Ensemble Methods for Visual Anomaly Detection in Manufacturing Settings

Toller Thesis Titel
Master thesis by Marc Saghir
Date of submission: May 3, 2024

1. Review: Super Supervisor
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit erkläre ich, Marc Saghir, dass ich die vorliegende Arbeit gemäß § 22 Abs. 7 APB der TU Darmstadt selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 3. Mai 2024

M. Saghir

Abstract

Abstract

Contents

Figures and Tables

List of Figures

List of Tables

Abbreviations, Symbols and Operators

List of Abbreviations

Notation	Description
DDPG	Deep Deterministic Policy Gradient
DQN	Deep Q Network
IAD	Image Anomaly Detection
ML	Machine Learning
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
SAC	Soft Actor Critic
TRPO	Trust Region Policy Optimization

List of Symbols

Notation	Description
A	continuous action space
S	continuous state space
$\mathcal{H}(\cdot)$	entropy
$\pi(a\|s_t)$	Policy

1. Introduction

Image Anomaly detection as a form of quality control is a widely popular practice in modern manuacturing processes. This also holds true for industrial settings. Ever since the industrial revolution, the need for manufactured metal parts has skyrocketed to alltime highs, due to new innovation and a growing usage of metal in modern applications. With rising innovation and production also comes a high need for quality assurance alongside raised standards and requirements. These strict conditions serve among other things to avoid product failure in situations that could cause fatal consequences. Here the quality control in form of anomaly detection often starts at the individual parts manufactured for a single purpose, which demands a large effort and lots of resources due to factors like the named increasing production rate. In earlier days this meant procedures like manual stochastic quality checks of produced parts, a practice that in its nature cannot give complete certainty and requires lots of human labour and thus time and money. Later with the rise of computers and especially sophisticated computer vision methods this process of quality control was more and more being automated using methods like IAD, to create a more efficient and easier quality control process. This alongside the striving for even higher reliability and recent developments in artificial intelligence brought forth IAD as the popular research field that it is today. IAD in our context is a subcategory of general anomaly detection and aims at distinguishing images of a category that conform to some chosen norm from anomalous images of the same category that dont . An example would be creating a classifier that is given the image of a screw and can detect whether or not it conforms to our expectations, which in a manufacturing setting likely means to meet the companies quality standards.
With IAD being a very recent and popular field, there are many different deep learning apporaches that have established themselves over the last couple of years. The best performing ones have generally been unsuperised learning apporaches. This stems from the fact, that in any manufacturing setting, there usually exist far less anomalous parts than regular ones, which creates a significant data imbalance. Moreover it can pose as difficult to actually obtain a large number of data points and great variance, since it is a lot of work to coordinate with adequate manufacturing facilities and also implement the necessary infrastructure to take pictures. This problem is supported by the fact that there are little well established datasets being used for modern IAD research. There are still some credible and widely used datasets, amongst them the MVTecAD [1] dataset acting as some sort of gold standard. The dataset will be discussed in greater detail in the background section. Regarding the kind of anomaly detection models, there is again a great variety of approaches that follow a somewhat different strategy of differentiating between the classes. Still most of them can be categorized with two classes: representation or reconstruction based methods. While representation approaches aim at creating a feature based representation in different forms to then compare the features of new input images, reconstruction based ones try to learn how to recreate the part shown in the image as an anomaly free object, and then comparing the constructed product to the original input. Both workflows are visualized in figure xyz, which showcases the different steps of the respetive methods as described. It is to be said that both approaches offer high quality predictions, yet feature representation methods have more frequently shown in latest research to achieve state of the art results.
The current state of IAD generally consists of very high performing classifiers. Here it is important to differentiate between different applications of those classifiers. There is anomaly detection in form of image

classification, which was already mentioned. Furthermore there is anomaly locaization. This describes the process of image segmentation to point out the specific regions in which the deteted anomaly occurs. Lastly besides the applications, one can also categorize kinds of anomalies. The most researched anomaly types are so called structural anomalies, which can be described somwhat as superficial damages of the parts material or shape, i.e. a strongly bent screw or one that is broken in the middle. Yet recently there has been a new dataset from the creates of MVTecAD that covers logical anomalies, namey the MVTecAD LOCO [7] dataset. Logical anomalies denote ones that violate an abstract set of rules. More concretely this can mean instances like a metal part with an irregular number of holes, or a label missing. Whereas state of the art approaches regurlarly produce performance metrics of up to 99.6% on classification of structural anomalies, they strongly differ in anomaly localization performance. Moreso does the performance plummet when approaching to classify and localize logaical anomalies. Additionally models often show inconsitencies between different subtypes of structural and logical anomalies, especially during localization. Another important consideration is, that many different anomaly detection approaches have different distinct weaknesses, which worsens most of their performances in certain situations. These inconsitencies and perforance gabs demonstrate that IAD as such is not yet solved and still has a need for improved robustness and generalizability. This need also holds true due to logical anomalies making up an important new domain of automated quality control, as more complex parts could be tested for requirements. Moreover the showcasing of performance inconsistencies between structual and logical anomalies indicate logical anomalies of being a differnt problem domain. Achieving better translation between those settings could serve as a basis for tackling other problems in this field that may arise in future settings.

1.1. Contributions

This research provides multiple contributions to the field of image anomaly detection, in an effort to further push the progress of robust anomaly localization in differnt domains.

1. To address the problems mentioned at the end of the last section, we attempt to build a heterogeneous feature level ensemble network, combining differnt state of the art IAD approaches, with the goal to improve general performance but also robustness in image localization and logical anomaly detection. This ensemble network is then tested on the MVTecAD LOCO dataset to observe its performance regarding both anomaly types.
2. Furtheremore an extensive study on the performance of a diverse selection of IDA methods on the MVTecAD LOCO dataset is performed. This serves to highlight the current state of anomaly detection with respect to logical problems, and also investigate the application potential of those approaches a form of transfer learning setting.
3. Second to last we introduce a new category to the MVTecAD LOCO dataset to further increase the diversity of this dataset and strengthen the focus of this thesis on metal maufactured parts. Many datasets either use synthetic data or images in a very linical setting, therefore this attempt for variance is also a step towards IAD on more realistic datasets.
4. Finally the mentioned network and experiments are also streamlined into an easy to use pipeline to be used for future experiments in that area.

The contributions mentioned firstly benefit faster research entry and an accelerated experimentation process, with an intuitive setup, as well as potential industrial applications. Here it is to be mentioned that since the

ensemble already is of heterogeneous nature, it is particularly uncomplicated to experiment using various IAD apporaches.
Furthermore they give more insight into the capabilities of existing methods in an industrial setting and thus also provide a more various and practical setting than the prior categories in the MVTecAD [1] dataset. The same methods are also testet on their limitations regarding logical anomalies which was earlier made out to be a relevant aspect of anomaly detection in current manufacturing quality control settings. Lastly through the use of a robust ensemble approach for heterogeneous classifers, this opens up possibilities for expanding the field of application of SOTA IAD methods to other domains with robust performance and may also produce more usable results in real world IAD settings. The presented network can also be used as a foundation for future experiments in different directions. For example, the pipeline may be efficiently used to start investigations on multiperspective datasets in anomaly detection, a topic that also could further advance current IAD applciations.

In this work we will firstly discuss relevant background knowledge to get a grasp on the latest important IAD research, aswell as fundamental principles that are relevant to the ensemble approach presented here. Section 2.1 and section 2.2 give a broad overview of state of the art IAD approaches aswell as intuitive knowledge on how to view them. Afterwards sections 2.3 and 2.4 offer in insight into the testing and evaluation environments of this context. Lastly section 2.5 and section 4.2.4 are focussed on background knowledge to common approaches regarding the ensemble model talked about before.
In chapter 4 we present the concrete implementation of this works contributions, starting with section 4.1 dealing with the survey on logical anomalies. Section 7.3 presents the novel dataset category introduced in this paper and finally section 4.2 concerns the realisation of this ensemble approach.
Chapters 6 and 7 afterwards deal with an in depth analysis of our findings as well as an outlook on how to interprete these results and what future research in this topic may look like.

2. Background

2.1. Categorization of Anomaly detection

Over the last years a great amount of different approaches to solving IAD have been published. In order to compare different anomaly detection methods, categorizing them with respect to different aspects serves a better comparative analysis in regards of strengths and weaknesses. It also helps gain a better understanding of the different approaches in the global IAD context and is useful detect possible patterns in performance correlated to categories. This obviously also holds true for IAD approaches. Survey papers like Xie et al. [8] and Liu et al. [2] primarily compare but also categorize IAD approaches, leading to the conclusion that there are effective and generally applicable ways to group certain approaches, as visualized in Figure xyz. Here it has to be said that the following categorizations naturally are a generalization, as there may be different niche settings or even approaches who combine multiple categories.
The first distinction relevant to our work is between supervised and unsupervised settings. Current deep learning approaches that have established themselves as state of the art in image anomaly detection are almost exclusively unsupervised approaches. This partially stems from the fact that in practical situations, anomalous images occur far less than normal images, hence the word "normal". This is especially true in industrial settings, due to the high performance of production sites nowadays. Therefore if one were to consider using a classical supervised learning approach to detect anomalies, either a strong class imbalance or a nonrepresentative class distribution would constitute a problem. While there are some solutions for this, they often either do not suffice for imbalances of this magnitude or far to resource extensive. To overcome these issues, some supervised approaches [9] operate in a few-shot setting which limit the training data amount needed for proper training. Nevertheless as the focus on unsupervised IAD methods in current research persists, this work will also restrict itself to such approaches. This also facilitates the execution of the ensemble approach presented in chapter 4.
Looking into the unsupervised anomaly setting, the next important distinction is between reconstruction based approaches and representation based ones. They differ in the sense, that the former are comparing the distances between two images and the latter measure distances between feature representations.
Reconstruction approaches first learn to reconstruct the objects given in the input images. This is done by feeding the network normal train data, aswell as noisy data. Noisy data are are which are altered using noise, although the exact noise application is depending on the specific approach. During testing, after having successfully learnt to reconstruct anomaly free images, the method is then given an input image, reconstructs it and compares both images using some sort of distance measure. This process is also depicted in figure xyz. Representation approaches on the other hand use feature embedding methods to obtain feature representations of images and compare those. As shown in figure xyz, during training the model is learning to correctly extract features of input images. When given an anomalous sample during testing, the model then also extracts the features from the input data and compares those to its prior feature level representations of the class object. A decision is then made aswell using a distance measure. Both classes of IAD methods have shown to produce state of the art results. Yet currently more approaches are representation

