Karlsruhe Institute of Technology

# Anomaly Detection in the Latent Space of VAEs

Bachelor Thesis


Simon Klaus

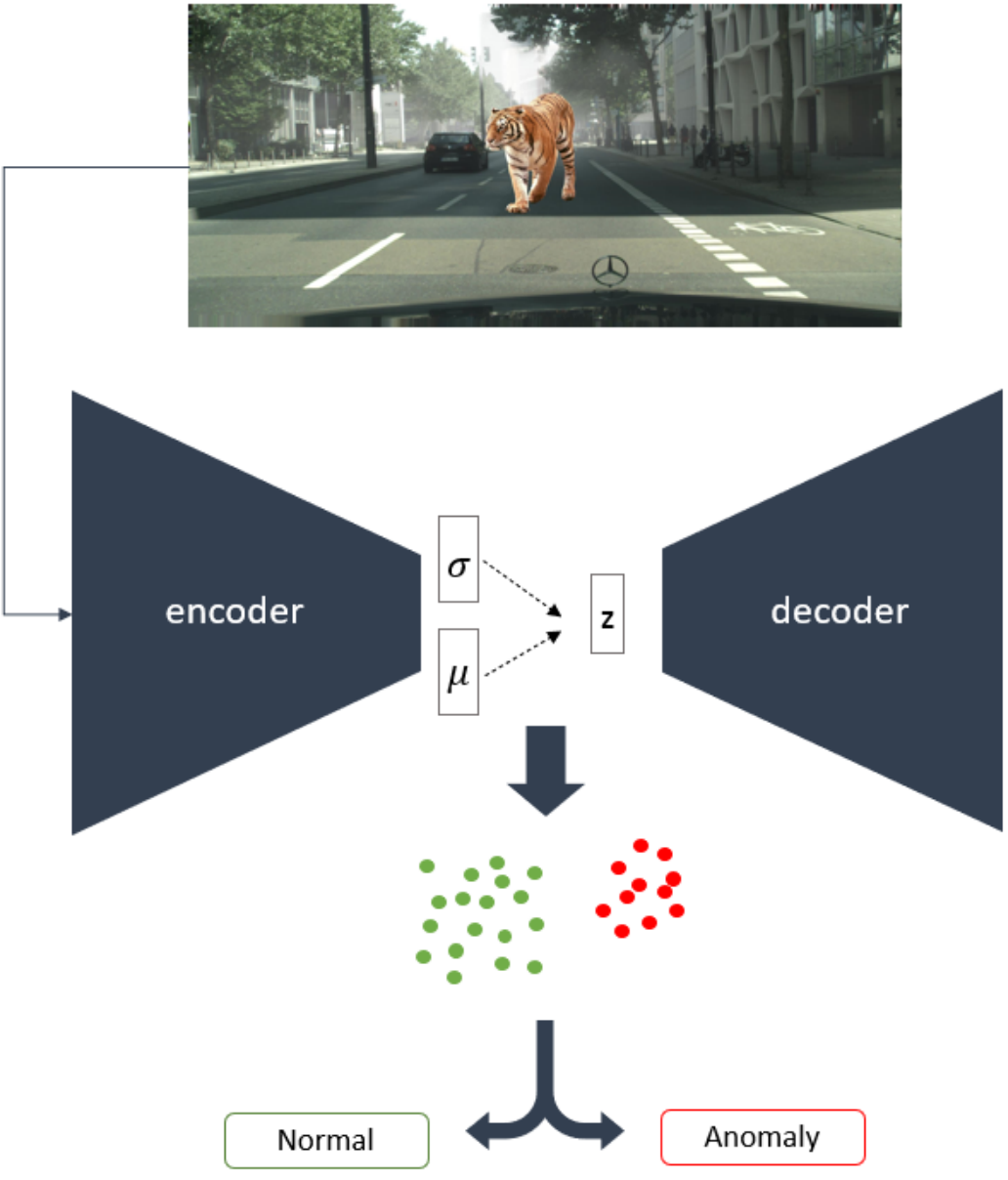
Department of Economics and Management
Institute of Applied Informatics and Formal Description Methods
and
FZI Research Center for Information Technology


| | |
|---|---|
| Reviewer: | Prof. Dr.-Ing. J.M.Zöllner |
| Second reviewer: | Prof Dr.-Ing. A. Oberweis |
| Advisors: | M. Sc. Daniel Bogdoll |
| | M. Sc. Svetlana Pavlitskaya |

