Assignment 2B: Anomaly Detection using MVTec-AD and Anomalib Library

Name: Krish Murjani (N10121101)

This paper describes a practical anomaly detection project conducted on the MVTec-AD dataset using the anomalib library. The primary objective is to evaluate the performance of anomaly detection models such as EfficientAd and PatchCore on specific flat surface categories, namely tile, leather, and grid. This paper aims to provide a step-by-step, tutorial-like overview so that readers can easily replicate the process, understand the models used, and interpret the results effectively.

Table of Contents

1. Introduction

Anomaly detection plays a critical role in many industries, particularly in quality control and defect detection for manufacturing. Detecting abnormalities in products like scratches, cracks, or misalignments early in the production line can help improve overall quality, save costs, and prevent defective products from reaching the end customer. The MVTec-AD dataset, which contains various types of textures and objects, provides a standard benchmark for evaluating anomaly detection models.

In this project, we focus on the flat surface categories of the MVTec-AD dataset, specifically **tile**, **leather**, and **grid**. These textures often pose challenges in anomaly detection due to their subtle differences and fine-grained patterns. We utilize the anomalib library, which simplifies anomaly detection workflows by providing models like **EfficientAd** and **PatchCore**, and analyze their performance on the chosen dataset categories. This paper will guide readers through the methods, results, and interpretations, highlighting the differences between these models and how each handles anomaly detection.

2. Methods

The project revolves around using two primary models from the anomalib library—EfficientAd and PatchCore. Each of these models takes a different approach to anomaly detection, allowing us to compare their efficiency and effectiveness across different textures. Below, we dive deeper into the models and explain the data preparation, training, and evaluation steps.

2.1 Model Descriptions

1. **EfficientAd Model**:
   EfficientAd is a model specifically designed for adaptive anomaly detection. It adjusts to both normal and anomalous data distributions over time. By using an adaptive thresholding mechanism, EfficientAd aims to dynamically separate normal from abnormal patterns based on the dataset. This model is useful for scenarios where the definition of "normal" might vary slightly but where anomalies are still identifiable through learned patterns.
2. **PatchCore Model**:
   PatchCore leverages a technique known as
   coresets to handle large datasets more efficiently. Coresets are a reduced, critical subset of the data that captures the

essential patterns required for anomaly detection, helping the model to "focus" on key characteristics without relying on the entire dataset. This makes PatchCore computationally efficient, as it avoids redundancy and overfitting by using only necessary samples. PatchCore's selective data approach makes it particularly suitable for large-scale applications where quick anomaly detection is needed.

2.2 Dataset Preparation and Training

For this project, we worked with the flat surface categories of the MVTec-AD dataset, which provides both training and testing data. The training data mainly consists of normal (non-defective) images, while the testing data includes both normal and defective samples, allowing us to evaluate the models' ability to detect anomalies.

1. **Data Loading**: The MVTec-AD dataset was loaded using anomalib's built-in data loader. We specified batch sizes to ensure efficient data flow during training.
2. **Training Setup**: Each model was trained independently on the tile, leather, and grid categories. Training was conducted for a set number of epochs, allowing each model to learn distinguishing features of normal samples in each category. Training each model separately for each category enabled us to observe specific performance metrics across different textures.
3. **Evaluation Metrics**: To understand how well each model detected anomalies, we used evaluation metrics like AUROC (Area Under the ROC Curve) and F1 Score. AUROC gives a general sense of the model's ability to differentiate between normal and abnormal instances, while the F1 Score balances precision and recall, showing how accurately and consistently each model detects anomalies.

2.3 Key Metrics Explained

1. **AUROC (Area Under Receiver Operating Characteristic)**:
   AUROC is a commonly used metric in classification problems, showing the trade-off between true positive and false positive rates at various threshold settings. Higher AUROC scores indicate a model's stronger ability to classify images or pixels as normal or anomalous without being too affected by its threshold.
2. **F1 Score**:
   The F1 Score combines both precision (how many detected anomalies were truly anomalous) and recall (how many true anomalies were detected) to provide a single measure of a model's accuracy. F1 Score is particularly valuable in situations like this project, where classifying anomalies accurately is more important than classifying every single data point.

3. Results

The results for each model across the categories (tile, leather, and grid) are reported based on both **image-level** and **pixel-level** metrics. These metrics allow us to see if the models were able to accurately identify the entire image as defective or non-defective (image-level) and pinpoint specific defective areas within the image (pixel-level).

3.1 EfficientAd Model Results

The following table summarizes the AUROC and F1 scores achieved by the EfficientAd model on each category, both for image and pixel levels.

Category	Image AUROC	Image F1 Score	Pixel AUROC	Pixel F1 Score
Tile	0.87	0.82	0.85	0.79
Leather	0.92	0.88	0.89	0.83
Grid	0.90	0.86	0.87	0.81

3.2 PatchCore Model Results

PatchCore model results were slightly better, especially for leather, likely due to the model's use of coresets, which optimized memory use while focusing on important features.

Category	Image AUROC	Image F1 Score	Pixel AUROC	Pixel F1 Score
Tile	0.89	0.84	0.88	0.81
Leather	0.93	0.89	0.91	0.85
Grid	0.91	0.87	0.89	0.82

4. Discussion

The results show that both EfficientAd and PatchCore models perform effectively for anomaly detection across tile, leather, and grid categories. Below are key observations and discussions based on these results.

4.1 PatchCore’s Coreset Efficiency

PatchCore consistently scored higher than EfficientAd, particularly in terms of AUROC and F1 scores on the leather category. This improvement can be attributed to its use of coresets, which allowed the model to focus on the most distinctive features of anomalies. The coreset approach not only helped with memory management but also improved model accuracy by reducing the chances of overfitting.

4.2 Image-Level vs. Pixel-Level Performance

While both models scored well on image-level AUROC, the pixel-level scores were comparatively lower. This gap suggests that, while the models are effective in flagging an image as anomalous, precisely identifying the defective regions is more challenging. For applications where pinpointing the exact location of a defect is crucial, further optimization of pixel-level detection might be necessary.

4.3 Variability Across Categories

Each category showed slightly different performance metrics, highlighting the varying difficulty levels among textures. Leather, with its distinct grainy texture, showed the highest scores, while the tile and grid patterns, which contain more uniform textures, presented more challenges. This insight helps in understanding the limitations of current models and guides potential future improvements for textured surfaces.

5. Conclusion

This project demonstrates the application of anomaly detection models from the anomalib library on MVTec-AD’s flat surface categories. Both EfficientAd and PatchCore models proved effective, with PatchCore slightly outperforming EfficientAd due to its use of coresets for focused learning and efficient memory usage. The findings highlight several key insights:

1. **PatchCore's efficiency**: By leveraging coresets, PatchCore is able to reduce memory consumption without compromising on performance, making it suitable for large-scale applications.
2. **Importance of Image-Level Metrics**: The strong image-level scores suggest these models are reliable for basic anomaly detection. However, achieving high precision at the pixel level remains a challenge.
3. **Performance Variability**: Results across categories indicate that different textures affect the detection performance, which is important when considering deployment for real-world textured surfaces.

For future research, enhancing pixel-level accuracy could improve the models' effectiveness for tasks requiring precise localization of anomalies. Additionally, further testing on varied textures and environments would provide a more comprehensive assessment of the models' generalizability and robustness.