based [2] as they have shown SOTA performance more consistently. Nevertheless it is reasonable to focus on both kinds of IAD, as they may excel at different regions of anomaly detection and evaluation criteria. Namely reconstruction methods are often showing a notable performance at pixel level anomaly detection in comparison to feature embedding/representation methods, as their principle is based on pixelwise comparisons of input and reconstructed data.
Again, looking at the representation based approaches, some distinctions can be made on exactly how the method implements a representation based procedure. The main approaches in this category are ones featuring a memory bank, teacher-student architecture, distribution map and ones employ a one-class classification strategy. The characteristics of each strategy are visualized in figure xyz. The subcategory of memory bank denotes the procedure to store feature representations, that are extracted from training images, into a data collection structure. This structure is then used to compare new features from input images to the stored ones to form a decision. Memory bank approaches offer the upside of little training time and quick construction, yet the usually suffer from high memory usage and costly inference, due to the feature representations being stored into memory. Some papers have addressed this problem. Famously patchcore [10] introduced a coreset-subsampled memory bank, greatly improving on said bottlenecks and setting precedent for more efficient memory bank approaches. Teacher-student architectures reference the use of two networks for anomaly detection. This architecture has also been one of the more effective ones and its performance greatly depends on factors like the selection of the teacher model and way of knowledge transfer between the two networks. Here the teacher model is usually a pretrained backbone, that transfers knowledge onto the student model during training time, whereas the student model is simultaneously learning representations from the teacher model aswell as learning how to represent the input data by itself. During testing, the extracted features of teacher and student are compared, which would then be similar for normal images but have larger differences when presented an anomalous data point. Next, distribution map approaches try to map the features from their original distribution into a more suitable one. This vastly facilitates a identification of anomalous features as shown in figure xyz. Such an approach requires a method to map the features between distributions. Often times a variation of normalizing flow is utilized for this [2]. Normalizing flows as a class of generative models [11] are advantageous for transforming probability distributions because they provide a flexible framework to model complex distributions and efficient sampling. This enables accurate distribution mapping aswell as sampling. Lastly, one-class classification is somewhat similar to a distribution mapping approach. The key difference is that the latter maps features into a desired distribution and the former focusses on finding boundaries between normal and anomalous features. To efficiently do that, features are projected into a suited space using a network. To learn an accurate boundary, the approaches generate fake anomalous features to then differentiate from normal ones. This method greatly relys on the the quality of generated features and as such may be variyngly effective. A typical approach for generating fake samples in this works context is with the use of gaussian noise [3].
Circling back to reconstruction based methods, they also can be parted into subcategories. Here the most predominating one is the use of autoencoders to obtain a generated image. Many reconstruction approaches make use of typical AE encoder and decoder structures. They are mainly seperated by the method of resolving differences between the input image and the reconstructed one. While there are too many difference evaluation approaches to name, DRAEM [5] can be named here as one of the most famous reconstruction autoencoder approaches. It uses the output of the reconstructive network in combination with the original image as the input for a discriminative subnetwork and achieves very good results and demonstrated a nearly equal effectiveness of reconstruction based methods to representation based ones. Autoencoder approaches like DRAEM are, despite being popular, often computationally expensive to train for IAD, as they require a lot of training epochs and memory. Another, albeit less popular, approach in this category would be the use of GAN architectures/concepts to try and solve the detection problem. While we don't cover the basic principles of GANs in this chapter, GANs applied in IAD may still suffer the same disadvantages of regular ones, which include

training instability, high computational demand and a higher difficulty of correct training and evaluation. Hence the use of GANs for IAD is not commonly seen, although the nature of them may promise more realistic and higher quality training data than other approaches. Another kind of reconstruction based IAD involves the use of transformer structures. They allow for good caputiring of spatial relationships and long distance feature extraction [12], making them useful for not only structural but also logical anomaly detection. Papers like [13] make use of transformers for feature reconstruction and as they are very limited in reconstructing anomalous features well, making for an easy distinction. They show in their experiments that their approach ADTR is able to outperform all shown baselines, including a variety of different autoencoder apporoaches. Finally there also exists an approach that has been gaining popularity lately: the usage of the diffuion model [14]. Papers like [15] leverage the models ability to capture complex dependencies to detect anomalies in IAD, and other papers [16] further improved its efficiency by speeding up the denoising process.
After carefully categorizing the important classes of unsupervised IAD, it is now discernible how many different approaches towards anomaly detection exist, and may yield differnt merits. In section 2.2 we further elaborate on a few select IAD approaches from different categories who were utilized for the MVTecAD LOCO performance analysis or the ensemble model.

2.2. Anomaly Detection Methods

In this section we further dive into some of the IAD approaches that can be viewed as class representative to some extent. All following methods have demonstrated SOTA performance in at least some categories and are subjects in the performance study on the MVTecAD LOCO dataset. In the this work we investigated the most widely popular categories of IAD, namely memory bank, one-class classification, teacher-student, distribution maps and autoencoders.

2.2.1. PatchCore

As the memory bank approach patchcore [10] has been chosen. It has demonstrated very high accuracy in both image and pixel level anomaly detection and set a high standard for performance after its release. The fundemental principle of patchcore is visualized in figure 2.1 and is according to the basic memory bank principle. First a pretrained feature extractor is used to retreive features of the input data. Next in this paper, the features are turned into locally aware feature patches, meaning the feature maps are cut into small fields and applied preprocessing operations like pooling to ensure the patches to be locally aware and also of similar dimension throughout the training. This results in patches of image x_i that adhere to equation 2.1 where $\mathcal{N}_P^{(h,w)}$ describes a neighborhood of patches $\mathcal{P}$ in an image with height h, width h and patchsize p, as described in 2.2. Moreover we use $\phi_{i,j} = \phi_j(x_i)$ as the feature maps of hierarchy j of image x_i.

$$\phi_{i,j}(\mathcal{N}_P^{(h,w)}) = \{_{agg}(\{\phi_{i,j}(a,b) \| (a,b) \in \mathcal{N}_P^{(h,w)}\}) \tag{2.1}$$

$$\begin{aligned} \mathcal{N}_P^{(h,w)} = \{(a,b) \| a \in [h - [p/2], ..., h + [p/2]], \\ b \in [w - [p/2], ..., w + [p/2]], \} \end{aligned} \tag{2.2}$$

The effectiveness of this patchwise approach is justified with each patch, despite being small, having a large enough receptive field size to provide a meaningful anomalous context when learned. Before commiting the resulting patches to the memory bank, patchcore first induces coreset subsampling on the set of feature vecotrs. This ensures a reasonably small memory bank size that is large enough to offer good results, and yet is not to time and memory consuming when performing nearest neighbor search. As a memory bank structure, patchcore utilizes a search index by FaissNN(referenz) which offers the application of kNN with great speed, performance and even a desired tradeoff if necessary for a task. Since performance is the main criterion in IAD research though, this tradeoff is not utilized.

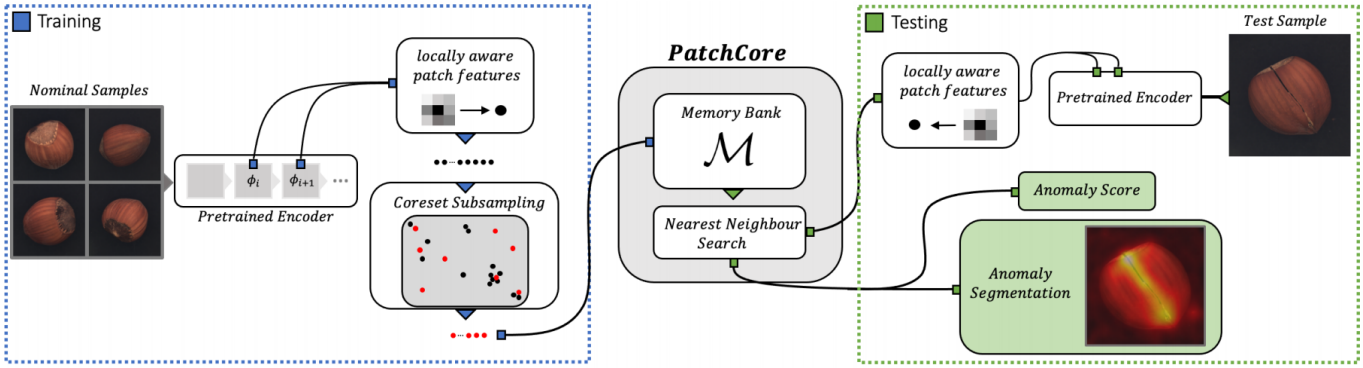


Figure 2.1.: Hier korrekt zitieren

At test time, the input images are processed exactly like the training images were, and afterwards the model searches for the nearest neighbor of each patch, and calculates a distance measument as in equation 2.3:

$$m^{test,*}, m^* = \underset{m^{test} \in \mathcal{P}(x^{test})}{\operatorname{argmax}} \underset{m \in \mathcal{M}}{\operatorname{argmin}} \|m^{test} - m\|_2$$
$$s^* = \|m^{test,*} - m^*\|_2 \tag{2.3}$$

The distance scores are then used twofold: The maximum distance in addition to a weighting relative to other neighbors distances, indicates the anomaly score on an image level. Meanwhile the patchwise distances are reshaped, interpolatesd and smoothed to a segmentation map for pixelwise anomaly detection.
Patchcore has greatly improved the status of prior thought slow and rarther inefficient memory bank approaches throough its subsampling approach. It moreover offers to train a model with little time since the only training that is really done is to fit the memory bank structure with patch vectors. Besides low training time it also requires low storage cost and offers very high and state of the art performance that can also be boosted using simple averaging ensembles in its paper. Its performance on the MVTecAD dataset [1] is documented in table xyz. Yet due to its nature as a representation based approach, patchcores efficiency is limited by the the pretrained feature extractors acting as its backbones. This problem persists but is also approached by the paper, as they offer a implementation to train the backbone extractor alongside the actual training.