Research Period: 01. April 2022 – 04. October 2022

# Anomaly Detection in the Latent Space of VAEs

by
Simon Klaus



**Bachelor Thesis**
October 2022

Bachelor Thesis, FZI
Department of Economics and Management, 2022
Reviewers: Prof. Dr.-Ing. J. M. Zöllner, Prof. Dr.-Ing. A. Oberweis

Abteilung Technisch Kognitive Assistenzsysteme
FZI Research Center for Information Technology

## Affirmation

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, *Simon Klaus*
October 2022

# Abstract

One of the most important challenges in the development of autonomous driving systems is to make them robust against unexpected or unknown objects. Many of these systems perform really good in a controlled environment where they encounter situation for which they have been trained. In order for them to be safely deployed in the real world, they need to be aware if they encounter situations or novel objects for which the have not been sufficiently trained for in order to prevent possibly dangerous behavior. In reality, they often fail when dealing with such kind of anomalies, and do so without any signs of uncertainty in their predictions. This thesis focuses on the problem of detecting anomalous objects in road images in the latent space of a VAE. For that, normal and anomalous data was used to train the VAE to fit the data onto two prior distributions. This essentially trains the VAE to create an anomaly and a normal cluster. This structure of the latent space makes it possible to detect anomalies in it by using clustering algorithms like k-means. Multiple experiments were carried out in order to improve to separation of normal and anomalous data in the latent space. To test this approach, anomaly data from multiple datasets was used in order to evaluate the detection of anomalies. The approach described in this thesis was able to detect almost all images containing anomalous objects but also suffers from a high false positive rate which still is a common problem of many anomaly detection methods.

# Kurzfassung

Eine der größten Herausforderungen bei der Entwicklung von autonomen Fahrsystemen besteht darin, sie robust gegenüber unerwarteten oder unbekannten Objekten zu machen. Viele dieser Systeme funktionieren gut, solange sie sich in einer kontrollierten Umgebung bewegen, in der sie ausschließlich auf Situationen treffen, für die sie trainiert wurden. Damit sie sicher unter realen Bedingungen eingesetzt werden können, müssen sie jedoch in der Lage sein zu erkennen, wenn sie auf Situationen oder Objekte treffen, für die sie nicht ausreichend trainiert wurden. Unerkannt können solche Situationen zu möglicherweise gefährlichem Verhalten führen. In der Realität scheitern sie daran jedoch oft, ohne dabei Anzeichen für Unsicherheit zu zeigen. Diese Arbeit befasst sich mit der Erkennung von Anomalien in Verkehrsbildern im latenten Raum eines VAEs. Um dies zu ermöglichen, wurde der VAE so gestaltet und trainiert, dass sich Anomalien und normale Daten in seinem latenten Raum um zwei verschiedene Normalverteilungen sammeln. Dadurch wird im latenten Raum des VAEs ein anomales und ein normales Cluster erstellt. Diese Struktur des latenten Raums ermöglicht es, mithilfe von Clustering-Algorithmen wie k-means Anomalien in diesem Raum zu erkennen. In dieser Arbeit wurden mehrere Methoden getestet, um die Trennbarkeit von Anomalien und normalen Daten zu erhöhen. Um diesen Ansatz zu bewerten, wurden mehrere Datensätze verwendet, welche Anomalien beinhalten, um zu testen, ob diese erkannt werden. Der in dieser Arbeit beschriebene Ansatz war in der Lage, fast alle Anomalien als solche zu erkennen, jedoch produziert er auch eine hohe Falsch-Positiv-Rate, was ein häufig auftretendes Problem vieler Methoden zur Anomalie Erkennung ist.

