ReConPatch : self-supervision for patch-based anomaly detection

Computer Vision

Master's Degree in Artificial Intelligence and Robotics

Dario Loi **Elena Muiá** **Martina Doku**

1940849 1938610 1938629

Academic Year 2023/2024

SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

Introduction

Anomaly detection is a crucial task in many real-world applications, such as medical imaging, surveillance, and quality control.

In this work, we implement a **self-supervised** method for anomaly detection on industrial images, based on contrastive learning and patch-based representations.

We propose a **generalized approach** to anomaly detection that can be applied to different types of images and anomalies.

Objective

The goal of this work is to develop a method for anomaly detection that can be applied to a wide range of industrial images and anomalies.

We aim to create a **generalized** model that is trained in a **self-supervised** manner, on all the normal images of the dataset simultaneusly unlike the traditional per-category training.

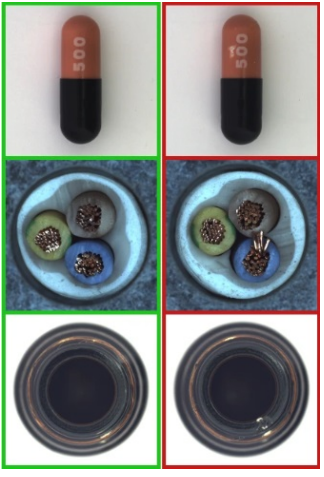
The model should be able to detect anomalies in images of unseen categories and anomalies, without the need for any labeled data.

Table of Contents

Data

We worked on the **MVTec AD** dataset by MVTec Software GmbH (2019), which is a widely used benchmark for anomaly detection in industrial images.

The dataset contains images of **15 object categories**, each with a set of normal and anomalous images. The anomalies are introduced by adding various types of defects to the normal images, such as scratches, holes, and dents.



Data

Each of the defective images has a corresponding **ground truth mask**, which indicates the location of the anomaly in the image.

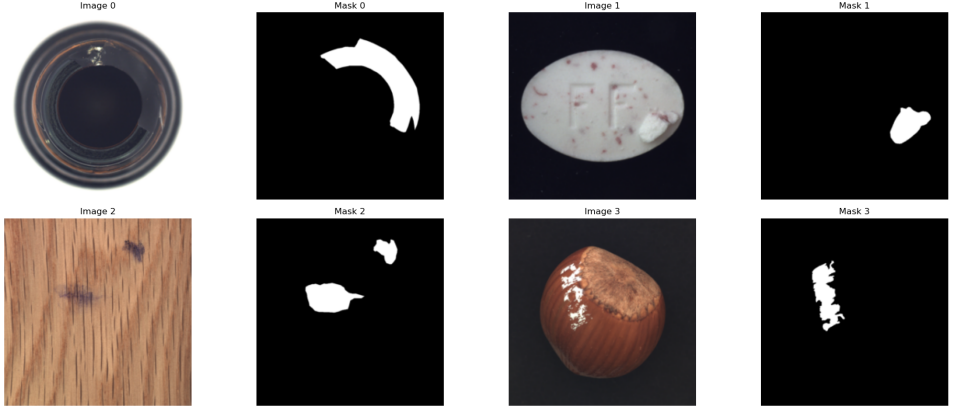


Table of Contents

3 Method

Self-Supervised Learning

3 Method

- A type of learning where the system learns to predict often through pretext tasks that don't require labeled data.
- The model is trained on the normal images of the dataset, without any information about the anomalies.
- The model learns to encode the normal images in a way that captures the underlying structure of the data.
- During inference, the model can detect anomalies by identifying images that deviate from the learned Embedding.

Patch-based Anomaly Detection

- The **loss** $\mathcal{L}$ on which the model is trained is based on the similarity between patches of the same image.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij} \delta_{ij}^2 + (1 - \omega_{ij}) \max(0, m - \delta_{ij})^2 \tag{1}$$

where:

- δ_{ij} denote the relative distance between embedding vectors in a mini-batch
- ω_{ij} indicates the similarity between two patch-level features
- m is the margin parameter

Similarity

3 Method

- The **similarity** ω between two patches is computed using two different measures: contextual similarity and pairwise similarity.
- The final similarity is computed as a weighted average of the two measures:

$$\omega_{ij} = \alpha\omega_{ij}^{contextual} + (1-\alpha)\omega_{ij}^{pairwise} \tag{2}$$

where α is a hyperparameter that controls the contribution of each measure.

Similarity Measure

3 Method

- **contextual similarity**

$$\omega_{ij}^{contextual} = \frac{1}{2}(\hat{\omega_{ij}}^{contextual} + \hat{\omega_{ji}}^{contextual}) \tag{3}$$

$$\hat{\omega}_{ij}^{contextual} = \begin{cases} \frac{|\mathcal{N}_k(i) \cap \mathcal{N}_k(j)|}{|\mathcal{N}_k(i)|} & \text{if } j \in \mathcal{N}_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

- **pairwise similarity**

$$\omega_{ij}^{pairwise} = \exp(-\frac{||\tilde{z}_i - \tilde{z}_j||_2^2}{\sigma}) \tag{5}$$

where:

— $\tilde{z}_i$ is the embedding of patch i

— σ is a hyperparameter that controls the smoothness of the similarity function.

