UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Automated visual inspection in the industrial setup via deep learning

Author:
Pavlo SEMCHYSHYN

Supervisor:
Ph.D. Taras Firman

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences and Information Technologies
Faculty of Applied Sciences

Lviv 2023

Declaration of Authorship

I, Pavlo SEMCHYSHYN, declare that this thesis titled, "Automated visual inspection in the industrial setup via deep learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"For me context is the key – from that comes the understanding of everything."

Kenneth Noland

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Automated visual inspection in the industrial setup via deep learning

by Pavlo SEMCHYSHYN

Abstract

With the rise of Industry 4.0, much attention is attracted to the field of automated visual inspection. Automation of the quality check in the production environment can reduce labor costs significantly, therefore, especially with the rise of deep learning-based algorithms, anomaly detection became one of the most researched topics in a machine learning community. Visual anomaly detection aims to detect inconsistencies in image data, which can be classified as anomalies. This can be used in many areas apart from manufacturing, including the detection of abnormal areas in medical imaging, surface inspection, or photo editing. This task becomes quite common when we have access only to normal samples as anomalies are rare compared to normal data and are usually hard to collect. Therefore, visual anomaly detection is usually solved in an unsupervised setting, where we take advantage of only normal data. One of the approaches to solving visual anomaly detection is image reconstruction. Recently, diffusion models became state-of-the-art in the image generation task, being especially prominent in terms of image quality and diversity of the generated samples. In this study, we leverage diffusion models to the task of visual anomaly detection in a manufacturing setting, show its strengths and weaknesses as well as compare it with other existing methods and provide extensive benchmarks on the subject.

Acknowledgements

I would like to express my gratitude to Taras Firman, who initially made me familiar with the topic of research, shared his ideas and materials as well as provided me with guidance during the whole work. I would also like to express my thanks to the ELEKS DSO team for providing me with help and various resources and to my teachers who enabled me to complete with work by providing feedback and support.

Contents

Declaration of Authorship i

Abstract iii

Acknowledgements iv

1 Introduction 1

2 Related works 3
2.1 Visual Anomaly Detection . . . 3
2.1.1 Image reconstruction . . . 3
Autoencoders . . . 4
Generative Adversarial Networks . . . 5
2.1.2 Feature-based approaches . . . 5
One-class classification . . . 5
Density estimation . . . 6
Student-Teacher . . . 6
Memory bank . . . 7
2.2 Diffusion models . . . 7
2.2.1 Background . . . 8
Forward diffusion process . . . 8
Backward diffusion process . . . 9
Training objective . . . 9
2.2.2 Improvements in DDPMs . . . 10
Reduction in sampling costs . . . 10
The increase in likelihood . . . 10
Conditional DDPMs . . . 11
Unsupervised VAD with DDPMs . . . 11

3 Methodology 12
3.1 Reconstruction with DDPM using a self-aware sampling technique . . . 13
3.1.1 Reconstruction procedure . . . 13
3.1.2 Inpainting mask creation . . . 15
3.1.3 Anomaly score . . . 16
Multi-Scale Reconstruction Error Anomaly Map . . . 16
Learned Anomaly Map . . . 16
3.1.4 Backbone model . . . 17

4 Experiments and Results 18
4.1 Dataset . . . 18
4.2 Implementation details . . . 20
4.3 Training details . . . 20
4.4 Evaluation . . . 20

4.5 Experiments with the proposed method 20

5 Conclusion and Future work **24**
5.1 Conclusion 24
5.2 Future work 24

A **25**

Bibliography **26**

List of Figures

2.1 Trade-offs between different classes of generative models 8

3.1 The examples of reconstruction of image (A) when the number of noise steps t is too low (20), which results in the anomaly being reconstructed well (B), and when t is too large, which results in losing too much local information (C) 12
3.2 The strategy for generating anomalies proposed in DRAEM [64] 17

4.1 Image examples (A) and class distributions (B) in the MVTec AD dataset 19
4.2 Our experiments with resampling: both reconstructed images are obtained by setting the number of noise steps t to 80. The number of resampling steps is 1 in (C) and 5 in (D) 20
4.3 The anomaly masks used for inpainting depending on hyperparameter p for abnormal image (A) 21
4.4 The results of our method: (A) - original image, (B) - reconstructed image, (C) - inpainting mask m, (D) - anomaly map obtained with multi-scale reconstruction error map, (E) - anomaly map obtained with segmentation network trained on artificial anomalies and their reconstruction, (F) - Ground-truth anomaly map 23

List of Tables

4.1 Anomaly Detection Performance (AUCROC) on MVTec: comparison with GAN-based methods 21
4.2 Anomaly Detection Performance (AUCROC) on MVTec AD: comparison with recent SOTA methods 22

List of Abbreviations

AD	**A**nomaly **D**etection
VAD	**V**isual **A**nomaly **D**etection
SOTA	**S**tate **O**f **T**he **A**rt
AE	**A**utoencoder
NF	**N**ormalizing **F**low
KDE	**K**ernel **D**ensity **E**stimation
KNN	**K** **N**earest **N**eighbours
FID	**F**rechet **I**nception **D**istance
DDPM	**D**enoising **D**iffusion **P**robabilistc **M**odel
DDIM	**D**enoising **D**iffusion **I**mplicit **M**odel
GB	**G**igabytes
VAD	**V**isual **A**nomaly **D**etection
SSIM	**S**tructure **S**imilarity **I**ndex **M**easure

Dedicated to my family

Chapter 1

Introduction

We live in the Industry 4.0 era, which is about making use of information technology to promote industrial transformation. And the core of the fourth industrial revolution is intelligent manufacturing, which can be currently considered a trend in building a manufacturing system. Intelligent manufacturing implies the production process is smooth and information-based so that no production inputs are wasted and no additional costs are spent. For example, intelligent manufacturing can be used for capturing deviations in the production process. If any are present, the system will process them immediately, and the producer can make quick adjustments in no time. Such systems rely on artificial intelligence, which is nowadays the core of intelligent manufacturing. By incorporating it into the production process, the requirements for human resources, such as technical experts and quality monitoring personnel, can be significantly reduced, therefore the original labor force can be saved.

The traditional approaches for performing an inspection in the production processes are based on subjective judgments of manual human evaluators, which does not guarantee the accuracy of identifying deviations. With the rise of computer image processing technology, many computational algorithms were proposed and implemented for discovering abnormalities in the manufacturing process, which rapidly formed a whole new field in machine learning known as AD (anomaly detection) with the end goal of automating the visual inspection process.

AD is a very important task in machine learning. First of all, it deals with the assumption of an open dynamic system, where the learning algorithms are expected to infer abnormalities from normal data. Anomaly detection algorithms characterize and model available normal data and then develop anomaly detectors to check for abnormal regions in the newly observed data. When the data samples are represented as images, then we deal with VAD (visual anomaly detection).

VAD can be applied to many possible scenarios. Except for the automation of inspection in the manufacturing process, VAD can be also applied in the field of medical analysis, for instance, in the detection of abnormalities in MRI or in the field of intelligent security by inspecting the video recordings for any anomalous events. Given its significance, a lot of research was attracted to the field of VAD. Worth noticing, due to the limited number of anomaly samples and the labor-intensive labeling process, detailed anomaly samples are not available for training. As a result, most recent studies on visual anomaly detection have been performed without prior information about the anomaly, i.e., unsupervised paradigm.

A new spike in research of VAD was caused in line with the development of deep learning algorithms. That is mainly due to the fact that classical algorithms are unable to handle high-dimensional data such as images (the problem known as a curse of dimensionality), while deep learning algorithms can model high-dimensional

data very effectively. With the introduction of this paradigm, neural networks became new SOTA approaches in the VAD task.

Starting with the seminal works [51, 20], diffusion-based generative models have improved the generative modeling of artificial visual systems [14, 43, 26], becoming SOTA models in image generation task. Due to the nature of these methods, they are easily adaptable for image reconstruction tasks, which in its turn can be efficiently used in the field of VAD.

In this study, we propose a model that can be used to capture defects in images from industrial manufacture. This model can be incorporated into the manufacturing process to automate visual inspection by making it independent of manual checks. Formally, we solve the unsupervised VAD problem by relying on diffusion-based models using data that resembles the industrial setup. We split our work in the following way. In Chapter 2 we provide an overview of the VAD problem itself and the common approaches to solving it as well as an overview of the current state of diffusion-based models. In Chapter 3, we describe the proposed approach in detail. In Chapter 4 we describe experiments conducted with our method, provide results and compare them with other existing approaches. The implementation link can be found in Appendix A.

Chapter 2

Related works

2.1 Visual Anomaly Detection

VAD can be categorized mainly into two sections: supervised and unsupervised. In many application scenarios, the collection of abnormal images requires massive human and financial costs. In addition, anomalies can vary significantly in shape, color, and size, and they don't have stable statistical laws. These factors make it difficult for the supervised model to generate appropriate features for abnormal image detection. Therefore, the state of current research focuses mostly on unsupervised VAD, meaning that only normal samples will be included in the training set, while testing will be performed on both normal and abnormal images.