# Contents

# 1 Introduction

During the last years there have been tremendous advances in machine learning using Deep Neural Network (DNN) which also led to big advancements in autonomous driving systems. While a lot of machine learning models are trained on a closed world assumption, which means the test data is assumed to be drawn from the same distribution than the training data [87], this is not the case in real world applications. In order for autonomous driving systems to be used in real-life situations, they must be robust against unexpected safety critical situations. The field of Anomaly Detection (AD) focuses on the detection of scenes for which the model was not sufficiently trained for. These rare scenes can occur in an open world environment while not being represented in the data used to train the model and therefore pose a threat to safety critical applications. While there are multiple definitions of the term scene [57, 28], this thesis follows the one used by [77] where a scene is defined as a snapshot of the environment which includes scenery, dynamic elements and all actors, observers and self-representations together with the relationships among these entities. A scenario by their definition, contains multiple scenes which are linked by actions and events.
Anomaly detection is relevant for a broad spectrum of applications like medicine, fraud detection, detecting errors in production or autonomous driving. This work focuses on anomaly detection in the case of autonomous driving, where it is highly relevant in order to deploy safe autonomous systems in an open world scenario. In autonomous driving, anomaly detection can be performed based on lidar, radar or camera data. Anomalies can be different things like error in the data, novel situations, malfunctioning sensors or technical failure and can have dangerous consequences. This work focuses on the camera based scenario where anomalies in the form of unknown objects should be detected in image data. What makes it even more challenging is the fact that the same item can be normal or anomalous depending on the context of the situation where it appears. A tree standing next to the road is a common scenario and therefore normal but a tree lying on the road is unusual and an anomaly which could create a possibly dangerous situation. Many of the models used for tasks in autonomous driving are developed under the assumption that the test data follows the same distribution as the training data [87]. In reality this cannot be guaranteed and cars will inevitably encounter novel situations where they lack training. Especially in safety critical applications like autonomous driving, these autonomous systems need to be aware of the situations for which they are not trained for in order to handle them with more care or flag them for human intervention. In reality, though, they often fail when encountering these situations in an open world environment and even worse, they often do that without any sings of failure and provide high confidence predictions while being incorrect. As shown by Nguyen et al. (2015) [59] and Kumano et al. (2022) [49], DNN models can classify unrecognizable images with near-certainty as member of a specific class, while humans can clearly distinguish between these. Also small pixel distortions which are barely visible to humans (adversarial images) can mislead most DNN [76, 32]. This

shows that there is still a large perception difference between computers and humans which leaves room for errors in computer vision tasks.
Making mistakes with high confidence is not only a problem with high complexity datasets containing traffic scenes, but even happens on simple datasets like MNIST[3]. As shown by [39], even random noise fed into a classifier trained to differentiate between different handwritten digits of the MNIST dataset gives a class probability of 91% which shows that even classifiers for simple datasets fail to show when they are unsure about a prediction. A naive approach might be to just train a model on anomalies to detect them. This fails due to the fact that the number of possible anomalies is not limited, so there can always be new scenarios which are unknown to the model. This makes it impossible to include all types of anomalies in a dataset. Furthermore, simply training a model to detect fooling images doesn't prevent it from misclassifying new ones in the future [59]. This shows that there is still a long way to go in order for autonomous systems to reliably detect anomalies, which is crucial for their deployment in a safety critical open world setting.
Some recent approaches like VOS [25] use virtual outliers, sampled from low likelihood regions of normal representations in the latent space to provide a decision boundary for detecting Out-of-Distribution (OOD) data. Similar to this, the approach described in chapter 4, follows the assumption that anomalies and normal data have a different representation in the latent space. Therefore, the goal of this thesis is to create latent representations of image data and to evaluate if it is possible to detect anomalous objects in real world driving scenes by clustering the latent space of a VAE. In order to do that, the latent space needs to be structured in a way that anomalies are separated from the normal data. To place more attention on small unknown objects on less on the environment, discrepancy maps which highlight small unknown objects were used as additional model input (see chapter 4.1). While there are many different definitions on what an anomaly is (more in chapter 3.1), the focus of this work is to detect images containing unknown objects which are not represented in the training data or appear in a different context. An example of such unseen objects could be an image of a tiger if the training data has no tigers in it. A tree lying on the road would also be considered anomalous, even if the normal data contains images of trees standing next to the road, because it would appear in a different context than usual. These anomalous objects can appear in different sizes and different locations in the image. Such situations, just like unknown objects, can be difficult to handle properly and therefore can pose a threat to autonomous systems.