2.2.2. SimpleNet

SimpleNet [3] is a recent one-class classification approach that has also shown to compete with other state of the art methods. Table xyz records its performance to be on the same standard as patchcore for most of

the categories, Moreover it was introduced by (authors of simplenet) as a easy to use or application friendly approach, (nebensatz hier).

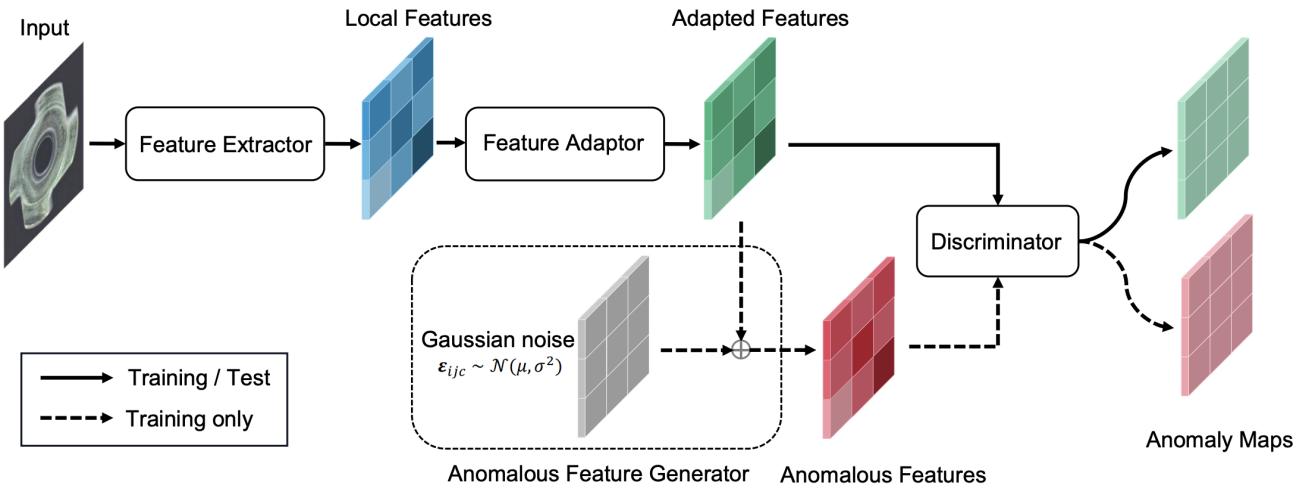


Figure 2.2.: Hier korrekt zitieren

As a one-class classificaion or feature projection approach, SimpleNet utilizes a sub network to project extracet features into another space, to then find a decision criterion between nomral and abnormal features. The exact pipeline of simplenet is illustrated in figure 2.2. While training, a pretrained feature extractor is used to retreive features from the input data. Here it is noteworthy, that SinmpleNet adopts PatchCores approach of transforming extracted features into locally aware patches. This means in figure 2.2, the blue tiled pane represents features in form patch vectors that represent the characteristics of the image. On the basis of SimpleNet being a one-class classification approahc, the features are projected in the next step, visually at the green tiled pane. [3] conducts multiple ablation experiments on different implementations of the feature adapter. They conclude that a single fully connected layer network yield the best performance. The extracted features are then projected using the feature adapter, which is also simultaneously trained during the training session. Before training the discriminator, the resulting features are used in combination with gaussian noise to produce fake features. This is done by adding the generated noise onto the features. Afterwards the discriminator is trained by giving it the normal patch vectors of the input data aswell as the artificial fake features, to then learn to differentiate between them. For the discriminator, the paper reports the use of a two-layer MLP to be very effective. The training objective for the discriminator is composed of a truncated $l1$ loss and batchwise normalization:

$$\mathcal{L} = min \sum_{x^i \in \mathcal{X}_{train}} \sum_{h,w} \frac{l^i_{h,w}}{H_O * W_0} \tag{2.4}$$
$$l^i_{h,w} = max(0, th^+ - D_\psi(q^i_{h,w})) + max(0, -th^- - D_\psi(q^{-i}_{h,w}))$$

Here

$$th^+$$

and

$$th^-$$

are truncation terms preventing potential overfitting, and

$$D_\psi(q^i_{h,w})$$

represents the estimated normality of adapted features by the discriminator.

At test time the features from the input image are undergoing the described process and the projected patch vectors are given to the discriminator, resulting in a score for each patch. Similar to PatchCore, to generate segmentations, the scores are reshaped, bilinearly interpolated and smoothened. Also the largest score acts as the image anomaly score.

SimpleNet offers great results as to be seen in table xyz, in anomaly detection and localization. According to [3] its inference speed is with 79 frames per second (FPS) not only comparably slower that PatchCore [10] with only 10 FPS and PaDiM [17] (1 FPS), all being representation based approaches, but also slower than many reconstruction based approaches like DRAEM with 67 FPS. As other representation approaches its performance is also dependent on the quality of the backbone feature extractor.

2.2.3. RevDist

As a teacher student model, [6] has shown to produce state of the art results on current datasets. The paper is an extension of a previous reverse distillation approach from [18]. The base paper utilizes knowledge distillation through a classical teacher student architecture as explained in section 2.1, though they stand out due to two key differences visualized in figure xyz. Firstly they present a unconventional knowledge distillation flow as shown. The direct distillation through the teacher network still exists, yert the input image does not directly influence the student learning anymore. This results in increased compactness as the low dinmensional output of the teacher model is fed to the student, whereas the student model has to predict the teachers feature representation. This mimics the encoder - decoder structure and therefore may yield advantges that come with this approach. Secondly the paper introduces a further one-class bottleneck embedding module. This is comprised of a multi-scale feature fusion block to aggregate low and high level features, and a one-calss embedding for retaining essential information important to the students decoding process.
Circling back to [6], the paper extends the prior reverse distillation approach in a way that limits possible abnormal patterns to heavily impact student learning. To do so they introduce firtly a multiscale teacher architecture aswell as projection blocks after each respective teacher block. Those multiscale projection blocks are made of stacked convolutional blocks and are designed to regulate possible abnormal information patterns that might flow from teacher blocks to the student. This is depicted in figure xyz, which shows the image to describe the approaches training pipeline. For inference the teachers informations are passed to the respective multilayer projection blocks, as well as their students counterparts informations after the reverse distillation process.

2.2.4. CSFlow

CSFlow [4] is one of the more robust distribution map approaches for IAD. As a distribution map approach its pipeline can be seen in figure 2.3. One of the concepts that make approaches like CSFlow stand out of other distribution mapping methods is the feature extraction and mapping on different image scales. This cross-scale flow is using cross-convolutional blocks [2] to make use of context between different scales. The goal of optimization is to find out of distribution features which are assumed to have a small likelihood.

With normalizing flow [11] as mapping approach on a cross scale level, this makes it possible to infere likelihood information of between scales contexts, leading to a much more improved and robust localization of anomalies. The blocks used to map between scales are convolutional to preserve spatial dimensions. This method automatically results in an interpretable segmentation map at different scales. The values of the segmentation map are then used to produce an image score for classical anomaly detection as usual.

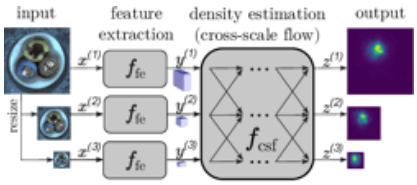


Figure 2.3.: Hier korrekt zitieren

CSFlow has proven state of the art performance on the MVTecAD dataset like the other approaches (table xyz), and makes for precise localization on this dataset. Additionally the cross-scale flow allows for convenient handling of high dimensionality input with little training data [4], which has been a previously reported problem in works like [19].

2.2.5. DRAEM

DRAEM [5] is one of the reconstruction based approaches that were selected for this work. This paper is one of the more representative ones for the reconstruction branch of IAD techniques. The method is based on the reconstruction abilities of autoencoders to recreate normal images from altered ones. While in recent research representation based approaches did show SOTA performance more often, DRAEM demonstrated that representation based methods are able to compete in the same region. Tabel xyz shows DRAEMs results on the MVTecAD dataset as reported in their(/authors name) paper.