Patch-Level Features Extraction

3 Method

- The model uses a **pre-trained** WideResNet-50 to extract the patch-level features from the input images.
- The features are extracted from the **last convolutional layer** of the network.
- The features are then **flattened** and passed through a double **representation learning network** to train the feature embeddings.

Double network structure

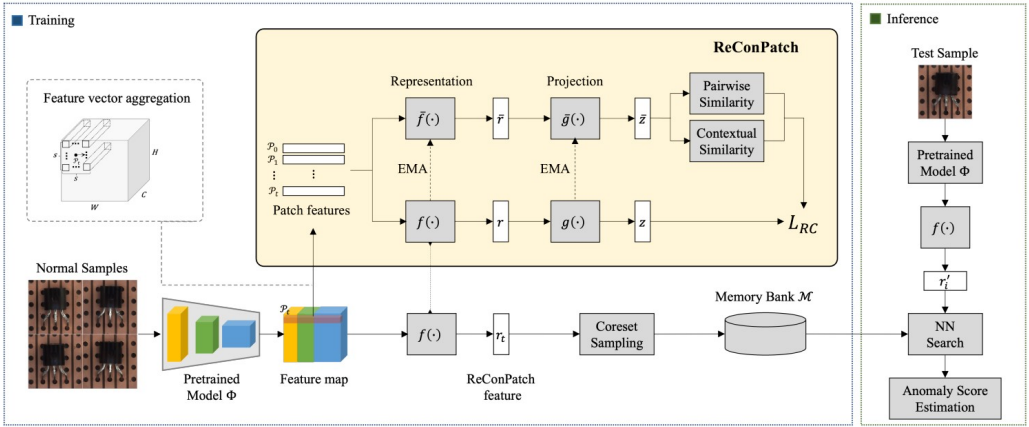
3 Method

- The model consists of an online network and a target network.
- **Online network**:
  - The online network itakes as input the patch-level features.
  - The online netwrok processes the features through a representation and projection layers and sends the result as input of the loss function.
  - The online network is updated during training using the contrastive loss.
- **Target network**:
  - The target network is a copy of the online network that is used to compute the similarity between patches.
  - The target network sends the result of the similarity computation as input of the loss function.
  - The target network is updated using an exponential moving average of the online network's parameters.

Table of Contents

Model

4 Experiments



Training

4 Experiments

- The model is trained on the normal images of the dataset using the contrastive loss.
- The **optimizer** used is Adam with a learning rate of 10^{-4}.
- The **batch size** is set to 32
- The model is trained for a total of **100 epochs**, with an **early stopping criterion** based on the validation loss, with a patience of 10 epochs.

Inference Pipeline

- During inference, the model is used to encode the test images and compute the anomaly score for each patch.
- After training a **coreset** of features r^* is selected using a greedy approximation algorith and stored in a **Memory Bank** $\mathcal{M}$.
- The **anomaly score** for each patch is computed as:

$$s_i = (1 - \frac{e^{s'_i}}{\sum_{r' \in \mathcal{N}_{b(r^*)}} e^{\mathcal{D}(z_i, r')}}) \cdot \mathcal{D}(z_i, r^*) \tag{6}$$

where:

- s'_i is the similarity between the patch and its nearest neighbor in the Memory Bank
- $\mathcal{N}_{b(r^*)}$ is the set of neighbors of r^* in the Memory Bank
- $\mathcal{D}$ is the distance function

Metrics

4 Experiments

- The performance of the model is evaluated using the **Area Under the Receiver Operating Characteristic curve (AUC-ROC)**

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(FPR)d(\text{FPR}) \tag{7}$$

  where:
  - TPR is the True Positive Rate
  - FPR is the False Positive Rate
- The AUC-ROC was measured both on the **patch-level** and the **image-level**.

Results

4 Experiments

We evaluated the model on the **MVTec AD** dataset using the **AUROC** metric. We obtained an **AUROC** of **0.60** on the **patch-level** and **0.70** on the **image-level**.

Embedding

4 Experiments

Table of Contents

Conclusion

5 Conclusion

We have presented a self-supervised method for anomaly detection based on patch-based representations. The model is trained on the normal images of the dataset, and is able to detect anomalies in images of all categories and anomalies, obtaining an increased generality compared to traditional methods.

Future Work

5 Conclusion

There are still several aspects of the model that can be improved, to increase the performance and generality of the model:

- **Hyperparameter tuning**: The model has several hyperparameters that can be tuned to improve performance.
- **Heavier training**: The model can be trained for longer or with a larger batch size to improve performance.
- **Anomaly scores**: The anomaly scores can be further refined to improve the detection of anomalies.

References

ReConPatch : self-supervision for patch-based anomaly detection

Thank you

for listening!
Any questions?