# 2 Background

The approach described in this thesis focuses on detecting anomalies with the help of a VAE. To better understand VAEs and why to use them for anomaly detection, the following chapter gives a short introduction to variational autoencoders, what makes them special and what are the main advantages for anomaly detection when compared to normal autoencoders (see Fig. 2.1).

### 2.0.1 Autoencoder

An autoencoder (AE) is a neural network which is trained in an unsupervised manner to reproduce its input [30]. The model consists of two parts, the encoder and decoder. The encoder can be seen as a function $f(x) = z$ which compresses the input $x$ to a much smaller fixed size latent vector $z$. The decoder $g(z) = \hat{x}$ then tries to reproduce the input from the latent representation $z$. To achieve this goal, the encoder must learn a rich feature representation of the input and store as much relevant information as possible in the compressed representation so that the decoder reproduction is similar to the original input. Autoencoders are trained to minimize the objective

$$L(x, g(f(x))) \tag{2.1}$$

where $L$ is a loss function which penalizes $g(f(x))$ for being dissimilar from x, e.g., with the $L^2$ norm of $x$ and $\hat{x}$. Given their ability to compress and reconstruct data, they can be used as compression models, denoising models or feature extractors. As $f$ and $g$ are deterministic functions, the whole model is deterministic and maps $x$ to a specific $z$ value. Unlike VAEs, AE are not suited to generate new images. If the latent representation of an encoded image is passed through the decoder, then the original image is reconstructed. But if a point close to this latent representation is selected and decoded, the result is not necessarily similar to its latent neighbour and potentially just random noise. This is because the AE just learns to map a given input to the latent space and reconstruct it resulting in a unstructured latent space where a point close to another point doesn't need to have a similar input. Because of this, one can not select latent representations (e.g. via sampling from a distribution like in VAEs) from which new meaningful images can be generated and therefore autoencoders are not seen as generative models [30].

### 2.0.2 Variational Autoencoder

A variational autoencoder, compared to an autoencoder, is a generative probabilistic model [30]. It is based on the assumption that a data sample $x$ is generated by some random process which involves an unknown random variable $z$. $z$ is generated from some prior distribution $p_\theta(z)$ and the value of the data x is generated by the conditional distribution $p_\theta(x|z)$ while the true value of $z$ and

$\theta$ are unknown. The marginal distribution over the observed data $p_\theta(x)$ is given by:

$$p_\theta(x) = \int p_\theta(x,z)dz = \int p_\theta(z)p_\theta(x|z)dz \quad [2.2]$$

which is also called the marginal likelihood of the data. The problem is that the marginal likelihood of the the data is typically intractable due to the integral. This makes the true posterior density

$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} \quad [2.3]$$

also intractable. A VAE provides an efficient way to tackle the problem of estimating the likelihood and posterior in our data. In its original version, the VAE consists of two parts [46]. The first part is a recognition model $q_\phi(z|x)$, this can be seen as a probabilistic encoder that produces, for a given input $x$, a distribution over possible values of $z$ from which x was generated. It approximates the intractable true posterior

$$q_\phi(z|x) \approx p_\theta(z|x) \quad [2.4]$$

which can be used to optimize the marginal likelihood. The second part is the probabilistic decoder $p_\theta(x|z)$ which produces a distribution over possible values of $x$ for a given $z$. The model parameters $\phi, \theta$ are jointly learned. The VAE is trained to optimize the Evidence Lower Bound (ELBO), also called variational lower bound. The marginal likelihood is given by:

