Anomaly Detection in Textured Surfaces using Pre-trained CNN Features and Autoencoders

1 Introduction

Anomaly detection is a critical task in various industrial applications, such as quality control in manufacturing, where identifying defects or irregularities in products is essential. Traditional methods often rely on handcrafted features and rule-based systems, which may not generalize well to complex textures or unseen anomalies.

This project explores an anomaly detection framework that leverages pre-trained convolutional neural network (CNN) features from ResNet-50, combined with a simple autoencoder architecture. The goal is to detect anomalies in textured surfaces, specifically using the MVTec dataset.

2 Dataset

MVTec Anomaly detection dataset is used, as it is the benchmark in this field.the training set only consists of defect-free images(OK), the testing set is a mix of normal and defective samples. Effectively, the autoencoder only learns what a normal(OK) image looks like, anything out of it, is likely to be an anomaly.

3 Preprocessing

3.1 Image Transformation

To prepare the images for input into the neural network, the following preprocessing steps are applied:

- **Resize**: Images are resized to 224×224 pixels to match the input size expected by ResNet-50.

- **Normalization**: Pixel values are scaled to $[0, 1]$ by converting them to tensors of type `torch.FloatTensor`.

4 Model Architecture

4.1 Feature Extractor

A pre-trained ResNet-50 model is used as a feature extractor. Specifically, features are extracted from the output of Layer 3, balancing the trade-off between high-level semantic information and spatial resolution.

4.1.1 Mathematical Representation

Let x be the input image, and $f_{\text{ResNet}}^{(L3)}(x)$ be the features extracted from Layer 3:

$$F = f_{\text{ResNet}}^{(L3)}(x) \in R^{N \times 1024 \times H \times W}$$

4.2 Simple Autoencoder

An autoencoder is designed to reconstruct the extracted features. It consists of:

- **Encoder**: Compresses features to a bottleneck representation.
- **Decoder**: Reconstructs the original features.

4.2.1 Mathematical Representation

$$\hat{F} = D(E(F))$$

where E is the encoder function and D is the decoder function.

5 Training Procedure

5.1 Loss Function

The Mean Squared Error (MSE) loss is used to measure reconstruction error:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|F_i - \hat{F}_i\|_2^2$$

5.2 Optimization Algorithm

The Adam optimizer is used with a learning rate of 0.001.

6 Anomaly Detection

6.1 Reconstruction Error Map

For each input, a reconstruction error map is computed:

$$E_{\text{map}} = \left(F - \hat{F}\right)^2$$

6.2 Decision Function

An anomaly score is computed by averaging the top 10 highest values in the error map:

$$\text{Anomaly Score} = \frac{1}{10} \sum_{k=1}^{10} \text{sorted}(E_{\text{map}})_k$$

6.3 Threshold Calculation

The threshold is determined as:

$$\text{Threshold} = \mu_{\text{train}} + 3\sigma_{\text{train}}$$

7 Results and Analysis

7.1 Visualizations

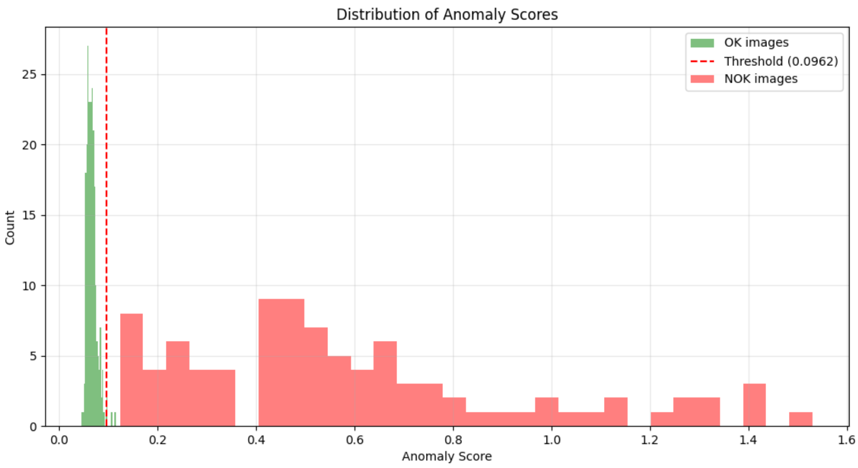

Figure 1: Distribution of Anomaly Scores for Normal and Anomalous Images.