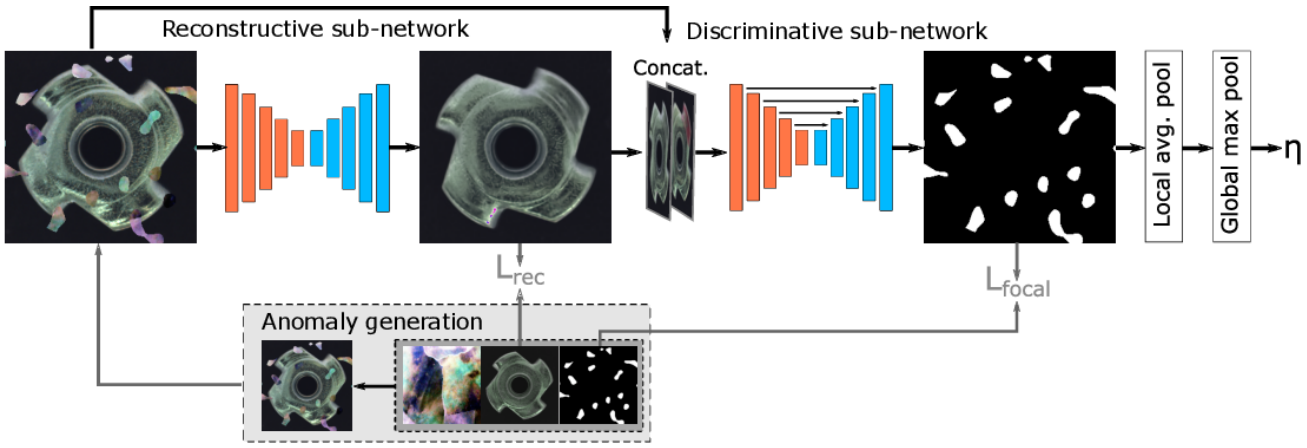


Figure 2.4.: Hier korrekt zitieren

DRAEM utilizes two subnetworks in their anomaly detection pipeline as seen in figure 2.4. The first on is the so called reconstructive sub-network and is an encoder-decoder structure. The network is trained to receive an image with artificial noise and reconstruct it into its original state. As the distance measurement between the

reconstructed image and the original one, the second network is used, namely the discriminative sub-network. This network consists of a u-net (u net referenz) like architecture. Input is a channel-wise concatenation of the altereed input image and the reconstructed one. Since the reconstructed image is supposed to be anomaly free, the two inputs should differ to some extent, which is to be learned by the second sub-network. The network outputs a segmentation map, which can be used as the pixelwise anomaly detection map. Moreover the resulting map is further used for image level detection. To be interpretable, the map is smoothed by a convolution filter and the maximum value is taken as the image anomaly score. Reconstruction methods often utilize similarity functions such as SSIM [20] to obtain segmentation masks [5] [2]. While the first sub-network makes SSIM as part of their loss function, the discriminative sub-network breaks free of this as it automatically learns an appropriate distance measure, making it much more robust.

DRAEM offers a exemplary good performance, robustness and moderate speed with around 67 frames per second at inference time [3]. Still this approach may struggle on the reconstructive part of its pipeline when synthesizing near-in-distribution anomalies on the original input images [2] (rausfinden was das heißt und umschreiben).

Method	Bottle	Cable	Capsule	Carpet	Grid	Hazelnut	Leather	Metal Nut	Pill	Screw	Tile	Toothbrush	Transistor	Wood	Zipper	Average
PC [10]	1.0	0.997	0.981	0.982	0.983	1.0	1.0	1.0	0.971	0.990	0.989	0.989	0.997	0.999	0.997	0.992
SN [3]	1.0	0.999	0.997	0.997	0.997	1.0	1.0	1.0	0.990	0.982	0.998	0.997	1.0	1.0	0.999	0.996
CS [4]	0.998	0.991	0.971	1.0	0.990	0.996	1.0	0.991	0.986	0.976	1.0	0.919	0.993	1.0	0.997	0.987
DRAEM [5]	0.992	0.918	0.985	0.970	0.999	1.0	1.0	0.987	0.989	0.939	0.996	1.0	0.931	0.991	1.0	0.980
RevDist [6]	1.0	0.992	0.990	1.0	1.0	1.0	1.0	1.0	0.984	0.989	0.997	1.0	0.985	0.993	0.986	0.994

Table 2.1.: Collection of image auroc results of reviewed IAD methods on the MVTecAD [1] dataset. The data was collected from [2] [3] [4] [5] [6]

Method	Bottle	Cable	Capsule	Carpet	Grid	Hazelnut	Leather	Metal Nut	Pill	Screw	Tile	Toothbrush	Transistor	Wood	Zipper	Average
PC [10]	0.986	0.987	0.991	0.987	0.988	0.988	0.993	0.990	0.986	0.995	0.963	0.989	0.971	0.952	0.990	0.984
SN [3]	0.980	0.976	0.989	0.982	0.988	0.979	0.992	0.988	0.986	0.993	0.970	0.985	0.976	0.945	0.989	0.981
CS [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DRAEM [5]	0.991	0.947	0.947	0.955	0.997	0.997	0.986	0.995	0.976	0.976	0.992	0.981	0.909	0.964	0.988	0.973
RevDist [6]	0.988	0.984	0.988	0.992	0.993	0.992	0.994	0.981	0.983	0.997	0.966	0.991	0.943	0.958	0.98	0.983

Table 2.2.: Collection of pixel auroc results of reviewed IAD methods on the MVTecAD [1] dataset. The data was collected from [2] [3] [5] [6]. The numbers for [4] were not recorded

2.3. Metrics

Metrics are known to be an important part of developing any artificial intelligence related models. Many of them are used to infer different characteristics of model performance and should be used in different appropriate circumstances, depending on which aspect is important for the current application. Therefore, before the actual developing, one must first choose appropriate metrics to optimize and evaluate on later. IAD as a research area themselves has certain metrics that are the main performance evaluation tool across most papers. A collection of different metrics in this domain are displayed in table 2.3, which is taken from

[2]. Visible are well known ones from many other machine learning models like precision, recall, TPR, FPR and the F1-Score. These are generally applicable in most cases, but are not listed in any recent important papers and thus are not important for any analyses in this work. The other metrics are more IAD specific. By a large margin, the most important scoring standard is the AUROC. This metric is usually referenced for image level binary classification and gives an indication on how good the model is able to distinguish between both classes. Its calculation can be seein in table 2.3. Moreover it can be used on a pixel level, which is also a popular approach but not utilized everywhere. Next in importance is the per-region overlap(PRO) score or also the area under the PRO score(AU-PRO). This metric denotes the per-region overlap of two areas on a pixel level and can be calculated using the averaged sum of the ratio between true positives and ground truth pixels per predicted area. The two areas compared are generally an image mask and the according segmentation by the model. The AU-PRO is then calculated by plotting the PRO score at different threholds for the segmentations, and reporting the area under the curve. This can be used to rate the segmentation performance of different models and is also a frequently featured metric in IAD related research. Related to this score is the saturated per-region overlap (sPRO) and also the according are under the curve, the AU-sPRO. This metric was introduced in [7] and is briefly mentioned in section 2.4. The method of deriving the sPRO score is shown again in equation abc, where m denotes the amount of anomalous regions in an image, $A_i|i \in \{1, ..., m\}$ an anomalous region among them and $s_i|i \in \{1, ..., m\}$ a respective saturation threshold. P is considered to be the pixels classified as anomalous in the target image. The sPRO score is the calculated by averaging the intersections of all predictions and ground truths of an image, while norming the values by the saturation threshold and proividing an upper limit of 1 per region. It is to be said that this gives a similar view on the segmentation performance as the PRO score, as it is a generalized form of it and can producet the same results if the saturation threshold would be equal to the amount of anomalous pixels per region. However, due to its cap of 1, it also rates differently large segmentations equally in cases where the anomalous position posesses some uncertainty. Figure xy demonstrates this behaviour in the case of a logical anomaly of the pushpin class. As visible, the logical anomaly consists of an empty pushpin compartment. The missing pushpin could be placed in any place of this smaller box for it to be valid, therefore an amount of pixels equal to the amount a visual pushpin posesses would suffice. Yet the conventional PRO score would keep on rising as the segmented area gets larger within the anomalous region. Due to the saturation score and limit. this is prevented by the sPRO metric as figure xy shows it to be already saturated once the minimum required amount of pixels is achieved. The saturation scores for each anomaly have to be individually set for each anomaly, and are given for the five classes of MVTecAD LOCO [7].

Metric/Level	**Formula**	**Remarks/Usage**
Precision (P) ↑	$P = TP/(TP + FP)$	True Positive (TP), False Positive (FP)
Recall (R) ↑	$R = TP/(TP + FN)$	False Negative (FN), True Positive Rate (TPR)
True Positive Rate (TPR) ↑	$TPR = TP/(TP + TN)$	True Negative (TN)
False Positive Rate (FPR) ↓	$FPR = FP/(FP + TN)$	True Negative (TN)
Area Under the Receiver Operating Characteristic curve (AU-ROC) ↑	$\int_0^1 (TPR)\ d(FPR)$	Classification
Area Under Precision-Recall (AU-PR) ↑	$\int_0^1 P\ d(R)$	Localization, Segmentation
Per-Region Overlap (PRO) ↑	$PRO = \frac{1}{N} \sum_i \sum_k \frac{P_i \cap C_{i,k}}{C_{i,k}}$	Total ground-truth number (N), Predicted abnormal pixels (P), Defect ground-truth regions (C)
Saturated Per-Region Overlap (sPRO) ↑	$sPRO(P) = \frac{1}{m} \sum_{i=1}^{m} \min(\frac{A_i \cap P}{s_i}, 1)$	Total ground-truth number (m), Predicted abnormal pixels (P), Defect ground-truth regions (A), Corresponding saturation thresholds (s)
F1 Score ↑	$F1 = 2(P \cdot R)/(P + R)$	Classification
Intersection over Union (IoU) ↑	$IoU = (H \cap G)/(H \cup G)$	Prediction (H), Ground truth (G)/ Localization, Segmentation