According to the historical development of visual VAD works, the research can be divided into two separate stages: pre-deep learning and after-deep learning. Before the deep learning era, the research was concerned with the following task. After obtaining handcrafted features of the image, using, for instance, SIFT [28], SUFR [2], and HOG [13] the struggle goes to developing detection algorithms relying on statistical and machine learning algorithms, including density estimators and one-class classifiers. The developed models should represent the distribution of normal samples, then, if the test images or their features don't meet the corresponding distribution of the model, they will be classified as anomalies. However, after the success of convolutional neural networks in computer vision applications [18, 50, 66], the attention of researchers on the VAD task shifted from classical machine learning approaches to deep learning methods, as their performance greatly surpasses that of their predecessors.

In addition, we can consider the problem of VAD at two different levels of granularity, which can be represented by two categories: image-level and pixel-level AD. The first one aims to determine whether the whole image is normal or abnormal, whereas the second one aims to localize abnormal regions in the image. With the release of the MVTec AD [4], which is now considered a standard benchmark dataset for unsupervised VAD, most methods try to solve both image-level and pixel-level categories of VAD, as the dataset provides both anomalous masks and labels for testing. In general, methods for solving the unsupervised VAD problem can be divided into two categories: feature-embedding and reconstruction-based.

2.1.1 Image reconstruction

Image reconstruction implies compressing the input image into the latent space using an encoder network and then, reconstructing the original version of the image from the latent space using a decoder network. During the training procedure, only normal samples are fed to the model, which learns to compress and reconstruct them. The anomaly detection mechanism of the reconstruction-based approaches

is based on the following assumption: considering that the training was performed only on normal samples, the model won't be generalized to anomalies during inference, i.e., the reconstruction of abnormal images will not be as good as normal ones. Finally, the reconstruction network will reconstruct the abnormal image in a manner similar to the normal image by eliminating the anomalous regions from it. The difference between the input image and the reconstructed one can be used to generate a prediction.

Autoencoders

Autoencoders are probably the most popular approaches for image reconstruction tasks. The idea of autoencoder was proposed in [19] with the basic idea behind it being redundancy compression and non-redundancy separation. Autoencoder compresses the input data through the hidden narrow layer and then regenerates the original input. As the hidden layer is very narrow, it is expected that the network compresses the redundant information in the input data while retaining and distinguishing the non-redundant information. [22] is the first to introduce the autoencoder into the field of AD with the assumption that redundant information in normal data may not be redundant information in abnormal data and vice versa. Following this work, [44] is a pioneer in utilizing the deep autoencoder for AD of high-dimensional image data. The main direction of research that applies the AE model for VAD goes to resolving the differences between the reconstructed image and the original image. The most simple solution is to take the pixel-wise difference between images. More recent methods account for higher-level differences instead of just comparing individual pixels. For instance, [3] combines the Structure Similarity Index Measure (SSIM) and L2 loss on AE reconstruction and anomaly segmentation, leaving a lot of space for further research. Chung et al. [10] present an Outlier-Exposed Style Distillation Network (OE-SDN) with the idea of style transfer between the original image and the reconstructed one in order to reduce false negative detections.

Many methods suggest increasing difficulties in image reconstruction. They may include various transformations on the original input image, such as geometric transformations, brightness adjustment, noise corruption, or inpainting. After these transformations are applied, the AE is trained to reconstruct the original image from the transformed one. For instance, RIAD [65] randomly masks patches of the training set image and reconstructs them using the AE network. During inference, RIAD randomly creates multiple random masks to generate a reconstructed image, which is then compared to the original image. DRAEM [64] is another popular AE-based technique for VAD. It introduces synthetic abnormal images and reconstructs them as normal, which has a positive effect on the reconstruction network's generalization capacity. In addition, DRAEM uses the additional segmentation network to predict abnormal regions, significantly enhancing the model's ability to segment anomalous regions. Without applying transformations to the input, the common problem was that AE models would generalize very well and reconstruct anomaly regions in abnormal images even if they were not available at the training stage. Increasing difficulties in image reconstruction by applying transformations greatly enhances the capability of the model for anomaly detection as they effectively increase the differences between normal images and abnormal ones after the reconstruction.

Generative Adversarial Networks

Goodfellow [16] proposed a new framework for estimating generative models via an adversarial process, in which there are two simultaneously trained models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. During training, D tries to discriminate between the original and generated by G samples, while the goal of G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. [47] adopts a generative adversarial network for VAD. The GAN model is trained on normal images only. At the test stage, the anomaly can be detected by calculating the difference between the test image and the normal image that is the closest to the test image. The closest normal image is determined using an iterative optimization process. Firstly, the closest latent code for the test image in the GAN's latent space is found with the gradient descent strategy. Then, the generator model from the pre-trained GAN is used to obtain the corresponding normal image. Due to the usage of an iterative search process, the efficiency of this approach is unsatisfactory in practice. Some methods incorporate both adversarial and reconstruction loss for improving performance. For example, [42] leverages autoencoder and adversarial training simultaneously for VAD. In that work, generator G is represented as the autoencoder and it tries to reconstruct a transformed/degraded input image. Similarly to the autoencoders, the corruption of the original input image can increase difficulties in the reconstruction of anomalous regions in abnormal images, which will increase the anomaly scores and therefore improve the detection performance.

Summarizing the current state of image reconstruction for VAD, first of all, it is very intuitive for pixel-level AD, as the anomalies are detected by taking the pixel-wise difference between the original image and its reconstruction. Therefore, it is expected that there will be no difference between a normal image and its reconstruction. However, high-quality image generation is currently a very challenging task and most approaches struggle with reconstructing sharp edges and complex textures. This leads to the problem that normal images don't get reconstructed well enough in some regions, which leads to many false positive samples.

2.1.2 Feature-based approaches

While reconstruction-based approaches detect anomalies as the pixel-wise difference between the input image and its reconstruction, the feature-based approaches detect anomalies in the feature space. Features used for detection can be either handcrafted [59, 7], or learned [23, 49]. Then, a machine learning model can be utilized for modeling the feature distribution of the training data. If the test image features deviate from the modeled distribution, this image will be classified as an anomaly. Numerous categories of approaches are presented that utilize features of the images.

One-class classification

One-class classification on images is a task, which attempts to create a decision boundary of the target class (normal images) using feature representations of the input images. Classic approaches are one-class support vector machines OCSVM [48] and support vector data description (SVDD) [48]. They try to fit a hypersphere to distinguish normal features from abnormal ones during training. Then, at the inference stage, they determine the level of abnormality of the input features based on the relative position of these features to the fitted hypersphere. The advantage

of these methods is that they don't require large amounts of training data once the relevant features are extracted. However, they are quite susceptible to the curse of dimensionality. Usually, deep convolutional neural networks are used for extracting features from the original images. For instance, [21] is a research on a one-class classification method based on transfer learning, which fine-tunes the pre-trained convolution network to extract discriminative image features and then takes the nearest neighbor classification method to construct the one-class classifier.

Density estimation

The idea behind the density estimation is to fit the probability distribution to the normal images or their features. The level of abnormality in the input image during inference is determined by checking input against the established distribution. If the likelihood of the input belonging to this distribution is lower than some threshold, it means that the input deviates from the normal samples, therefore it will be classified as an anomaly. A variety of methods can be used for density estimation, including fitting parametric distributions, such as the Gaussian mixture model [36], and non-parametric estimators, such as KDE or KNN. The predominant method for density estimation used for VAD is Normalizing Flows (NF)-based approaches. Normalizing Flows [37] is a method for constructing complex distributions by transforming a probability density through a series of invertible mappings. By repeatedly applying the rule for change of variables, the initial density 'flows' through the sequence of invertible mappings. At the end of this sequence, we obtain a valid probability distribution and hence this type of flow is referred to as a normalizing flow. In the context of VAD, NF methods extract features from normal images using pre-trained models, such as ResNet-based [17] or Transformer-based [56] models, and transform the feature distribution into a multivariate Gaussian distribution. During the test stage, those images that deviate from the Gaussian distribution after passing through NF will be classified as anomalies. DifferNet [41] is the first research to use NF to address the industrial image AD issue. FastFlow [62] is currently the best-performing NF-based model for VAD. It stacks large and small convolution kernels in the NF module to account for both global and local features of the image, achieving excellent results on the MVTec AD dataset.

Student-Teacher

Similarly to other feature-based approaches, student-teacher heavily depends on pre-trained models such as ResNet and or Transformers. The selection of the ideal teacher model is crucial. Commonly, student-teacher works the following way: the pre-trained backbone model serves as a fixed parameter teacher and is used for feature extraction. During training, the student model learns to mimic the teacher's output. During inference, in case the image is normal, the outputs of the teacher network and student network should be similar, whereas if the input image is abnormal, the outputs of the networks should be distinct. The anomaly map is generated by comparing the feature maps generated by the two networks. This anomaly map could be rescaled to the size of the image in order to obtain anomaly segmentation. [5] is the first to introduce a student-teacher approach into the field of AD. MKD [45] explores that lighter student architecture leads to better performance of the model. RSTPM [61] uses a mechanism for feature transfer from the teacher network to the student network in order to enhance feature reconstruction. Student-Teacher methods are among the predominant approaches in terms of performance for VAD.