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x)] \quad [2.5]$$

$$= \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)]}_{\mathcal{L}_{\theta,\phi}(x)} + \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log q_\phi(z|x) - \log p_\theta(z|x)]}_{\geq 0} \quad [2.6]$$

where the right term is the Kullbach-Leibler (KL) divergence. The KL divergence measures the similarity between two distributions $p(x)$ and $q(x)$ and is defined as

$$\mathcal{D}_{KL}[p(x)||q(x)] = \sum_{x \in X} p(x) ln \frac{p(x)}{q(x)} \quad [2.7]$$

if $x$ is a discrete random variable, and as

$$\mathcal{D}_{KL}[p(x)||q(x)] = \int_{-\infty}^{\infty} p(x) ln \frac{p(x)}{q(x)} dx \quad [2.8]$$

if x is continuous. In the case of VAEs, it measures the difference between $q_\phi(z|x)$ and $p_\theta(z|x)$ which is the similarity between the approximation and the true posterior which is unknown and intractable. This term is equal to zero if $q_\phi(z|x)$ equals the true posterior distribution. Note that the KL divergence is not a true distance measure as it is asymmetrical with $D_{KL}(p||q) \neq D_{KL}(q||p)$. Because the KL divergence is always non-negative, the first term denotes a lower bound and is

called the ELBO which can be written as:

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z|x)) \quad [2.9]$$

The first part represents the expected reconstruction error and the second part acts as a regularizer which forces estimated posterior to be close to the prior which in most cases is a univariate Gaussian. This regularization results in a more structured latent space compared to AE. From that structured latent space we can sample and generate new data. The regularization and the resulting structure in the latent space ensures that semantically similar images also have similar latent representations. This is a huge advantage and one of the main reasons the VAE is better suited for anomaly detection in the latent space. Because $z$ is a random variable and we can not directly backpropagate gradients through it, the authors of [46] proposed the so called reparameterization trick which is shown in Fig.2.2. They express the random variable $z \sim q_\phi(z|x)$ as a deterministic variable $z = g_\phi(\varepsilon, x)$ using another independent random variable $\varepsilon$. For a univariate Gaussian case where $z \sim p(z|x) = N(\mu, \sigma^2)$ the reparameterisation would result in $z = \mu + \sigma\varepsilon$ with $\varepsilon \sim N(0,1)$ [46, 47]. In a variational autoencoder, the probabilistic encoder and decoder can be represented by a neural network where the encoder approximates the parameters of a distribution which usually is a univariate Gaussian with the parameters $\mu, \sigma$. The choice for a Gaussian distribution is made because it is assumed that the relationship between variables in the latent space is is way simpler than in the original input space [8]. Compared to AE, the VAE calculates a reconstruction probability rather than a reconstruction error and provides a theoretical foundation