Table 2.3.: Description of metrics

2.4. Datasets

The datasets used in image anomaly detection are scarce, especially when it comes to anomaly detection in a manufacturing setting. There are many datsets and approaches that specialize on certain materials [21] [22] [23] and often only one classs. What currently stands out as a gold standard among IAD datasets is the MVTecAD [1] dataset. The authors created it as a highly representative and standardized set of anomalous images along with training images. It has 15 classes from capsules to screws. Moreover the dataset provides image labels aswell as segmentation ground truths, making it versatile and applicable for multiple algorithms. The masks come as black and white grayscale images, while the image labels are given through its folder structure. Its paradigmatic structure tree can be seen in figure xy. As shown, each class contain train images, which only consist of regular examples, and test images. The data among the testing images is categorized by a title describing the anomaly. The ground truth folder contains according ground truths on a pixel level. Example images of the dataset are to be seen in figure z. They typically are of a rectangular shape and their resolutions range from 700x700 to 1024x1024. More specifications can be found in Bergmann et al. [1] and the whole dataset is publicly available at the official website[24].
The MVTecAD[1] dataset is regarded highly among IAD papers, and has since its introduction been used in most relevant papers as a dataaset to benchmark the respective approaches on. This is also likely to remain the trend, since many state of the art algorithms in the recent years have primarily been benmarked on it, forcing new approaches to also be benchmarked on this dataset to be comparable to the current highest performance holding approaches. Despite this work focussing on manufacturing settings MVTecAD is one of only two datasets relevant to this work, and serves as a comparison for the performance investigation of this papers approaches on the second dataset. This is mainly due to the dataset's importance and its relation to the second dataset.
Later in 2022 Bergman et al. has introduced another IAD dataset that is loosely related to their original MVTecAD dataset, namely the MVTecAD LOCO dataset [7]. This dataset works with the same ground ideas as their original MVTecAD set, but extends the conceptual contents of the dataset by logical anomalies(neu formulieren das klingt scheiße). It consists of five classes: breakfast box, juice bottle, pushpins, screw bag and splicing connectors. The difference to the other dataset is that the anomalous categories for each class are only seperated into good images, images with structural anomalies and images with logical anomalies. As mentioned in the introduction structural anomalies are visible damages to the objects, similar to the MVTecAD dataset. Logical anomalies denote violations against arbitrary restrictions imposed by the authors. To illustrate this by an example: The class of pushpins represents a birds view of a compartmentised box of pushpins(see figure a). A rule added was, that each compartment is only to contain one pushpin. This means that if one region were to miss their contents, or contain more than one pushpin, it would constitute a logical anomaly. If on the other hand a pushpin would have a crooked or broken tip, it would be labelled a structural anomaly. Structurally the differences of the MVTecAD and the MVTecAD LOCO dataset can be seen when comparing figures a and b, which showcases the anomaly classification, aswell as the method of storing segmentations. Here there eixists an image file for each anomalous ground truth area, which are mapped to the image by the folder name they are in. Lastly there exists a validation set in this dataset,
The addition of logical constraints opened an interesting area of research, since the high performance of current state of the art algorithms were only measured on structural anomalies so far. Yet it would be insightful to see if those models could also detect logical anomalies, since those also ocurr in real life settings, such as manufacturing settings. Another concept introduced in [7] is the saturated per-region overlap score, also sPRO. The metric is further analysed in setion (metrics section), but in short gives a measure on how well two regions overlap, while also accounting for regions overlapping in a way, that is seen as sufficient. The criterion of sufficiency is given by a file in the respective class, which maps a saturation score to each kind of

anomaly. Bergmann et al.[7] lastly also released a new IAD model together with the new dataset. The model uses autoencoders(bissi besser beschreiben hier). Since the source code has not been made public, this work refrains from using the method proposed in the paper.

2.5. Ensembles

When it comes to ensembling classification models, there are multiple approaches to do so. Many ensembling methods are focussed on combining homogeneous models, meaning a set of related models with similar architecture but different parameters or initializations. Typical methods [25] include averaging, bagging, boosting, and more. There are also new approaches like the CAWPE [26] who try to improve on standard methods. They extend averaging ensembles by a weighted average that is depending on the ensemble members accuracy, yet balances the relevancy of all classifiers so that no strong classifier completely outweighs the rest. Homogeneous ensembles are popular, since they tend to boost the performance and robustness of a base classifier without lots of additional work, since the ensemble is normally created by initializing the models in different ways. Heterogeneous classifier ensembles on the other hand are not necessarily combinable that easily, since they usually consist of models with different network architectures. This can lead to results, that should be interpreted as the same but differ by large margins. Yet ensembles of such variety are often desirable since they offer loads of information from different perspectives or domains when done right. Thus to bridge this gap at the output, a common approach is to first calibrate [27] and then ensemble each models output. For the last combination step, all ensemble techniques suited for homogeneous ensembling can be applied, due to the outputs being in a comparable state then. There are also approaches to collectively calibrate the hyperparameters of each heterogeneous classifier while classifying [27]. While performance varies, combining these models in such a way is not necessarily regarded as the highest achievable robustness, especially when the classifiers work with features or some other form of inner representation. This stems from the fact that the model outputs are merely a small result of larger inner representations that may focus different aspects of information among the inputs. Therefore in turn, you cannot obtain all relevant information that can be offered by simply calibrating the model outputs. A more robust approach to address that problem, would be to ensemble the aforementioned inner representations, i.e. feature maps and in turn train another classifier for the final meaningful output. Another limitation of both kinds of ensembles, being homogeneous and heterogeneous, is that all models have to actually be trained seperately to then utilize the different classification ouputs. This leads to a higher training time and thus also higher computational cost, which is desirable to be reduced in real world manufacturing firms. It should be said that while offering a potential increase in robustness and overall performance, naturally feature level ensemble may also come with certain disadvantages. For instance, it is more difficult to calibrate features from the ensemble members if possible, which may be necessary depending on the nature of the data. An example of our context would be that certain IAD approache project their features into a different space to be effective, making it difficult to ensure that all features are in the same space when dealing with an ensemble. Moreover feature level ensembles also are vulnerable to and reliant on the quality of the input features. This makes the decision on where to cut off the base models very important. The robustness and efficiency has been demonstrated in Heller et al. [28] The authors utilize a feature level ensemble of multiple convolutional neural networks with different architectures and tasks to improve inference speed and accuracy in plant disease detection. They show that cutting off several, potentially heterogeneous, classifiers after a couple of network layers and ensembling the resulting feature maps yields firstly a significant improvement in training time compared to classical output ensembles. This stems from the fact, that all base classifiers of the ensemble only have to be trained once for every following training approach. During this the model still stays compact(zitat aus ensemble

paper markieren), giving it an memory usage advantage over many supervised approaches. Moreover they compared the performance of different ensemble combinations with conventional output ensembles via the softmax function and reported in all cases no significant drop in performance. In cases where this approach allowed for different inputs via multispectral cameras(zitat markieren) there even was a similar performance of this ensemble to other state of the art ensembles visible. Keeping in mind the compactness of this new ensemble model combined with an equal performance and possible increased robustness, as argued prior, it is a promising ensemble approach for this work. To obtain ensembled feature maps the paper proposes to bring all feature maps to the same sizes using bilinear interpolation. Since it is not necessarily desirable to keep every available feature map, as this would create inputs with way too many features, the amount of feature maps is reduced using principle component analysis(PCA). This allows for the ensemble to focus only on the most important features, while maintaining an equal amount of maps as if it were composed of a single classifier. To be more specific Heller et al. [28] introduced two different approaches to perform this ensemble. The first is a global transformation block as seen in figure 2.5. Here the features are first all resized to the same dimensions and then connected along their channels through a concatenation layer. Afterwards PCA is applied along the channel dimension to obtain a result with N remaining feature maps, where N can be adjusted for ones needs. This method offers the advantage of efficiency, as PCA is only run once per feature ensembling, and may be applied when there is an almost even number of feature maps per classifier with same input representations(erklärlicher schreiben?). Yet this approach is also prone to a couple disadvantages. If the different input data is collected from fundamentally different sources, there may be a significant loss of information when globally applying PCA. Furthermore this approach cannot be balanced when confronted with classifiers with large discrepancies in channel number. If the amount of feature maps from one classifier completely predominates, there is a high likelihood that most feature maps that are selected are from this classifier, if not all.

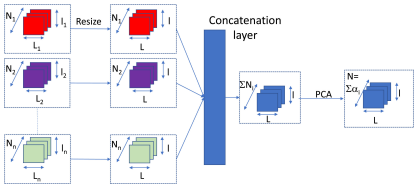


Figure 2.5.: Hier korrekt zitieren

To combat this at a cost of lesser efficiency Heller et al. also introduced a second approach, namely the independent transformation block, visualized in figure 2.6. This procedure is only differing in the sequencing of the actions. Therefore PCA is firstly applied to every set of feature maps, keeping a certain number of feature map components per classifier. They are then all resized to the desired dimensions and concatenated through the concatenation layer. This sequencing allows for maximum information preservation through individual PCA and also to predefine the number of feature maps to be kept per classifier, preventing larger imbalances.

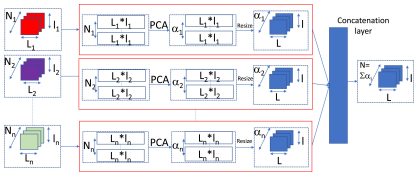


Figure 2.6.: Hier korrekt zitieren

2.6. Model Calibration

Calibration or rather confidence calibration is the process of adjusting/scaling your models ouput so that it indicates how likely it is to be correct. This is an important action, as nowadays models demonstrate increasingly good performance, scoring very high accuracies on classification tasks. Models that show to be correct that often, generally also have porential to be deployed in real world use cases, which especially holds true in IAD research for manufacturing contents. [27] reports that most modern neural networks are poorly calibrated in regard to their confidence. Current IAD methods investigated in this work confirm this, only returning anomaly scores devoid of any confidence indication. Correct confidence calibration can help evaluate models using new metrics, increase the users trust into the application and also help to decide on how to utilize the predictions of the model in ones context [29].
[27] review multiple promising ways to calibrate the confidence of a model. The paper assumes certain components as method input. For a sample x_i there exists the prediction probability $\hat{p}_i$ that $y_i = 1$, which is poorly calibrated. Moreover we are given the according non probabilistic output or logit z_i for the input. The goal is to derive a well calibrated output confidence $\hat{q}_i$. On a top level, the authors distinguish between calibrating binary classification models and multiclass ones. For binary models they present histogram binning, isotonic regression, bayesian binning into quantiles (BBQ) and lastly platt scaling. Histogram binning is a non-parametric approach and involves binning the uncalibrated prediction possibilities into distinct bins. Each bin B_m is then assigned a confidence score θ_m. During test time the models prediction probability $\hat{p}_i$ is mapped the score θ_m of the bin B_m it falls into, resulting in a calibrated confidence $\hat{q}_m = \theta_m$. The boundaries of the bins are here chosen so that they minimize the bin-wise squared loss in accord to:

Another non parametric approach is the solution with isotonic regression. This means to learn a piecewise constrant function $\{ \ni \hat{q}_m = \{(x_i)$. This may serve as a optimized strict generalization of the histogram binning approach as the function can be written as: Another extension of histogram binnning would be BBQ, which marginalizers out all possible binning schemes to produce $\hat{q}_i$ (entweder zutat kennzeichen oder neu formulieren). A binning schemes is denoted by [27] to be a pair $(M, \mathcal{I})$, with M being the number of bins and $\mathcal{I}$ a corresponding partition of $[0, 1]$ into disjoint intervals.