Memory bank

Memory-based methods require two components: a powerful pre-trained network for feature extraction and additional memory space. These models are constructed very quickly as they don't require the training procedure. The idea behind them is to utilize this additional memory for storing normal image feature embeddings, previously extracted by the pre-trained network. At the test stage, features of the test image are compared to features stored in this additional memory called memory bank. The abnormality is checked by comparing the distance between test features and those stored in the memory bank. Semantic Pyramid Anomaly Detection (SPADE) [12] utilizes a memory bank for storing a multi-resolution feature pyramid to obtain pixel-level anomaly segmentation results. PatchCore [40] is one of the most significant advancements in industrial VAD that relies on a memory bank, significantly raising the performance of MVTec AD. In PatchCore, the memory bank is subsampled using a coreset-subsampling strategy, which decreases inference time yielding much better results than random subsampling. The level of abnormality of the test image is determined by calculating the distance between the test sample's nearest neighbor features in the memory banks. As for now, memory-bank approaches show the best performance for unsupervised VAD tasks.

In general, feature-embedding methods are more studied and used for the task of unsupervised VAD, as they have superior performance over the reconstruction-based approaches and similar inference time. As of April 2023, the first reconstruction-based method used for VAD in the MVTec AD dataset is 13th in terms of the AUROC detection metric and is released in 2021. This shows that attention to the reconstruction-based approaches for unsupervised VAD tasks gradually declines.

2.2 Diffusion models

Denoising diffusion probabilistic models are a new class of generative latent variable models inspired by nonequilibrium thermodynamics. Generally speaking, denoising diffusion probabilistic models (which can be called "diffusion models" for brevity) are parametrized Markov chains trained using variational inference to produce samples matching the data after a finite time. Transitions of this chain are learned to reverse a diffusion process, which is a Markov chain that gradually adds noise to the data in the opposite direction of sampling until the signal is destroyed.

In the pre-diffusion era, the most effective approaches to image generation problems were GANs, as the quality of generated images was superior to that of alternative approaches such as Autoencoders and Normalizing Flows. However, GANs are notorious for their unstable training and little diversity in the generated samples. In addition, GANs don't model the likelihood function of the generated samples explicitly. They allow only to sample new data, without calculating its likelihood. In comparison, Autoencoders and Normalizing Flows model optimize explicitly the likelihood function of the data, but the quality of the generated samples is not comparable to that of GANs. Diffusion models are likelihood-based approaches and were first mentioned in [51, 53], but a huge amount of attention was attracted to them with the study of denoising diffusion probabilistic models [20]. In this work, the authors managed to achieve SOTA results in terms of the FID metric on the CIFAR10 [27] dataset in the image generation task, surpassing the performance of GANs and also achieving high-quality generations on other datasets with a performance comparable to GANs.

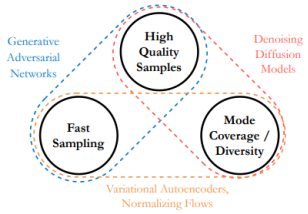


FIGURE 2.1: Trade-offs between different classes of generative models

2.2.1 Background

The essential components of DDPMs are the forward diffusion process (the stochastic process where the signal gets corrupted) and reverse diffusion process (the modeled process, which tries to "heal" the corrupted input)

Forward diffusion process

The forward diffusion process consists of T steps during which the Gaussian noise is gradually added to the input data. It is assumed that after completing all T steps of the process, the signal is completely destroyed (becomes pure Gaussian noise). Let the original input distribution be the image $\mathbf{x_0} \sim q(\mathbf{x})$. The noisy samples obtained during the forward diffusion process will be denoted $\mathbf{x}_1, \ldots, \mathbf{x}_T$. The magnitude of each step of the noising process is indicated by the variance schedule parameter $\{\beta_i \in (0,1)\}_{i=1}^{T}$. The noisy samples at step t come from the following distribution:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = \mathcal{N}\left(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbb{I}\right)$$

And the whole forward diffusion process can be specified by a joint distribution of noisy input:

$$q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) = \prod_{t=1}^{t=T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$$

The authors of [20] use a nice property to sample a noisy signal at arbitrary time step t given the original input x_0. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_{i:}$:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \quad ; \text{ where } \epsilon_{t-1}, \epsilon_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbb{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \quad ; \text{ where } \bar{\epsilon}_{t-2} \text{ is a standard Gaussian as well.} \\
&= \ldots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon;
\end{aligned}$$

Therefore, noisy inputs at any given time step can be obtained by sampling from the following distribution, conditioning only on the original signal:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbb{I}\right)$$

Backward diffusion process

By sampling from $q\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$ (which is, as noted by the authors of the paper, approximately Gaussian as well) it is possible to recover the original signal. However, this denoising process is intractable without conditioning on a less noisy version of the signal. Therefore, the idea is to approximate this conditional distribution by a modeled one p_θ. The joint distribution:

$$p_\theta\left(\mathbf{x}_{0:T}\right) := p\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right), \quad p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right)$$

is called a reverse diffusion process.

Training objective

The learning of p_θ is performed by optimizing the variational lower bound of the negative log-likelihood:

$$\begin{aligned}
\mathbb{E}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] &\leq \mathbb{E}_q\left[-\log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_T\right) - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}\right] \\
&= L
\end{aligned}$$

Further on, the loss function L can be decomposed into several separate components:

$$\begin{aligned}
L &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where } L_T &= D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_T\right)\right) \\
L_t &= D_{\mathrm{KL}}\left(q\left(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t+1}\right)\right) \text{ for } 1 \leq t \leq T-1 \\
L_0 &= -\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)
\end{aligned}$$

Each term, except for L_T, which is constant, and L_0, which is optimized separately, is a KL-divergence of two Gaussians and it can be computed in a closed form.

Noteworthy, for original distribution of the input data q adding conditioning on x_0 makes $q\left(x_{t-1}|x_t\right)$ tractable, having the following distribution:

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = \mathcal{N}\left(\tilde{\boldsymbol{\mu}}\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right) \text{ , where } \qquad \tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t\right)$$

Given that $\mathbf{x}_t$ is available at the training stage, the authors move parametrization from $\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)$ to noise term ϵ_t instead, so that we have $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t)\right)$

Finally, the closed form of the loss term L_t is the following:

$$
\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\left\|\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t,t\right)\right\|_2^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t,\mathbf{x}_0\right)-\boldsymbol{\mu}_\theta\left(\mathbf{x}_t,t\right)\right\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\left\|\boldsymbol{\Sigma}_\theta\right\|_2^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)-\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t,t\right)\right)\right\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\left(1-\alpha_t\right)^2}{2\alpha_t\left(1-\bar{\alpha}_t\right)\left\|\boldsymbol{\Sigma}_\theta\right\|_2^2}\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t,t\right)\right\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\left(1-\alpha_t\right)^2}{2\alpha_t\left(1-\bar{\alpha}_t\right)\left\|\boldsymbol{\Sigma}_\theta\right\|_2^2}\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t,t\right)\right\|^2\right]
\end{aligned}
$$

The authors also found, that removing the scaling parameter improves the training of the model. The simplified version of the loss term is:

$$
L_t = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t,t\right)\right\|^2\right]
$$

2.2.2 Improvements in DDPMs

Reduction in sampling costs

While being able to generate high-quality images, diffusion models are notorious for their slow inference. Starting from Gaussian noise, the model should pass the data sequentially through the whole reverse diffusion process to generate noise-free samples. The number of steps in the diffusion Markov chain was set to 1000 in the original research [20] and many subsequent works [1, 9, 57]. This parameter is rarely changed irrespectively of the dataset and it is generally considered a sufficient number for learning the data distribution. This means that one needs to pass data 1000 times through the learned model in order to generate new samples. Therefore, the primary line of research in the field of diffusion models deals with speeding up the sampling process. In [30] the authors propose to learn variances $\sum_\theta(x_t,t)$ during the backward diffusion process. They find that this modification allows sampling in fewer steps with very little change in sample quality. Specifically, 50 passes were enough to achieve high-quality samples compared to hundreds in the original work. Additionally, their modification improves the log-likelihoods of the generated samples. The authors of [52] replace the backward diffusion process to be non-Markovian proposing DDIM. They show that deterministic reverse diffusion can be used for sampling in fewer steps with enhanced image quality but reduced image diversity. [38] suggests compressing input data by powerful pre-trained autoencoders. Then, diffusion will be learned on the latent representations of the input. After both forward and backward diffusion is completed, the image is passed to the decoder, which will decompress it to its original size.

The increase in likelihood

Given that the diffusion models are optimized for the variational lower bound of negative log-likelihood, another line of improvements to DDPMs investigates the possibilities for increasing the log-likelihood of the model making them more competitive with other likelihood-based models. [25] adds Fourier features to the input

data before passing it to diffusion models. The authors perform a thorough ablation study to confirm that this modification leads to the increased likelihood of the model. Similarly, [54] proposes a new weighting scheme for loss terms of the training objective, which results in an improved likelihood of the model.