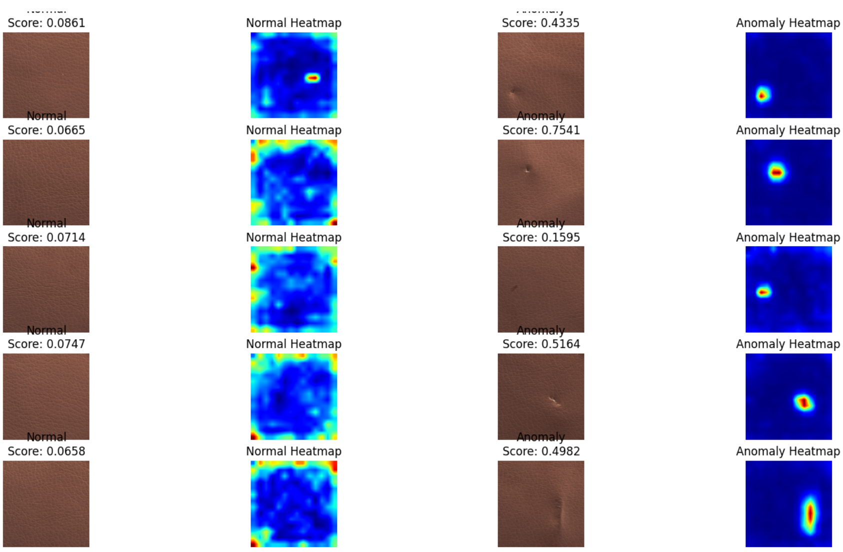

Figure 2: Heatmap visualization of sample predictions.

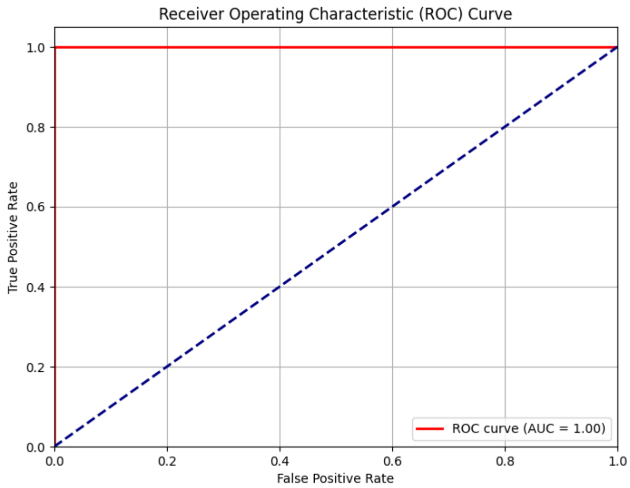

Figure 3: ROC Curve and AUC (1.000).

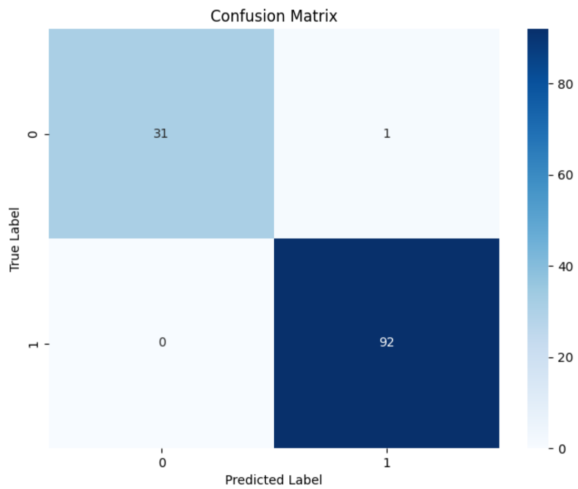

Figure 4: Confusion Matrix.

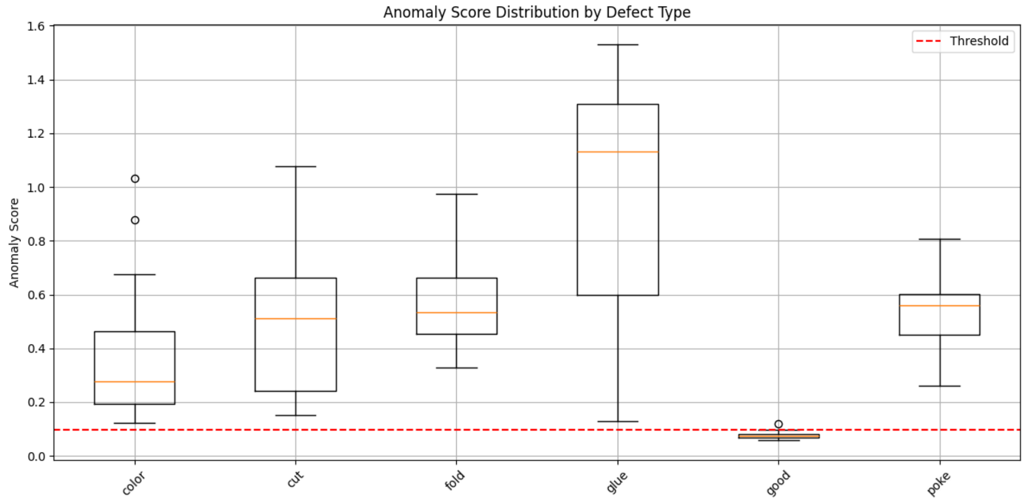

Figure 5: Anomaly Score Distribution by Defect Type.

8 Conclusion

The combination of pre-trained ResNet-50 features and a simple autoencoder effectively detects anomalies in textured surfaces with high accuracy, precision, recall, and an AUC score of 1.000.