Lastly there is the parametric approach for binray prediction classifiers called platt scaling. This is the only approach that doesn't require the uncalibrated predicted probabilty and solely utilizes the classifiers' logits to infer a calibrated confidence. The approach can be summarized with fitting a logistic regression model on the uncalibrated non-probabilistic model outputs. An exemplary solution in the context of neural networks may be to optimize two scalar parameters $a, b \in \mathcal{R}$ so that $\hat{q}_i = \sigma(az_i + b)$ achieves an optimal fit. [27] stress that this method only calibrates the output after training, while the networks parameters stay fixed during this calibration aswell as thereafter.
As mentioned the paper also touches on model calibration for multi class prediction models, like certain binning method extensions or matrix and vector scalings. These will not be discussed here, since the problem context of this work revolves around binary classificaion methods.

3. Novel Dataset Class

In this chapter we present the novel dataset category representing a flat connector. Mainly the sections will revolve around the specifics of the flat connector object, the setup used to record the data and produce labels, as well as the anomalies that can be incorporated into this object. Section 3.3 also reviews the steps that went into producing the specific anomalies and how to recreate them.

3.1. Flat Connector

As mentioned prior to this chapter, this work will also deal with the introduction of a novel dataset category that is an extension of the MVTecAD LOCO [7] dataset. Despite simply extending the range of objects covered in the dataset, this also serves a more various investigation of IAD method performance in industrial settings. The motivation behind adhering to MVTecAD LOCO standards is multifold. Firstly the setting of this work is already in the logical anomalies context, which greatly facilitates the evaluation of the new class. Additionally, as discussed in section 2.4, there is only a slight technical differnce between the MVTecAD LOCO and the classical MVTecAD [1] dataset, meaning that this novel class can easily be evaluated by the vast majority of IAD approaches, as most of them already report their performance on the MVTecAD dataset. This holds true for past approaches but also new ones to come.

The flat connector object was chosen for this since it meets multiple requirements for an adequate object class. It is of metal manufacturing natrue, and can comfortably be photographed from an overhead perspective. Although the MVTecAD LOCO dataset already contains a screw bag, the property of the plastic bag could potentially shift the focus of the performance from solely the metal part perrformance to the anomaly localization in more difficult conditions. in the The latter is a common characteristic of the other classes in the set. Finally the flat connector has some properties that make it a very favourable manufacturing part to include: It is of a lighter metal, giving us much more freedom to produce anomalies. A steel block for example would be vastly harder to meaningfully process into it representing different kinds of anomalies. Moreover the differently sized holes in set configurations that the flat connector posesses, make for a lot of opportunities to introduce logical anomalies without the need of arbitrary rules like the pushpin or splicing connector class. For example the latter class may need multiple objects in a certain arrangement per image to then predefine rules as logical constraints, whereas the flat conenctor as such may posess multiple logical violations within only one object without the need of additional artifical rules.
Further regarding the flat size specifications, we used regulatory flat connectors of size $100x35mm$ which are widely available at hardware stores and at [30]. The surface of the flat connectors is galvanized and displays a CE-label acoording to DIN EN 14545. Images of normal flat connectors without manipulations can be viewed in figure xyz.

3.2. Data Acquisition Setup

The images for the dataset were taken in the university facilities. An exemplary setup for image acquisition can be viewed in figure xyz(images von setup). The contraption to capture images was done by mounting a camera on to a pole that was attached to a small table, pointing the camera lens at the table from an overhead perspective. The distance between camera lens and the table surface was approximately (angabe) cm and the main light source in the images was delivered by a softbox(stimmt das?). Regarding the background of the object, a black cloth was chosen to be layed flat on the table, where the object would then be put. For an increased variety, multiple flat connectors were bought and filmed. Here each one was manually rotated after each shot, to acheive different object orientations and thus a more robust training set. Each flat connector was rotated a total of six times per side, resulting in 12 normal images per flat connector object.
The camera used to take pictures was a (camera modell) which produced the images of the flat connectors in a (pixel angaben) resolution. (guccken obs stimmt) Due to the large size of the resulting images, they were resized to (pixel angaben) with amounts to roughly (prozen angabe) % of the original image size.
To obtain segmentation labels, a free polygon annotation software was used to then label anomalous regions in the images. The annotations were done by hand and by the author of this work. In accord to MVTecAD LOCO standards [7] each kind of anomaly would be assigned a different pixel value to differentiate them when working with saturation thresholds as mentioned in setion 2.4. Example annotations can be viewed in figure xyz, where several original images are depicted next to the produced segmentation labels. Just like in the original MVTecAD LOCO dataset, image labels were not explicitly given since they can be inferred by the anomaly class.

3.3. Anomalies

The common MVTecAD LOCO clases all posess a somewhat balanced amount of logical and structural anomalies. Therefore the collection of possible anomalies for this class was kept at an even structural to logical ratio. The anomalies listed below can be viewed in figure xyz where exemplary anomalous images are displayed next to their corresponding ground truth mask.

3.3.1. Structural Anomalies

Structural anomalies in this case were comparably simple to think of an execute since they only involved damaging some area of the flat connector. Here we also tried to represent a variety of certain characteristics of anomalies among the ones we chose. An example would be to include anomalies with larger areays as well as ones with small ground truths.
First of we produced the structural anomaly of a cut off corner. Ths was done by simply using a metal saw to remove a corner of the flat connector. This is an anomaly with a comparably large surface area and clean edges to predict. Next we again used a etal saw to damage the edges of the flat connector by cutting out pieces of or simply sawing into the edges of the object. One image of this anomaly type contained multiple damages spots on multiple edges. Due to the cuts partially being very small, this makes for an anomaly with very small ground truth areas as seen in figure xyz. As a last structural anomaly the saw was used to apply deep scratches on the object. This is mimicing another type of anomaly sometimes found in the classes of splicing connectors and pushpins in the original dataset. The common characteristic is that the region is very slim and long, making for better comparisons.

3.3.2. Logical Anomalies

The logical anomalies comprise the more interesting part of the anomaly types, as they are the focus of the MVTecAD LCOO dataset. The first logical anomaly in this class is the case in which a hole is missing in the flat connector hole pattern. This anomaly had to be improvised, since it is difficult to obtain a properly manufactured flat connector that was misproduced. To still include this type of anomaly, the solution was to first fill the hole with moldable material and then spray paint over it to hide color differences. For realism the spray paint color was the same as the object surface and the whole object was spray painted to avoid color artifacts. Partialy this anomaly was one of the larger and more clean cut ones since the holes mostly chosen to be filled were one of the two bigger ones on the object. For increased variety there were also anomlaous cases added where a small hole would be concealed. As a second anomaly the case of an extra hole was realised. To produce such an anomaly the facilities offered a drill, which was used to drill another small hole into the object at an unusual location. The size of the additional hole was the same as the smaller ones on the object. To also test capabilities of detecting multiple anomalies in a single image, this anomaly was put together with an differently sized hole. The final anomaly to be intruduced was a differently sized hole at a correct location. This would mean that the anomaly does not consist of the hole location but rather the size difference between this holes and the other ones in similar places. For this type of anomaly it is admissible to predict the whole hole as an anomalous region. Here a drill tip was used with a slightly higher radius that the original hole to enlarge the diameter.

3.3.3. Saturation Thresholds and Outlook

The saturation scores for the anomalies, as discussed in section 2.4 were put at the amount of pixels in the anomalous region for all above listed anomalies respectively. This was because the nature of presented anomalies of this categories calls for a full segmentation of the respective anomaly for a perfect result. Unlike in the pushpin example given in that section there are no cases that warrant multiple possible placements of anomalies.

4. Methods

This chapter will cover the specific implementation details of the concepts researched in this thesis. To reiterate, this work conducts a systematic investigation on performance of previous state of the art IAD approaches on logical anonomalies and interprets the results. Secondly this paper introduces a feature level ensemble approach for combining potentially heterogeneous anomaly detection methods to achieve greater robustness. The just named contributions are thematized in regards to their execution in the respective sections 4.1 and 4.2 of this chapter.

4.1. Logical Anomaly Detection

As discussed in the introductory chhapter 1, logical anomalies represent a signifacant part of image anomaly detection in modern manufacturing settings. The experiments also serve as an extensive comparison of SOTA methods for IAD versus recent approches that where introduced with special mind to logical anomalies, like GCAD [7]. Moreover, for a qualitative evaluation of the performance change when using feature level ensembles, one first needs to evaluate the base performance of each relevant classifier of the set. Hence this work features experiments to evaluate IAD approaches mainly evaluated on the classical MVTecAD dataset. To do so, the original code from each paper was taken and not modified in regards to any reportet parameters and/or arguments. This was to prevent possible unwanted deviations in original performance by changing up synergies of hyperparameters. This paper recognizes the possibility of improved performances on the logical anomalies dataset with different combinations of model parameters. Yet this work focusses on the result assessment of current unmodified apporoaches and more importantly the increased robustness through the use of ensembles. Therefore research regarding this hypothetical improvement would have to be done in a future work. Metrics that are specifically looked at in this context are the AUROC, pixel AUROC(weitere maybe einfpgen) and the sPRO. If the functionality to evaluate these metrics was already given, the results of inference were taken from the original code, else the according functionality was implemented in this work and used to produce the according metrics. Moreover the results investigate possible causes and effects regarding the segmentation/localization results of the classifiers. This is not done according to an official metric but in a more descriptive sense. Papers whose approaches were evaluated using the MVTecAD LOCO dataset were: SimpleNet [3], PatchCore [10], [4] and [5]. (list of paper references with names). These papers were discussed in more depth in the backgrounds section and any specifics like hyperparameters can be viewed in the corresponding paper. Furthermore all named classifiers were including, among other variable measures, a preprocessing step to resize the input image. This makes for a variable model input and also the ability to process rectangular images, which is important due to MVTecAD LOCO images being rectangular unlike the squared input from the standard MVTecAD dataset. The only necessary modification to the whole process of anomaly detection was the generation of image masks. The MVTecAD LOCO dataset stores its masks in multiple seperate black and white images, one for each individual anomaly. To fix errors stemming from this fact, additional code was added that pastes all masks belonging to one image into a single mask before

iterating through the data. There also exists a minor ablation experiment experimenting with the elimination of possible background artifact removal on images of our novel dataset category. For this we programmatically set every pixel in a certain radius around the main object to black, to investigate segmentation artifacts of certain classifier methods. The results can be viewed in section xyz.