Conditional DDPMs

Diffusion models are capable of modeling conditional distributions of the form $p(z|y)$ by modeling $\epsilon_\theta(z_t, t, y)$ Therefore, DDPMs can be easily extended to the tasks of inpainting, deblurring, prompt generation, or other text-to-image and image-to-image tasks. Incorporating different conditioning information, such as text, masks or class labels can be implemented in diffusion models by a few different approaches. [38] propose including the conditional information with direct changes to the denoising backbone model, specifically, by adding the cross-attention mechanism [56], which is known to work for various input modalities.

A different approach [14] proposes a new method for incorporating class information in DDPM. The authors train a classifier on noisy samples. They show that adding a scaled gradient (with respect to data) of this classifier to the noise prediction in the reverse diffusion guides the network to produce samples that correspond to the label information. Their approach is called the ablated diffusion model with classifier guidance and it resulted in outperforming the SOTA approach.

Unsupervised VAD with DDPMs

Concerning unsupervised VAD, at the time of the writing, diffusion models were only applied in the field of medical analysis, specifically, unsupervised MRI segmentation. [58, 33] are the only representatives found for tackling this specific problem.

[58] proposes a new noising scheme, replacing Gaussian noise corruption with simplex noise [32]. Their assumptions are the following: in natural images lower frequency components contribute more to the image. Due to Gaussian white noise having a uniform spectral density, low-frequency components of partially diffused images do not become corrupted to the same extent as high-frequency terms. This limits the discriminatory power of an AnoDDPM model as low-frequency components are inferred to be relatively corruption free, resulting in large anomalous regions being reconstructed in the reverse process. Simplex noise tends to corrupt high-frequency components of the image very well, and that is where the authors expect the abnormalities to lie. Thus, these regions are going to get reconstructed more in the "normal" style, leading to high pixel-wise differences between normal and reconstructed images in these regions.

[33] train diffusion model in the latent space of the pre-trained autoencoder in order to reduce sampling costs during reconstruction. In addition, the research investigates the difference in the meaning of the noise at different time steps. By using the pixel-wise difference between normal input and the reconstruction on the normal validation set, the authors use it for thresholding to guide reconstruction during the inference stage. The proposed solution performed competitively compared with the SOTA methods on both synthetic and real data, thus making the diffusion model very relevant in further research in the medical analysis field.

Chapter 3

Methodology

Given that the unconditional diffusion model is trained on normal samples, one can utilize it in the following way. During the inference stage, a test image would be passed through the t steps of the forward diffusion process giving us the noisy version x_t of the original image. Then, x_t would be fed into the reverse diffusion process, which would gradually denoise the corrupted image until all noise is removed and the reconstructed version of the image $\hat{x}$ is obtained. If the image is normal, we expect to see a very accurate reconstruction of the input. In the case of an abnormal image, the model shouldn't be able to reconstruct the anomalous regions due to their out-of-distribution nature, while leaving non-anomalous regions unchanged. This would lead to the high pixel-wise difference $|\hat{x} - x|$ between the input image and its reconstruction in the poorly reconstructed areas, which are by assumption anomalous. This pixel-wise difference can be used then as an anomaly map, providing the localization of the anomalous regions in the image. In such a setup, the choice of t is crucial. In general, the value t can be logically interpreted as a trade-off between preserving local and semantic information of the image. That is because as we increase the level of noise, the image starts to be less and less distinguishable from its original version, thus losing its local information, while at the same time, it is being pushed to the pure Gaussian noise distribution, which is the respective latent of the original input containing its semantic information. A small t would lead to a tiny amount of noise being added to the image, enough for the model to reconstruct even the anomaly during the reverse diffusion process. On the contrary, a large t would cause a lot of local information to be hidden, which can result in weak reconstruction even in normal images.

(A) Original (B) Reconstruction (20 steps) (C) Reconstruction (300 steps)

FIGURE 3.1: The examples of reconstruction of image (A) when the number of noise steps t is too low (20), which results in the anomaly being reconstructed well (B), and when t is too large, which results in losing too much local information (C)

The described approach requires one essential component - the unconditional

denoising diffusion model trained on the normal samples. In addition, according to this method's assumption, the noise amount t must be carefully chosen. In this case, t can be considered as a hyperparameter of the anomaly detection process.

In this study, we try to build on this standard reconstruction procedure to enhance the capability of diffusion models specifically for VAD. The reasons for such enhancements lie mainly in the impracticality and moderate performance of the approach described. First of all, the selection of t is very dependent on the type of visual data used in the training of the diffusion models. It means, for example, that for the multiclass dataset, optimal for reconstruction t can be very different in each class, therefore, it should be validated separately for each one of them. Secondly, the diffusion models tend to generalize very well. In the context, of VAD, it means that even anomalous regions usually get a great reconstruction under a low-to-moderate amount of noise. Increasing the amount of noise further is usually not appropriate for solving this problem, as we then start losing too much local information, leading to the worse reconstruction of normal pixels.

Considering this, we try to eliminate the disadvantages of this basic DDPM denoising approach by proposing a new method for VAD based on the diffusion models:

- Reconstruction with DDPM using a self-aware sampling technique

By introducing this approach we aim to improve the performance of the basic denoising approach as well as to reduce the dependence on hyperparameter t.

3.1 Reconstruction with DDPM using a self-aware sampling technique

We suggest treating AD under the light of image inpainting, which is the task of complementing the image with the content in arbitrarily specified locations. [29] utilizes unconditional pre-trained DDPMs for the task of inpainting. The setup is quite the same as for the basic reconstruction algorithm described at the beginning of the chapter. The DDPM is trained on the given dataset without the incorporation of any knowledge of inpainting masks. At the inference stage, an inpainted image is passed through the forward diffusion process, and then, the corrupted input is reconstructed in the reverse diffusion process replacing inpainted regions with relevant content. The authors also introduce a resampling technique - a change in the reverse diffusion process so that the unmasked regions remain unaltered and well-harmonized with the rest of the image. In our method, we also include modification in the reverse diffusion process using a technique similar to resampling.

3.1.1 Reconstruction procedure

Suppose we have a trained unconditional DDPM. We denote x as the image coming from the training distribution. The inpainting is performed by the binary image mask m so that $m \odot x$ is the known part of the image and $(1 - m) \odot x$ is the inpainted part.

Given that in the reverse diffusion process, the backbone model predicts the noise ϵ of the noisy image $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$, we can derive x_0 from the noisy version of the image and the predicted noise. Then, according to the basic denoising algorithm, one should sample x_{t-1} from $q(x_{t-1}|x_t, x0)$ and continue the same steps until a denoised version of the image is achieved. Instead of this procedure,

we suggest replacing x_0 with the sum $m \odot x + (1 - m) \odot x_0$. By doing this, we will leverage the known regions of the image in order to infer the masked ones. Then, we follow the strategy described in [29]. Instead of sampling directly $q(x_{t-1}|x_t, m \odot x + (1 - m) \odot x_0)$, we sample $q(x_t|m \odot x + (1 - m) \odot x_0)$ and repeat this procedure n times. In the basic approach, $n = 1$. The authors argue that though the model is leveraging on the context of the known region, the obtained unmasked image is not harmonizing well with the rest of the image. However, increasing the value of n will make the model adjust x_0 during each of these resampling steps, leading to a more harmonized output image.

Algorithm 1 Algorithm which performs inpainting of the image with mask m

Data: Trained model $\epsilon_\theta(x_t, t)$, input image x, inpainting binary mask m, number of noise steps T, number of resampling steps N
Result: $\hat{x}$ - reconstructed version of the input image x
$\epsilon \sim \mathcal{N}(0, \mathbb{I})$
$x_{masked} \leftarrow m \odot x$
$x_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_{masked} + \sqrt{1 - \bar{\alpha}_t}\epsilon$
for $t = T$ **to** 0 **do**
    `// Denoising steps`
    for $n = 1$ **to** N **do**
        `// Resampling steps`
        $\hat{x} \leftarrow \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}$
        $\hat{x} \leftarrow m \odot x + (1 - m) \odot \hat{x}$
        $x_t \leftarrow \sqrt{\bar{\alpha}_t}\hat{x} + \sqrt{1 - \bar{\alpha}_t}\epsilon$
return $\hat{x}$

We note that reconstructing masked images with the described approach reduces the dependence on hyperparameter t. Firstly, by incorporating the unmasked part of the image to infer the masked part, we remove the upper threshold of t, e.g., if the model reconstructs inpainted regions with k noise time steps well enough, the near-same output will be achieved with any other number of time steps larger than k. This effect is due to conditioning on the unmasked part of the image, which preserves a lot of local and semantic image details and pushes the resulting output to be well-harmonized with the rest of the image. Additionally, it is not necessary to select t for different image classes for performance improvement. Previously, multiple ts are needed to cover a variety of anomalies. In the inpainting-based approach, all potential anomalies are covered with the same mask, thus removing the need to select t for any specific abnormality.

The described approach for image inpainting expects a mask m as an input. Ideally, this mask should cover the anomalous region in the image as only this part gets reconstructed while the unmasked regions remain unchanged. However, there is an inherent problem in the nature of anomalies. They can be irregular and diverse in size, form, and location. Therefore, it is impossible to choose one general mask, which would cover all possible locations of anomalies. That is why the mask should be selected separately for every image.

