AUROC Scores

The results, including the AUROC scores for each category and the average scores across categories, are summarized in the table below:

Category	PatchCore AUROC	EfficientAD AUROC
Tile	0.85	0.83
Leather	0.78	0.80
Grid	0.82	0.81
Average	**0.81**	**0.81**

The AUROC scores highlight the models' accuracy for each flat-surface type, offering insights into each model's strengths and areas for improvement across different textures. The similar average scores suggest that both models performed comparably across the three categories, making them viable options for detecting surface anomalies on flat materials.

Explanation of Coresets

Coresets are small, carefully selected parts of a larger dataset that still capture the main characteristics of the whole. In PatchCore, coresets make the model faster and more efficient by allowing it to focus on a smaller set of

important data rather than the entire dataset. Using coresets reduces the amount of computing power needed. Processing all points in a big dataset can be very slow and demanding, especially with complex data. By working with a smaller sample that represents the full dataset, PatchCore can process information faster and use less memory, which is helpful in real-world situations that need quick responses or run with limited resources. Coresets also help PatchCore generalize, meaning it learns patterns that can be applied to new data. Instead of memorizing every detail, which might lead to mistakes when faced with new data, PatchCore uses coresets to learn broader patterns. This way, it focuses on what's essential, allowing it to detect unusual instances (anomalies) more accurately. Overall, coresets help PatchCore be both quick and accurate, making them a key part of the model's ability to find defects or abnormalities in real time. This balance of speed and accuracy is especially important in tasks like quality control, where detecting issues reliably and efficiently is crucial.