4.2. Ensemble network

!abändern!

The following subsections deal with the individual components of our ensemble network pipeline, referencing concepts discussed in chapter 2. This network is conceptually based on different approaches. First we produce cut off models and an ensembling mechanism in accord to [28]. For the actual IAD process, this approach is based on the SimpleNet method [3]. The discriminator presented there potentially makes for an easy to implement, yet powerfull method of differentiating between anomalous and normal images, as seen by the performance of the original SimpleNet approach.

4.2.1. Ensemble Members

!Neu schreiben!

The first exploration of the ensemble method is done with the IAD methods PatchCore [10] and SimpleNet [3] as ensemble members. These two are conveniently both creating features in form of locally aware patches, making for easy preprocessing before ensembling. For combining them on a feature level, we need to intersect the trained models somewhere during the inference process to then exctract accurate representations that can be used. For SimpleNet this cut-off is happening after the extracted features are projected through the trained feature adapter (the green tiled pane in figure xyz). PatchCore requires more work, as the feature extraction and patch converting steps are essentially the same to the ones in SimpleNet. Moreover there are no alterings of resulting feature vectors after these steps, only nearest neighbor search. Yet solely utilizing features extracted from the pretrained feature extractor would not be sensible, since it would not add any new information to the ensemble that was not already represented in SimpleNets patch vectors. The approach realized in this work is based on the way patchcore scores its image anomalies. Since its decisions are fundamentally based on the L2 norm of the difference between newly extracted features and the nearest feature vector in the memory bank (see equation 2.3), it is to be assumed that there is information in the differences of the vectors. Thus the feature representation m_i^{final} for a image patch p_i is derived as $m_i^{final} = |m_i^{test} - m|$ where $m \in \mathcal{M}$ with $\mathcal{M}$ as the memory bank, m_i^{test} as the according input patch vector and m as the neirest neighbor of m_i^{test}. Additionally we introduce a projection layer similar to [3] which is supposed to help project the features taken from this ensemble member to be in a similar space as the ones by SimpleNet, which are already projected from the start. The nature of this projection layer is again a single layer network, being applied to the difference vectors.

4.2.2. Feature Level Ensembling

!Das stacken mit einbinden!

For combining the feature representations from different ensemble members, we chose the approach of the individual transformation block (see figure xyz, section 2.5). This means PCA was first performed on each set of feature vectors, before then resizing and concatenating them. The PCA was performed via the scipy library and fitted on the training data. As for resizing, the feature vectors were bilinearly interpolated to the largest diimensions of the ensemble candidates. Moreover the amount of feature maps kept per classifier, namely α in figure xyz (igure mit IFT), was derived by dividing up the number of feature maps from the first classifier in the list of ensemble members, which in this instance also was the maximum amount. If the number was not evenly divisible, excess maps would be split up among the remaining classifiers as evenly as possible. (vllt mal weniger maps nehmen? und anhand von explained variance gucken).

4.2.3. Ensemble Training

Our approach to use a small, compact discriminator to differentiate between regular and anomalous image features is based on the concept of the one-class classification class from the representation based approaches. Specifically the main inspiration of this work is the approach presented in SimpleNet [3]. Since our discriminators inputs in the ensemble pipeline, which stems from ensembled locally aware patch features, will be of the same nature as the inputs for SimpleNet's discriminator, it is reasonable to utilize their network architeture for this work. Looking back at section 2.2.2 and moreso figure 2.2, we thus will adapt the SimpleNet pipeline at the feature adapter step. This means, we will train a global feature adapter that ensures the projection of ensembled features into the right latent space. Additionally at train time a lightweight discriminator is trained, which then again differentiates between normal ensembled features and artificially created abnormal ones. Unlike in the originial approach, the input to the feature adapter does not consist of only pre extracted image features, but the ensembled features from section 4.2.2. The artificial anomalous features, depicted as the red tiled pane in figure 2.2 will also be provided during training time. Here we also adapt SimpleNet's approach of gaussian noise for producing those artificial features. (Simplex noise???). The discriminator is expected to provide positive and negative outputs for regular and anomalous features respectively. As to the discriminator network specifics, a regular fully connected network consisting of two layers is used. As optimizer, this work utilizes the adam optiizer by pytorch with a learn rate of (werte erst sauber aufschreiben bevor ich es hier hinschreibe). The loss is derived the same way it was in the approach that inspired this procedure, which is according to equation 2.4.

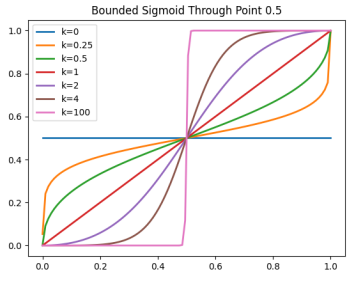
4.2.4. Calibration

As discussed in section 4.2.4, calibration can greatly improve understanding and usability of classifiers. Anomaly models researched in this work only return non-probabilistic outputs. These can be thresholded to derive decision critera but do not represent any confidence of the model at all. When looking back at [27], the only calibration methods for binary classifiers that did not need a probabilstic output was the platt scaling attempt. Therefore this basic principle was the method of choice for calibrating the ensemble networks outputs. It is to be noted here, that we merely calibrate the models final outputs after everything has been predicted. This stems from the fact, that [31] demonstrated how calibrated ensemble members do not necessarily yield a well calibrated final output, shifting our focus for calibration application. During the process, it became apparent that the platt scaling approach from [27] was not necessarily sufficient, as figure xyz demonstrates. The figure displays 20 predicted scores of the "broken large" anomaly of the bottle class of the MVTecAD dataset, along with the predicted anomaly scores of 20 normal images Due to the high performance of some IAD algorithms it is possible that some categories of the datasets are fully correctly classified, which in example

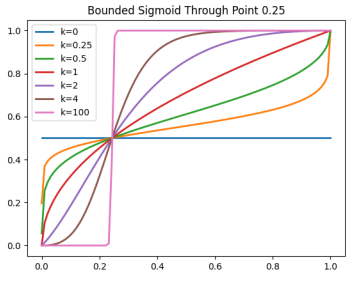
was the case for . This would mean, that the optimal fitted sigmoid curve on the points is a step curve, yielding either 100 percent confidence or none at all. This of course does not seem well calibrated, especially not looking at the exemplary score distribution in figure xyz, where it apprears that the predicted anomaly scores seem to be stretched pretty evenly over the interval. To overcome this problem we propose a generalized bounded sigmoid function with a varibale slope [32] as seen in equation 4.1. This allows the user to calibrate the steepness of the slope to account for larger uncertainty towards the decision threshold.

$$s_{k,t}(x) = \frac{1}{1 + (x^{\frac{log(2)}{log(t)}} - 1)^k} \tag{4.1}$$

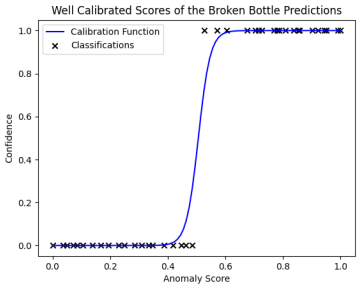
T denotes the point where the function has to pass through. This value is usually assigned the calculated threshold for the decision criterion of the classification task. K is a parameter that influences the growth of the curve along the axis. This value can either be chosen per hand or optimized in regards to uncertainty requirements. This means that if you generally want the ouputs to reflect a higher uncertainty towards the threshold it is resonable to chose a lower value for k and vice versa. Figure xyz demonstrates the effects of different values for k, as well as an exemplary well calibrated platt scaling variant of figure xyz, using the presented formula.



(a) Caption for Image 1



(b) Caption for Image 2



(c) Caption for Image 3

Figure 4.1.: Main caption

This caliration method is provided in the pipeline to produce an uncertainty metric. Yet in the light of the reported metrics of other IAD research the experiments and discussion will only review the same metrics, namely image and pixel AUROC and the PRO, or rather sPRO, score.

5. Experiments

In this chapter the experiments and experimental results of this work are displayed. First we establish the general experimental setup. Afterwards the results of the conducted MVTecAD LOCO [7] survey are presented in section 5.2. As a baseline for performance evaluation we uitilize the performance on the more conventional MVTecAD dataset [1], as well as a classifier comparison. In section 5.3 we review the performance of the classifiers on our novel dataset category. Lastly section 5.4 deals with the findings regarding the ensemble network approach, also conducted on the MVTecAD LOCO dataset.

5.1. Experimental Setup

All models trainings and result reproductions have been conducted on the IAD cluster student partition. The GPU in use for all nodes used by that partition (ist die formulierung gut?) is and RTX 2080Ti with 11GB of memory, and the CPU is an AMD Ryzen 9 16-Core processor. The cluster overview [33] serves to provide further detail for additional questions. As for software, pytorch 2.1.2 was utilized to implement the ensemble model. The specifications of other libraries, as well as the specifications for the MVTecAD LOCO experiment are documented in the environment files in the project code.