3.1.2 Inpainting mask creation

In this study, we will utilize the model's predictions as a way to form masks m for the task of inpainting. As described previously, the objective of the model is to minimize the difference between the predicted noise $\epsilon_\theta(x_t, t)$ and noise ϵ, which was initially used to corrupt the image. We assume that if the image is abnormal, then, the predicted and ground truth noise is going to be different in the anomalous regions as the model hasn't seen the anomaly during training. Such reasoning can be used directly to form the anomaly maps as a difference $|\epsilon_\theta(x_t, t) - \epsilon|$, similar to the basic approach described at the beginning of the section. However, we notice that there are a lot of separate areas unrelated to anomaly regions that are mispredicted by a large margin, therefore we don't use this difference as the end tool for anomaly detection. Instead, we look at $|\epsilon_\theta(x_t, t) - \epsilon|$ for multiple t, as different t correspond to different features within the image [33]. We collect such tensors for multiple normal images from a validation set (a portion of normal images that was not used during the training) to form a tensor $q \in \mathbb{R}^{b \times a \times h \times w}$, where a refers to the number of noise steps, b - the number of images in the validation set, h and w to the height and the width of the images respectively. Next, we average tensor q across the first dimension. We denote the averaged tensor as Q. Then, we must select the specific percentile p for Q to finally obtain the threshold map $Q_p \in R^{h \times w}$. For the test image, we compute a tensor $Q_{test} \in R^{h \times w}$, by calculating $|\epsilon_\theta(x_t, t) - \epsilon|$ for multiple t and averaging them. The inpainting mask m is obtained by comparing Q_{test} to Q_p. $m_{i,j} = 1$ if $Q_{test_{i,j}} > Q_{p_{i,j}}$ and $m_{i,j} = 0$ otherwise.

We formalize the process of obtaining inpainting mask m with the following procedure:

Algorithm 2 Algorithm for forming the inpainting mask m

Data: Trained model $\epsilon_\theta(x_t, t)$, validation set samples S, test image x, noise levels range $[t_{start}, t_{end}]$, threshold percentile p
Result: m - inpainting mask
$\epsilon \sim \mathcal{N}(0, \mathbb{I})$
$m \leftarrow empty \quad tensor \quad with \quad shape \quad of \quad x$
Function `calculateDifference`(x, t_{from}, t_{to})
  $Q \leftarrow []$
  for $t = t_{from}$ **to** t_{to} **do**
    $x_t \leftarrow \sqrt{\bar{\alpha}_t} x + \sqrt{1 - \bar{\alpha}_t}\epsilon$
    $\delta \leftarrow |\epsilon_\theta(x_t, t) - \epsilon|$
    $Q.add(\delta)$
  return $mean(Q)$
Function `calculateThreshold`(S, t_{from}, t_{to}, p)
  $q \leftarrow []$
  foreach $s \in S$ **do**
    q.add(`calculateDifference`(s, t_{from}, t_{to}))
  $Q_p \leftarrow percentile(q, p)$
  return Q_p
$Q_p \leftarrow$ `calculateThreshold`$(S, t_{start}, t_{end}, p)$
$Q_{test} \leftarrow$ `calculateDifference`(x, t_{start}, t_{end})
$m_{i,j} \leftarrow 1 \quad if \quad Q_{test_{i,j}} > Q_{p_{i,j}} \quad else \quad 0$
return m

3.1.3 Anomaly score

Multi-Scale Reconstruction Error Anomaly Map

By firstly obtaining the inpainting mask m and then performing reconstruction we will get the image $\hat{x}$ from the original image x. In order to obtain the final anomaly map we calculate the $|\hat{x} - x|$ at different scales to consider both pixel-wise and patch-wise reconstruction errors, as proposed in [6]. For different scales, $L = \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$, we compute the error for downsampled $\hat{x}$ and x and, then, upsample the error back to the original size. We denote such errors as $Err(x, \hat{x})_l$. The final anomaly map is obtained by averaging each scale's error map and applying a mean filter for better stability similar [65]. $Err(x, \hat{x}) = \frac{1}{N_L} \sum_{l \in L} Err(x, \hat{x})_l * f_{x \times s}$, where $f_{x \times s}$ is the mean filter of size $s \times s$ and $*$ stands for convolution operator. The anomaly score is then calculated as $max(Err(x, \hat{x})$.

Learned Anomaly Map

The anomaly map can be obtained in a discriminative way as well. For that, we will utilize an additional segmentation subnetwork. This idea was first proposed in DRAEM [64]. The segmentation subnetwork learns the anomaly map by training the joint representation of the original image and its reconstruction outputting the decision boundary between normal and anomalous samples. This method enables anomaly detection without the need for additional post-processing steps after the anomaly-free reconstruction is obtained. The training is performed by using artificially simulated anomalies. The concatenated image and its reconstruction are used as input to the segmentation model. We expect the model will learn the decision boundary between normal images and the reconstructed ones more accurately than in hand-crafted post-processing while generalizing from artificial anomalies to real ones.

We create artificial anomalies by the same procedure as in DRAEM. Worth noticing, the simulated anomalies don't require to resemble the ones from the target domain. They are needed only to generate appearances that deviate from the distribution of normal images. According to the DRAEM, this will allow learning the appropriate distance function to recognize the anomaly by its deviation from normality.

The algorithm for creating anomalies is the following one. Perlin noise [32] P is generated to capture a range of anomaly shapes. The noise is binarized by a threshold to create the anomaly mask I_a. The anomaly texture source image A is sampled from the dataset unrelated to the input image distribution. The 3 random augmentation functions from the set {posterize, sharpness, solarize, equalize, brightness change, color change, auto-contrast} are chosen and applied to texture image A. The augmented version of image A is masked with the anomaly mask and blended with the original image I to create anomalies. Formally, the image I_a used for the training of the segmentation subnetwork is obtained with the following formula: $I_a = \bar{M}_a \odot I + (1 - \beta)(M_a \odot I) + \beta (M_a \odot A)$, where $\bar{M}_a$ is the inverse of M_a and β is the opacity parameter in blending. This parameter is sampled uniformly from an interval, i.e., $\beta \in [0.1, 1.0]$.

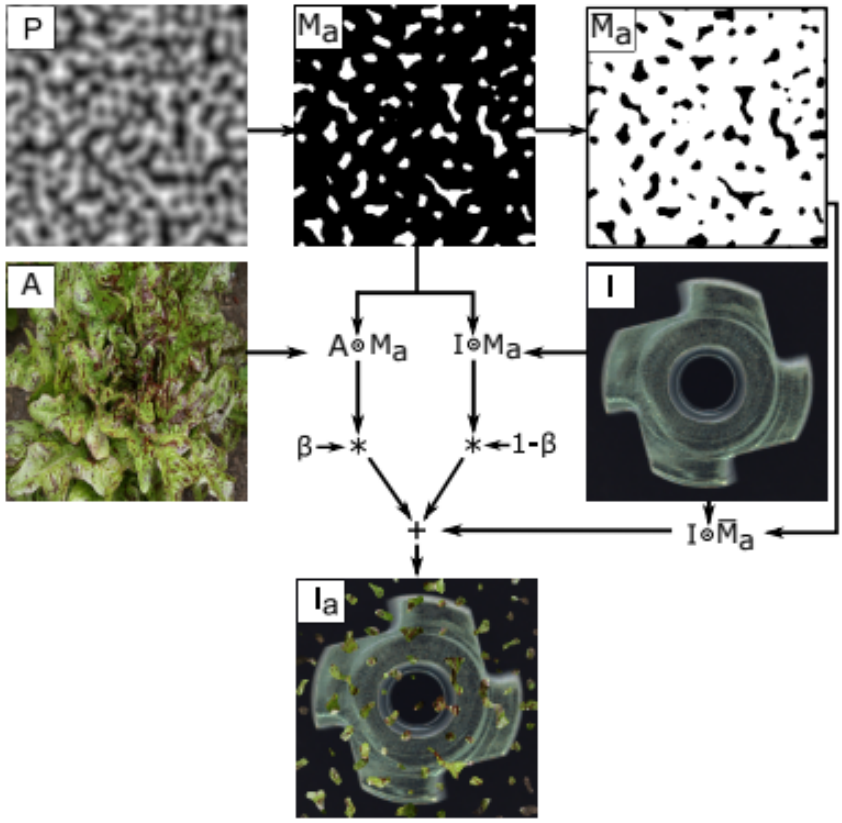


FIGURE 3.2: The strategy for generating anomalies proposed in DRAEM [64]

We use a basic UNet [39] architecture for the segmentation subnetwork. The expected input is a six-channel tensor of concatenated image and its reconstruction with the output being the anomaly map. The maximum value over this anomaly map will be used as the anomaly score for the input image.

