Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows

Subject:
Deep learning

Submitted to:
Marco Cristani & M. Uzair Khan

Submitted by:
Narges Samai

Contents

Objective 2
Challenges with Existing Techniques.. 2
Method 2
Experiment 3
Dataset 3
Training.. 3
Evaluation 4
Detection Results.. 4
conclusion 5
Reference... 5

Objective

In industrial manufacturing processes, it is crucial to constantly monitor and improve product quality because even small defects can cause big problems. However, detecting these defects is challenging because the types of defects are often unknown beforehand, and defects occur so infrequently that there are typically no examples of defective items to train on. Furthermore, new types of defects can arise unexpectedly due to unforeseen manufacturing anomalies. Traditional supervised machine learning methods are ineffective in this context because they require examples of defective items, which are typically unavailable. This paper proposes a robust, semi-supervised defect detection solution called DifferNet, which uses a technique called normalizing flows to detect and locate defects without needing many examples of defective products. This approach aims to provide a reliable way to find subtle defects in manufacturing data, helping to maintain high product quality even when examples of defects are scarce. This method improves on existing approaches and performs well on challenging datasets like MVTec AD and Magnetic Tile Defects.

Challenges with Existing Techniques

Existing methods for anomaly detection can be roughly divided into approaches based on generative models and pre-trained networks. Generative models often struggle with accurately capturing high-frequency structures and can be overly influenced by specific features of the training data, making them less effective for diverse and subtle defects. Pre-trained networks, on the other hand, are limited by their inability to handle unseen defect types robustly and are often costly and inefficient due to the need for extensive overlapping patches and limited flexibility. In contrast, DifferNet addresses these issues by leveraging normalizing flows to effectively model image feature distributions with minimal training data, allowing it to detect a wider range of defects and localize anomalies more precisely.

Method

The method presented in the paper employs normalizing flows to detect anomalies by estimating the density of image features extracted from anomaly-free training images. Initially, a pre-trained feature extractor maps images to a feature space. These features are then transformed into a latent space using a normalizing flow, which provides a well-defined probability distribution. Likelihoods for image samples are derived from this distribution, with normal images assigned higher likelihoods and anomalous images assigned lower likelihoods. The approach involves maximizing likelihoods across multiple image transformations to enhance robustness during both training and inference. Additionally, the feature extractor is designed to capture multi-scale features, which helps in creating a more detailed and descriptive representation of the image data, thereby improving defect detection accuracy. Figure 1 outlines the pipeline of the method.

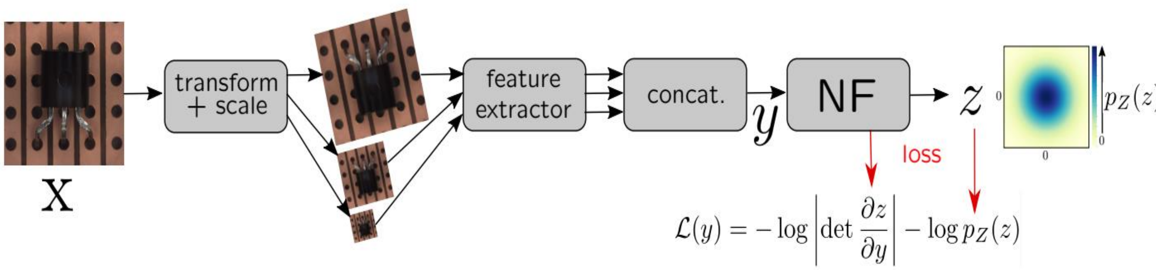

Figure 1 Overview of our pipeline: Multiple scales of a transformed input image are fed into a feature extractor. The distribution of its concatenated outputs is captured by transforming it via a normalizing flow (NF) into a normal distribution by maximum likelihood training.

Experiment

To implement the experiments described in the paper, we follow these steps:

1. Clone the DifferNet repository onto Google Colab.
2. Customize the code to match the specific requirements of the dataset.
3. Train the model for each class for 5 epochs and then evaluate the performance across all classes.

Dataset

The BTAD (beanTech Anomaly Detection) dataset is a real-world industrial anomaly dataset. The dataset contains a total of 2830 real-world images of 3 industrial products showcasing body and surface defects.

(https://www.kaggle.com/datasets/thtuan/btad-beantech-anomaly-detection)

Training

Initially, the convolutional layers of AlexNet are used as the feature extractor, with global average pooling applied to each feature map at every scale.
(https://download.pytorch.org/models/alexnet-owt-7be5be79.pth)

In the training phase, the DifferNet model is initialized along with the optimizer. Training proceeds over multiple epochs, with the model being updated using batches from the training data loader. During each epoch, the model computes the loss and updates gradients accordingly. Performance is evaluated on a test set, with losses and anomaly scores recorded throughout the process. After training, the model may be saved, and gradient maps can be exported if configured.

Training parameters:

- Epochs = 5
- sub epochs = 8
- input size = 448x448

Train epoch 0
Epoch: 0.0 train loss: -1.5922
Epoch: 0.1 train loss: 664.9653
Epoch: 0.2 train loss: -2.6735
Epoch: 0.3 train loss: -2.8362
Epoch: 0.4 train loss: -2.7487
Epoch: 0.5 train loss: -3.0618
Epoch: 0.6 train loss: -3.1848
Epoch: 0.7 train loss: -3.2701

Evaluation

The evaluation phase involves assessing the performance of the trained model on an image dataset. A pre-trained model is loaded, and images are processed through the model to compute anomaly scores. This involves applying either fixed or random image transformations. For each image, anomaly scores are calculated based on the mean squared activation, which reflects the extent of anomaly present in the image.

Detection Results

The detection performance of the model across three classes is evaluated using the Area Under the Receiver Operating Characteristic curve (AUROC). The results are as follows:

Dataset	Class1	Class2	Class3
AUROC	0.9553	0.7953	0.9240

These results demonstrate strong performance, with high AUROC values indicating effective detection of anomalies across the different classes.

conclusion

DifferNet effectively detects anomalies in manufacturing processes, with strong performance in Classes 1 and 3, as indicated by high AUROC scores. While the model performs well even with limited training data, the lower score for Class 2 suggests that further improvements are needed for certain defect types. Overall, DifferNet shows promise for robust anomaly detection and has potential for further refinement and application across various datasets.

Reference

Dataset:

https://www.kaggle.com/datasets/thtuan/btad-beantech-anomaly-detection

Model:

https://github.com/marco-rudolph/differnet