5.2. MVTecAD LOCO Experiments

In this section we review the performance of prior introduced anomaly detection methods. All experiments were performed with the same conditions explained in section (referenz von methods section über loco) and on the MVTecAD LOCO dataset [7]. The results of inference on the test set can be seen in table x (tabelle mit ergebnissen). As it can be seen, all models scored a significantly lower result on the MVTecAD LOCO dataset than on the normal MVTecAD one(exemplary scores seen in table xy(table mit normalen mvtec scores)). A lower performance is generally to be expected, since firstly logial anomalies are regarded as a more difficult problem than structual ones and secondly the average SOTA performances as seen in table x(tabelle mit ergebnissen) is already closing in on an AUROC of of 1. (den satz rechts von hier müsste man maybe rausmachen oder umschreiben)Therefore there is not much room for further improvement in similar settings, and a worse performance still aknowledgeable as very good. Yet there is an drop in cross-model average AUROC of approcimately (durchschnitts drop ausrechnen), which is a remarkable(synonym) difference. Most other metrics, namely (metrics names), also declined with an respective average of (respective averages). As explained in section (referenz zu metrics section von background), the sPRO (or rather AU-sPRO) was a score introduced in [7] to gain an advanced insight on the quality of segmentations. This means that all approaches who either were published before or did not include this paper in their research likely did not include this metric, which holds true for the approahces used for this experiment. Therefore no comparison in

sPRO/AU-sPRO can be shown(vllt einfach spro auch für allte ansätze implementieren?? dann kann ich den satz ändern). Comparing the sPRO scores of the SOTA methods in this experiment with the ones from compared to GCAD [7] shows asignificantly (abchecken ob wirklich) worse performance. Among the different models, the highest scoring one was PatchCore [10]. It scored an average (metrics einfügen) feature embedding based approaches like achieved the highest scoring

Interpretation of results hier, weiß nicht in welche section das eigentlich muss:

5.3. Flat Connector Experiments

5.4. Ensemble Network

6. Experimental Results

- analysis on how methods worked on own dataset individually -> if poor performance error analysis and also address different subclasses

- analysis of how ensemble model worked and if it improved performance

6.1. SOTA Methods Performance on classical LOCO Dataset

In this section we review the performance of prior introduced anomaly detection methods. All experiments were performed with the same experimental setup as explained in section (referenz of experimental setup section), the conditions explained in section (referenz von methods section über loco) and on the mvtec LOCO dataset [7]. The results of inference on the test set can be seen in table x (tabelle mit ergebnissen). As it can be seen, all models scored a significantly lower result on the MVTecAD LOCO dataset than on the normal MVTecAD one(exemplary scores seen in table xy(table mit normalen mvtec scores)). A lower performance is generally to be expected, since firstly logial anomalies are regarded as a more difficult problem than structual ones and secondly the average SOTA performances as seen in table x(tabelle mit ergebnissen) is already closing in on an AUROC of of 1. (den satz rechts von hier müsste man maybe rausmachen oder umschreiben)Therefore there is not much room for further improvement in similar settings, and a worse performance still aknowledgeable as very good. Yet there is an drop in cross-model average AUROC of approcimately (durchschnitts drop ausrechnen), which is a remarkable(synonym) difference. Most other metrics, namely (metrics names), also declined with an respective average of (respective averages). As explained in section (referenz zu metrics section von background), the sPRO (or rather AU-sPRO) was a score introduced in [7] to gain an advanced insight on the quality of segmentations. This means that all approaches who either were published before or did not include this paper in their research likely did not include this metric, which holds true for the approahces used for this experiment. Therefore no comparison in sPRO/AU-sPRO can be shown(vllt einfach spro auch für allte ansätze implementieren?? dann kann ich den satz ändern). Comparing the sPRO scores of the SOTA methods in this experiment with the ones from compared to GCAD [7] shows asignificantly (abchecken ob wirklich) worse performance. Among the different models, the highest scoring one was PatchCore [10]. It scored an average (metrics einfügen) feature embedding based approaches like achieved the highest scoring

Interpretation of results hier, weiß nicht in welche section das eigentlich muss:

6.2. Ensemble Performance

Notizen für diese section: - hier soll reportet werden wie das ensemble sich geschlagen hat - dazu brauche ich: -> metriken(AUROC sPRO und vllt pixel auroc) von dem ensemble auf flat connector + mvtec loco -> beispielhafte segmentierungen -> plots von loss und auroc über training

- drauf eingehen wo sich das ensemble wie gut geschlagen hat -> vergleich mit patchcore und simplenet wichtig, gerne auch mit DRAEM vergleichen als reconstruction representativer algo -> sagen bei welchen klassen es gut und nicht so gut geklappt hat, vergleichen mit ergebnissen aus LOCO studie oben drüber(vllt in conclusion?) -> mehr images in appendix anbieten

7. Conclusion and Future work

7.1. Ensemble Network

7.2. SOTA performance

7.3. Flat connector

7.4. Outlook

- schreiben die einbindung von andern approaches ins ensemble wie das füttern von patchcore mit diesen features - schreiben was es noch für coole dataset categories gab -> multiperspective angle und blech konstruktion mit schrauben

Bibliography

[1] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection," *International Journal of Computer Vision*, vol. 129, p. 1038–1059, Jan. 2021.

[2] J. Liu, G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, and Y. Jin, "Deep industrial image anomaly detection: A survey," *Machine Intelligence Research*, vol. 21, no. 1, pp. 104–135, 2024.

[3] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "Simplenet: A simple network for image anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20402–20411, 2023.

[4] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, "Fully convolutional cross-scale-flows for image-based defect detection," in *Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022.

[5] V. Zavrtanik, M. Kristan, and D. Skocaj, "DrÆm – a discriminatively trained reconstruction embedding for surface anomaly detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2021.

[6] T. D. Tien, A. T. Nguyen, N. H. Tran, T. D. Huy, S. T. Duong, C. D. T. Nguyen, and S. Q. H. Truong, "Revisiting reverse distillation for anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24511–24520, June 2023.

[7] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization," *International Journal of Computer Vision*, vol. 130, p. 947–969, Feb. 2022.

[8] G. Xie, J. Wang, J. Liu, J. Lyu, Y. Liu, C. Wang, F. Zheng, and Y. Jin, "Im-iad: Industrial image anomaly detection benchmark in manufacturing," *IEEE Transactions on Cybernetics*, vol. 54, p. 2720–2733, May 2024.

[9] W.-H. Chu and K. M. Kitani, *Neural Batch Sampling with Reinforcement Learning for Semi-supervised Anomaly Detection*, p. 751–766. Springer International Publishing, 2020.

[10] K. Roth, L. Pemula, J. Zepeda, B. Scholkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 14318–14328, Jun 2022.

[11] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, p. 3964–3979, Nov. 2021.

[12] G. Xie, J. Wang, J. Liu, J. Lyu, Y. Liu, C. Wang, F. Zheng, and Y. Jin, "Benchmarking anomaly detection algorithms," *Journal of LaTeX Class Files*, vol. 18, no. 9, 2020.

[13] Z. You, K. Yang, W. Luo, L. Cui, Y. Zheng, and X. Le, *ADTR: Anomaly Detection Transformer with Feature Reconstruction*, p. 298–310. Springer International Publishing, 2023.

[14] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arxiv:2006.11239*, 2020.

[15] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, June 2022.

[16] H. Zhang, Z. Wang, Z. Wu, and Y.-G. Jiang, "Diffusionad: Norm-guided one-step denoising diffusion for anomaly detection," *arXiv preprint arXiv:2303.08730*, 2023.

[17] T. Defard, A. Setkov, A. Loesch, and R. Audigier, *PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization*, p. 475–489. Springer International Publishing, 2021.

[18] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, June 2022.

[19] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but differnet: Semi-supervised defect detection with normalizing flows," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2021.

[20] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, p. 600–612, Apr. 2004.

[21] C. S. Tsang, H. Y. Ngan, and G. K. Pang, "Fabric inspection based on the elo rating method," *Pattern Recognition*, vol. 51, p. 378–394, Mar. 2016.

[22] D. Yang, Y. Cui, Z. Yu, and H. Yuan, "Deep learning based steel pipe weld defect detection," *Applied Artificial Intelligence*, vol. 35, p. 1237–1249, Sept. 2021.

[23] Y. Huang, C. Qiu, Y. Guo, X. Wang, and K. Yuan, "Surface defect saliency of magnetic tile," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, IEEE, Aug. 2018.

[24] "Download page of the mvtec ad dataset."

[25] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, p. 169–198, Aug. 1999.

[26] J. Large, J. Lines, and A. Bagnall, "A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates," *Data Mining and Knowledge Discovery*, vol. 33, p. 1674–1709, June 2019.

[27] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," ICML, 2017.

[28] G. Heller, E. Perrin, V. Vrabie, C. Dusart, M.-L. Panon, M. Loyaux, and S. Le Roux, "Multisource neural network feature map fusion: An efficient strategy to detect plant diseases," *Intelligent Systems with Applications*, vol. 19, p. 200264, Sept. 2023.

[29] S. McGrath, P. Mehta, A. Zytek, I. Lage, and H. Lakkaraju, "When does uncertainty matter?: Understanding the impact of predictive uncertainty in ml assisted decision making," p. 1237–1249, Transactions on Machine Learning Research, 2023.

[30] "Link to the flat connector that was recorded." https://www.hornbach.de/p/flachverbinder-100-x-35-mm-sendzimirverzinkt-1-stueck/735660/. Accessed: 2024-04-30.

[31] X. Wu and M. Gales, "Should ensemble members be calibrated?," ICLR, 2021.

[32] "Bounded sigmoid curve with varibal slope." https://math.stackexchange.com/questions/1832177/sigmoid-function-with-fixed-bounds-and-variable-steepness-partially-solv Accessed: 2024-05-03.

[33] "Overview of the cluster." https://pointed-wakeboard-5f3.notion.site/Overview-of-the-Cluster-c74c945e34274f4ea9a964a805cc7d7f. Accessed: 2024-04-30.

A. Appendix

Appendix here