3.1.4 Backbone model

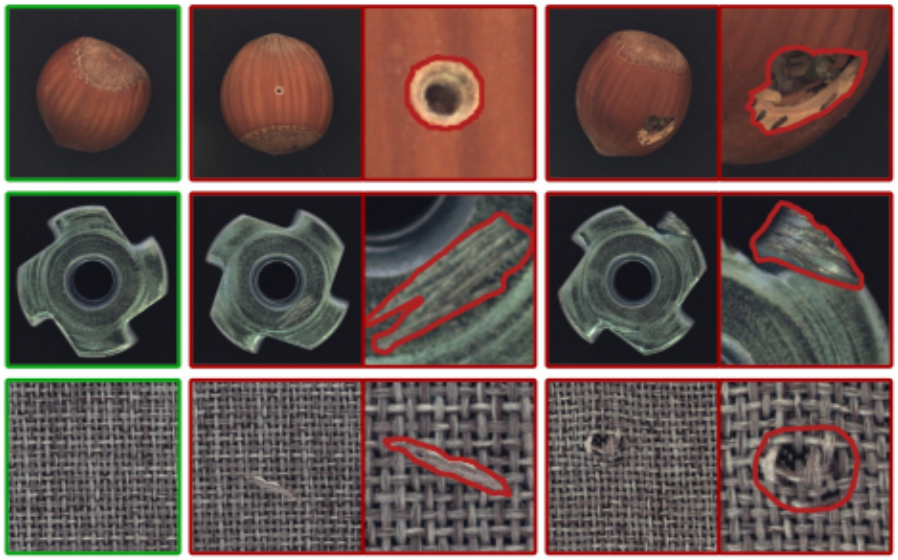
The backbone model that we use in the reverse diffusion process $\epsilon_\theta(x_t, t)$ to predict the noise ϵ is the same as in the original work [20], which is a UNet-like architecture similar to PixelCNN++ [46] based on Wide ResNet [63]. Parameters are shared across time steps t, which are encoded by the Transformer sinusoidal position embedding [56] into each block in the network. In total, there are 4 downsampling blocks with 4 mirrored upsampling blocks. Each resolution map consists of two convolutional residual blocks and a self-attention [56] block followed by downsampling/upsampling. Grouped normalization [34] is used between the convolutional layers. Residual connections are used between the downsampling and upsampling blocks. Complementing the original work, where the input to the model is expected to be the noisy image x_t, we modify it to be the concatenated x_t and the model's prediction of $\hat{x} = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t,t)}{\sqrt{\bar{\alpha}_t}}$ as proposed in [8]. This modification leads to the increased quality of predictions according to the authors. The resulting model consists of 271M of trainable parameters with a total of 273 megabytes.

Chapter 4

Experiments and Results

4.1 Dataset

For a long period of time, the common setup for evaluating the performance of anomaly detectors was to adapt the existing classification datasets with the class labels being available. The basic approach was to select a set of classes and re-label them as abnormalities, while the rest of the classes were considered normalities. The anomaly detection algorithm would be trained on the normal samples without being exposed to the anomalous samples according to the unsupervised VAD scenario. During the test stage, one should check if the trained model can discriminate images to be either inliers or outliers. This approach provides tons of training and testing data, however, the created anomalies differ significantly from the training distribution. Therefore, it is very unclear how the proposed methods would generalize to real-world cases, especially the industrial setting.



(A) Examples of normal/abnormal images

	Category	# Train	# Test (good)	# Test (defective)	# Defect groups	# Defect regions	Image side length
Textures	Carpet	280	28	89	5	97	1024
	Grid	264	21	57	5	170	1024
	Leather	245	32	92	5	99	1024
	Tile	230	33	84	5	86	840
	Wood	247	19	60	5	168	1024
Objects	Bottle	209	20	63	3	68	900
	Cable	224	58	92	8	151	1024
	Capsule	219	23	109	5	114	1000
	Hazelnut	391	40	70	4	136	1024
	Metal Nut	220	22	93	4	132	700
	Pill	267	26	141	7	245	800
	Screw	320	41	119	5	135	1024
	Toothbrush	60	12	30	1	66	1024
	Transistor	213	60	40	4	44	1024
	Zipper	240	32	119	7	177	1024
	Total	3629	467	1258	73	1888	-

(B) Statistical information about dataset

FIGURE 4.1: Image examples (A) and class distributions (B) in the MVTec AD dataset

In order to remove the ambiguities of classification datasets for unsupervised VAD, the MVTec AD [4] was introduced. This dataset mimics real-world industrial inspection scenarios and consists of 5354 high-resolution images of five unique textures and ten unique objects from different domains. There are 73 different types of anomalies in the form of defects in the objects or textures. For each defect image, there is a pixel-accurate ground truth regions that allow evaluating methods for both image and pixel-level anomaly detection. At the time being, this dataset is considered to be a standard benchmark for evaluating the performance of anomaly detection methods. Therefore, we will utilize this dataset to evaluate the performance of our approach as well, additionally extracting 20% of normal samples to form a validation set that won't be used for training.

4.2 Implementation details

We use Python 3 [55] as a programming language in this study. For the implementation of neural networks, we use the PyTorch [31] library. The training pipeline is created using PyTorch Lightning [15]. For image processing, we use the Pillow [11] library. We also use the Hydra [60] library for config manipulations. Wandb service is used for logging. The whole stack of libraries can be viewed in the Github repository.

4.3 Training details

Both training and inference of DDPM were conducted on NVIDIA GeForce RTX 3090 Ti GPU with 24GB of RAM. We train the DDPM model with the backbone described in 3.1.4. For the backbone, we use a batch size of 16 elements, and the learning rate is set to 0,00002. We use 1000 time steps in DDPM, the loss function is the same as described in 2.2.1, sigmoid function for the beta schedule is used. Adam [24] is used as an optimizer during training. We train the model for 500 epochs, choosing then the best one in terms of the value of the loss function on the validation set.

For training the segmentation subnetwork, we use a setup similar to the training of DDPM. The only difference is the increased batch size (32 elements), and changed learning rate (0,001). We use the Cross-Entropy and Focal loss in our experiments with the segmentation model. 50 epochs were enough for training for validation loss to converge.

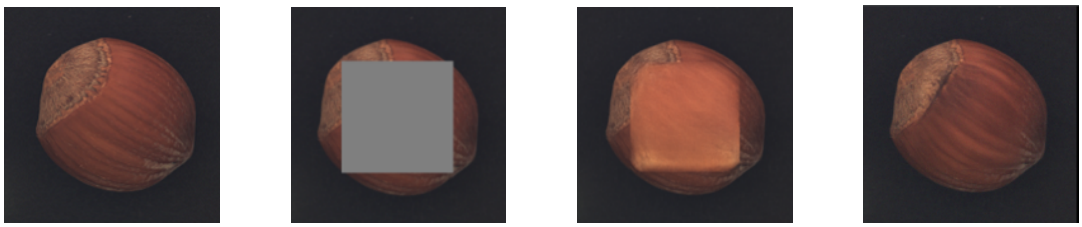
All images are resized to 256×256 size in all experiments.

4.4 Evaluation

We measure the performance of our method in terms of image-level detection AUCROC metric, which is commonly used for the evaluation of anomaly detectors on the MVTec AD dataset.

We don't assess the results of solving image inpainting problem separately, as our primary goal is to build a method that discriminates well between normalities and abnormalities.

4.5 Experiments with the proposed method

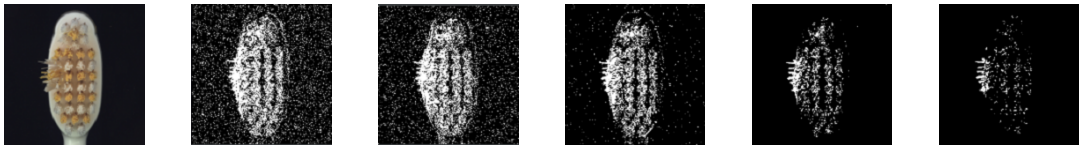


(A) Original image (B) Masked image (C) Restored (n=1) (D) Restored (n=5)

FIGURE 4.2: Our experiments with resampling: both reconstructed images are obtained by setting the number of noise steps t to 80. The number of resampling steps is 1 in (C) and 5 in (D)

Our experiments start with training the unconditional DDPM on the MVTec AD data with the configuration specified in 4.3. After that, we explore the effect of resampling steps for image inpainting under arbitrary masks. We note that resampling is crucial to generate well-harmonized outputs. Fig. 4.3 shows the reconstruction results of the image with a square mask covering 25% of the image area. In our anomaly detection pipeline, we set the number of resampling steps to 5, as further increases seem to have no effect on the reconstruction. The number of noise steps t is set to 50, given that it copes with inpainting well enough, when the resampling is set to 5.

The masks for anomaly detection are generated by the algorithm 2. Our observation is that the backbone model's noise misprediction is independent of the range $[t_{start}, t_{end}]$, i.e., an arbitrary range can be chosen to generate masks. We set t_{start} to 300 and t_{end} to 350. The percentile p is used to control the size of the mask. There is no direct mapping between p and the amount of space the mask would cover, as it is extremely specific to the image class. The lower p is, the larger space covered by the mask will be. Ideally, we would like to generate larger masks in order not to miss any anomalies. However, too large masks can pose problems for reconstruction, i.e., the unmasked part of the image would be too small to infer the rest of the image. For us, setting p to 0.8 works best in terms of the final detection metric.



(A) Image (B) $p = 0.5$ (C) $p = 0.7$ (D) $p = 0.8$ (E) $p = 0.9$ (F) $p = 0.98$

FIGURE 4.3: The anomaly masks used for inpainting depending on hyperparameter p for abnormal image (A)

Without retraining the DDPM, we compare the proposed approach with the basic one, in which we add t steps of noise to the original image and then denoise it in the reverse diffusion process. In this basic approach, we set t to 50, the same as in our method.

In terms of time costs, the basic approach is 5 times as fast as the proposed one for obtaining the reconstruction of the image due to the resampling factor. However, in this study, we don't perform specific time measurements for a time cost comparison.

	AnoGAN	GANomaly	Skip GANomaly	DAGAN	Basic DDPM denoising	Ours (using multi-scale anomaly map)	Ours (using segmentation subnetwork)
bottle	0.800	0.794	0.937	0.983	0.949	0.938	**0.996**
capsule	0.422	0.721	0.718	0.687	0.797	**0.852**	0.839
grid	0.871	0.743	0.657	0.867	0.990	**1.000**	**1.000**
leather	0.451	0.808	0.908	0.844	0.886	0.970	**0.982**
pill	0.711	0.671	0.758	0.768	0.604	0.744	**0.903**
tile	0.401	0.720	0.850	**0.961**	0.725	0.811	**0.960**
zipper	0.715	0.744	0.663	0.781	0.848	0.868	**0.999**
cable	0.477	0.711	0.674	0.665	0.616	0.607	**0.782**
carpet	0.337	**0.821**	0.795	0.903	0.577	0.659	0.662
hazelnut	0.259	0.874	0.906	1.000	0.968	**0.982**	0.981
metal nut	0.284	0.694	0.790	0.815	0.772	0.717	**0.891**
screw	0.100	**1.000**	**1.000**	**1.000**	0.964	0.970	0.842
toothbrush	0.439	0.700	0.689	0.950	0.889	**0.990**	0.938
wood	0.567	0.920	0.919	0.979	**0.965**	0.960	0.912
transistor	0.692	0.808	0.814	0.794	0.822	0.779	**0.962**
average	0.502	0.782	0.805	0.86.6	0.822	0.856	**0.910**

TABLE 4.1: Anomaly Detection Performance (AUCROC) on MVTec: comparison with GAN-based methods

	PatchCore	DifferNet	Basic DDPM denoising	Ours (using multi-scale anomaly map)	Ours (using segmentation subnetwork)
bottle	**1.000**	0.990	0.949	0.938	0.996
capsule	**0.980**	0.869	0.797	0.852	0.839
grid	0.986	0.840	0.990	**1.000**	**1.000**
leather	**1.000**	0.971	0.886	0.970	0.982
pill	**0.970**	0.888	0.604	0.744	0.903
tile	**0.994**	**0.994**	0.725	0.811	0.960
zipper	0.992	0.951	0.848	0.868	**0.999**
cable	0.993	**0.959**	0.616	0.607	0.782
carpet	**0.980**	0.929	0.577	0.659	0.662
hazelnut	**1.000**	0.993	0.968	0.982	0.981
metal nut	**0.997**	0.961	0.772	0.717	0.891
screw	**0.964**	0.963	0.936	0.970	0.842
toothbrush	**1.000**	0.986	0.889	0.990	0.938
wood	0.992	**0.998**	0.965	0.960	0.912
transistor	**0.999**	0.911	0.822	0.779	0.962
average	**0.990**	0.949	0.822	0.856	0.910

TABLE 4.2: Anomaly Detection Performance (AUCROC) on MVTec AD: comparison with recent SOTA methods

The results provided are compared with GAN-based methods 4.1 and feature-embedding methods 4.2.

We include our metrics when using both a multi-scale reconstruction error map and learned with an additional segmentation subnetwork anomaly map. Both approaches beat the basic DDPM denoising in terms of the detection AUCROC for most of the classes. The first one outperforms 3 out of 4 highlighted GAN-based approaches, while the learned anomaly map outperforms all 4 of them in terms of averaged detection metric. However, our method is not highly competitive with feature-embedding-based approaches, outperforming them only in 2 classes in terms of AUCROC.

We notice that poorly detected classes are roughly the same for the DDPM-based methods. Those are cable and carpet. The first one is the most difficult class in terms of structure while the carpet is a simple texture. We notice that the model struggles to reconstruct even normalities for the cable class, while, on the other hand, the model generalizes to anomalies and reconstructs the images very well for the carpet class.

In general, our experiments show that the proposed approach can be quite efficient for unsupervised VAD, outperforming other reconstruction-based approaches like GANs and even showing comparable results with feature-embedding-based SOTA solutions for some of the classes.

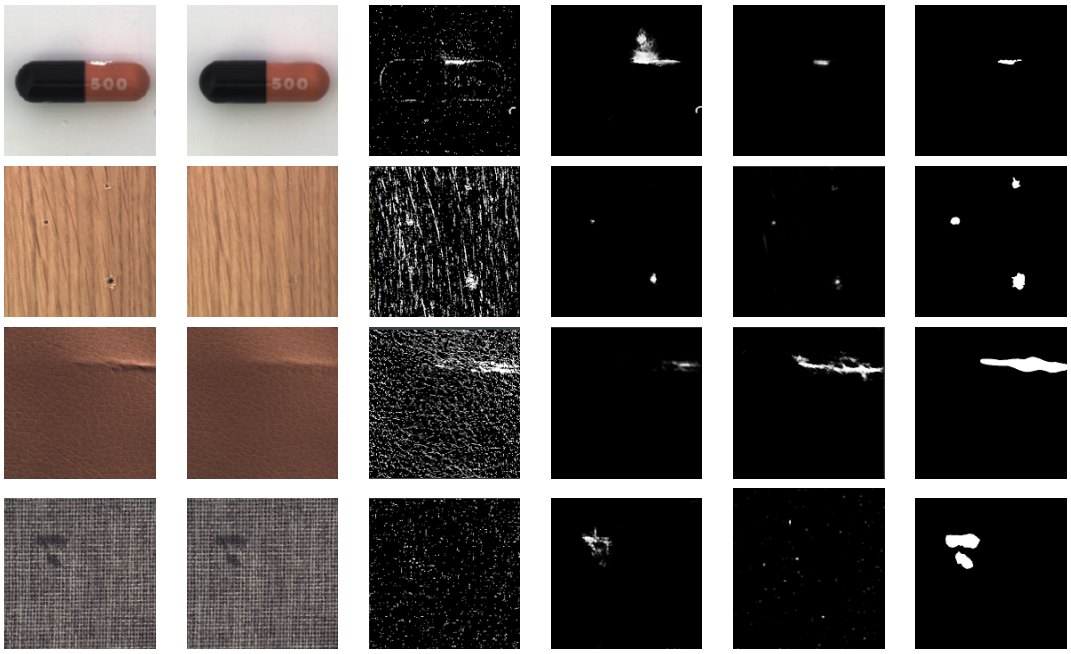


FIGURE 4.4: The results of our method: (A) - original image, (B) - reconstructed image, (C) - inpainting mask m, (D) - anomaly map obtained with multi-scale reconstruction error map, (E) - anomaly map obtained with segmentation network trained on artificial anomalies and their reconstruction, (F) - Ground-truth anomaly map

Chapter 5

Conclusion and Future work

5.1 Conclusion

In this study, we leveraged the power of denoising diffusion models for the task of unsupervised VAD. We proposed a reconstruction-based approach that treats AD as an image inpainting problem. We call the process of reconstruction a self-aware sampling as the mask for inpainting is produced by the same model that performs reconstruction. We also experimented with the segmentation subnetwork that learns the decision boundary between normalities and abnormalities using artificially simulated anomalies for training images. We showed that the suggested approach is superior to some of the GAN-based solutions such as GANomaly. However, the proposed model is still not competitive with feature-embedding-based methods on average, due to the poor performance of the DDPM in some of the classes.

5.2 Future work

Although the DDPMs are getting a lot of attention nowadays, they still remain not fully discovered in many fields, specifically in unsupervised VAD. This work explores the reconstruction capabilities of the DDPM, though it leaves a lot of space for improvement.

- One potential direction for future work is speeding up the diffusion process as the main bottleneck in time costs for the pipeline.
- It is possible to perform the diffusion process in the latent space of pre-trained autoencoders. Not only it can result in reduced time costs, but it can also affect the quality of reconstruction. After the diffusion is performed, the decoder network should bring the reconstruction back to pixel space. Given that it is trained on normal images as well as the DDPM, it is going to generalize poorly to anomalies that were potentially omitted during the diffusion.
- Training conditional diffusion model can be promising in the field of unsupervised VAD. For instance, it is possible to condition on CLIP [35] embeddings of the normal images or utilize metric learning to minimize the feature difference between the original image and its reconstruction. These enhancements can all lead to the poorer reconstruction of abnormal regions therefore to better anomaly detection.
- Many experiments can be done within the training process of DDPM itself. The number of total noise steps, the noise schedule algorithm, and the depth of the backbone network are all good candidates for potential improvement in DDPM, which is a crucial element in anomaly detection.

Appendix A

GitHub repository with code:

- Diffusion-based anomaly detection

Bibliography

[1] Fan Bao et al. *Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models*. 2022. arXiv: 2201.06503 [cs.LG].

[2] Herbert Bay et al. "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding* 110.3 (2008). Similarity Matching in Computer Vision and Multimedia, pp. 346 –359. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014. URL: http://www.sciencedirect.com/science/article/pii/S1077314207001555.

[3] Paul Bergmann et al. "Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders". In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2019. DOI: 10.5220/0007364503720380. URL: https://doi.org/10.5220%2F0007364503720380.

[4] Paul Bergmann et al. "MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[5] Paul Bergmann et al. "Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. DOI: 10.1109/cvpr42600.2020.00424. URL: https://doi.org/10.1109%2Fcvpr42600.2020.00424.

[6] Yuanpu Cao, Lu Lin, and Jinghui Chen. *Robustness for Free: Adversarially Robust Anomaly Detection Through Diffusion Model*. 2023. URL: https://openreview.net/forum?id=imIlOpuEsi.

[7] Diego Carrera et al. "Defect Detection in SEM Images of Nanofibrous Materials". In: *IEEE Transactions on Industrial Informatics* 13 (2017), pp. 551–561.

[8] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. *Analog Bits: Generating Discrete Data using Diffusion Models with Self-Conditioning*. 2023. arXiv: 2208.04202 [cs.CV].

[9] Jooyoung Choi et al. *Perception Prioritized Training of Diffusion Models*. 2022. arXiv: 2204.00227 [cs.CV].

[10] Hwehee Chung et al. "Unsupervised Anomaly Detection Using Style Distillation". In: *IEEE Access* 8 (2020), pp. 221494–221502. DOI: 10.1109/ACCESS.2020.3043473.

[11] Alex Clark. *Pillow (PIL Fork) Documentation*. 2015. URL: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

[12] Niv Cohen and Yedid Hoshen. *Sub-Image Anomaly Detection with Deep Pyramid Correspondences*. 2021. arXiv: 2005.02357 [cs.CV].

[13] N. Dalal and B. Triggs. "Histograms of Oriented Gradients for Human Detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* 1 (2005), pp. 886–893. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360.

[14] Prafulla Dhariwal and Alexander Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

[15] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.4. Mar. 2019. DOI: 10.5281/zenodo.3828935. URL: https://github.com/Lightning-AI/lightning.

[16] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[17] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[18] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[19] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313 (2006), pp. 504 –507.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG].

[21] Chenlong Hu et al. "One-class Text Classification with Multi-modal Deep Support Vector Data Description". In: 28.4 (2021), pp. 1053–1088. DOI: 10.5715/jnlp.28.1053.

[22] Nathalie Japkowicz, Catherine Myers, and Mark A. Gluck. "A Novelty Detection Approach to Classification". In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*. Morgan Kaufmann, 1995, pp. 518–523. URL: http://ijcai.org/Proceedings/95-1/Papers/068.pdf.

[23] Junfeng Jing et al. "Mobile-Unet: An efficient convolutional neural network for fabric defect detection". In: *Textile Research Journal* 92.1-2 (2022), pp. 30–42. DOI: 10.1177/0040517520928604. eprint: https://doi.org/10.1177/0040517520928604. URL: https://doi.org/10.1177/0040517520928604.

[24] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[25] Diederik P Kingma et al. "On Density Estimation with Diffusion Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer et al. 2021. URL: https://openreview.net/forum?id=2LdBqxc1Yv.

[26] Diederik P. Kingma et al. *Variational Diffusion Models*. 2022. arXiv: 2107.00630 [cs.LG].

[27] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-10 (Canadian Institute for Advanced Research)". In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[28] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[29] Andreas Lugmayr et al. *RePaint: Inpainting using Denoising Diffusion Probabilistic Models*. 2022. arXiv: 2201.09865 [cs.CV].

[30] Alex Nichol and Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. 2021. arXiv: 2102.09672 [cs.LG].

[31] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[32] Ken Perlin. "Improving Noise". In: *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '02. San Antonio, Texas: Association for Computing Machinery, 2002, 681–682. ISBN: 1581135211. DOI: 10.1145/566570.566636. URL: https://doi.org/10.1145/566570.566636.

[33] Walter H. L. Pinaya et al. *Fast Unsupervised Brain Anomaly Detection and Segmentation with Diffusion Models*. 2022. arXiv: 2206.03461 [cs.CV].

[34] Siyuan Qiao et al. *Micro-Batch Training with Batch-Channel Normalization and Weight Standardization*. 2020. arXiv: 1903.10520 [cs.CV].

[35] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[36] Douglas A. Reynolds. "Gaussian Mixture Models". In: *Encyclopedia of Biometrics*. 2009.

[37] Danilo Jimenez Rezende and Shakir Mohamed. *Variational Inference with Normalizing Flows*. 2016. arXiv: 1505.05770 [stat.ML].

[38] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: 2112.10752 [cs.CV].

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: vol. 9351. Oct. 2015, pp. 234–241. ISBN: 978-3-319-24573-7. DOI: 10.1007/978-3-319-24574-4_28.

[40] Karsten Roth et al. *Towards Total Recall in Industrial Anomaly Detection*. 2022. arXiv: 2106.08265 [cs.CV].

[41] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. "Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows". In: *Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021. URL: /brokenurl#arxiv,GitHub,YouTube.

[42] Mohammad Sabokrou et al. *Adversarially Learned One-Class Classifier for Novelty Detection*. 2018. arXiv: 1802.09088 [cs.CV].

[43] Chitwan Saharia et al. *Image Super-Resolution via Iterative Refinement*. 2021. arXiv: 2104.07636 [eess.IV].

[44] Mayu Sakurada and Takehisa Yairi. "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction". In: *MLSDA'14*. 2014.

[45] Mohammadreza Salehi et al. *Multiresolution Knowledge Distillation for Anomaly Detection*. 2020. arXiv: 2011.11108 [cs.CV].

[46] Tim Salimans et al. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*. 2017. arXiv: 1701.05517 [cs.LG].

[47] Thomas Schlegl et al. *Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery*. 2017. arXiv: 1703.05921 [cs.CV].

[48] Bernhard Schölkopf et al. "Estimating Support of a High-Dimensional Distribution". In: *Neural Computation* 13 (July 2001), pp. 1443–1471. DOI: 10.1162/089976601750264965.

[49] Yong Shi, Jie Yang, and Zhiquan Qi. "Unsupervised anomaly segmentation via deep feature reconstruction". In: *Neurocomputing* 424 (2021), pp. 9–22. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2020.11.018. URL: https://www.sciencedirect.com/science/article/pii/S0925231220317951.

[50] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[51] Jascha Sohl-Dickstein et al. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. arXiv: 1503.03585 [cs.LG].

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. *Denoising Diffusion Implicit Models*. 2022. arXiv: 2010.02502 [cs.LG].

[53] Yang Song and Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. 2020. arXiv: 1907.05600 [cs.LG].

[54] Yang Song et al. *Maximum Likelihood Training of Score-Based Diffusion Models*. 2021. arXiv: 2101.09258 [stat.ML].

[55] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[56] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[57] Daniel Watson et al. *Learning to Efficiently Sample from Diffusion Probabilistic Models*. 2021. arXiv: 2106.03802 [cs.LG].

[58] Julian Wyatt et al. "AnoDDPM: Anomaly Detection With Denoising Diffusion Probabilistic Models Using Simplex Noise". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2022, pp. 650–656.

[59] Xianghua Xie and Majid Mirmehdi. "TEXEMS: Texture Exemplars for Defect Detection on Random Textured Surfaces". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29.8 (2007), 1454–1464. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1038. URL: https://doi.org/10.1109/TPAMI.2007.1038.

[60] Omry Yadan. *Hydra - A framework for elegantly configuring complex applications*. Github. 2019. URL: https://github.com/facebookresearch/hydra.

[61] Shinji Yamada, Satoshi Kamiya, and Kazuhiro Hotta. *Reconstructed Student-Teacher and Discriminative Networks for Anomaly Detection*. 2022. arXiv: 2210.07548 [cs.CV].

[62] Jiawei Yu et al. *FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows*. 2021. arXiv: 2111.07677 [cs.CV].

[63] Sergey Zagoruyko and Nikos Komodakis. *Wide Residual Networks*. 2017. arXiv: 1605.07146 [cs.CV].

[64] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. *DRAEM – A discriminatively trained reconstruction embedding for surface anomaly detection*. 2021. arXiv: 2108.07610 [cs.CV].

[65] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. "Reconstruction by inpainting for visual anomaly detection". In: *Pattern Recognition* 112 (2021), p. 107706. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107706. URL: https://www.sciencedirect.com/science/article/pii/S0031320320305094.

[66] Yulun Zhang et al. *Residual Dense Network for Image Restoration*. 2020. arXiv: 1812.10477 [cs.CV].