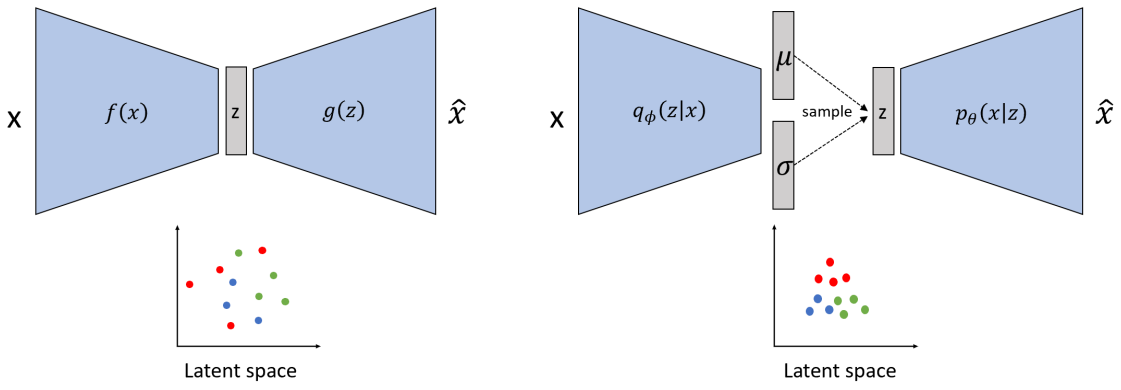



Figure 2.1: Basic architecture of AE/ VAE. On the left is a VAE where the encoder $p_\theta$ calculates the parameters of a distribution from which the latent variable $z$ gets sampled. The decoder $q_\phi$ then tries to reconstruct the input $x$. On the right is a AE where the encoder produces $z$ directly. As shown below, the latent space of a VAE is typically more meaningful and better structured when compared to an AE.

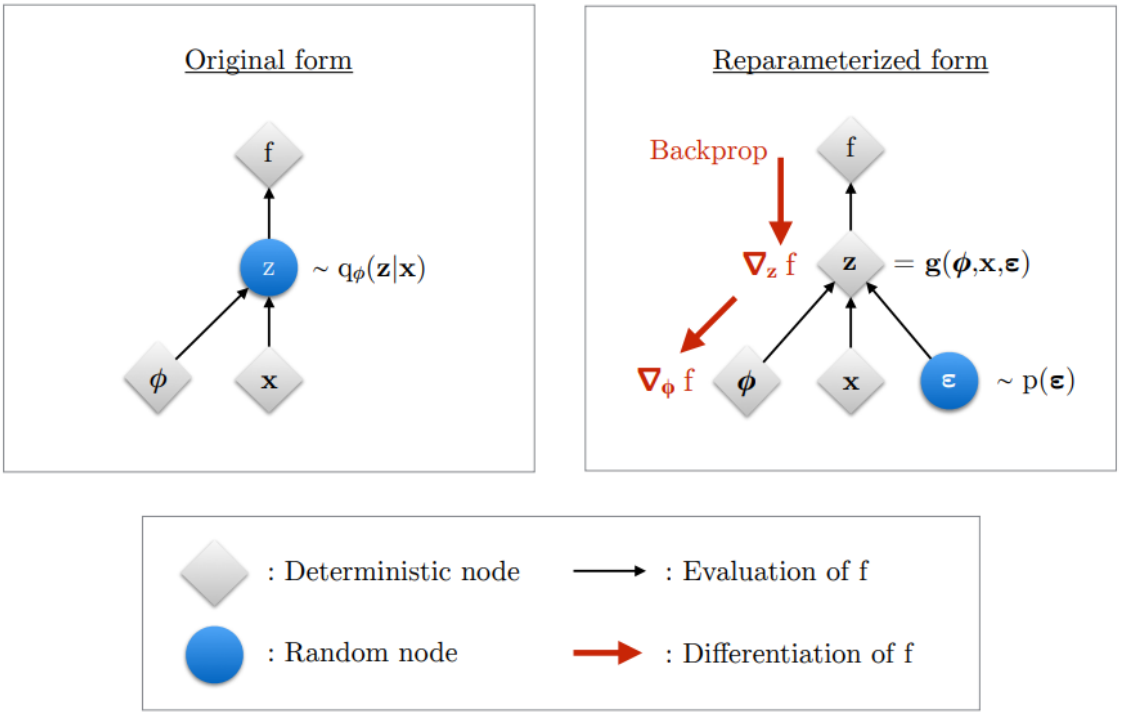



Figure 2.2: Illustration of the reparameterization trick. In the original form, calculating gradients for backpropagation is not possible because of the random variable $z$. In the reparameterized form, randomness is introduced with the help of an external node which makes backpropagation through $z$ possible. Reprinted from [47].

# 3 Related Work

In this chapter first provides an overview of different problem definitions and related terms in the filed of anomaly detection. Furthermore a summary of different approaches and methods used for anomaly detection ranging from non-deep learning to the current state of the art is provided. While anomaly detection is performed in many different fields like lidar [83] or radar [17] this section will mostly focus on camera based approaches and detecting anomalies in image or video data. For that, this chapter provides a overview of different methods using neural networks and at different approaches which use AE or VAE with reconstruction and latent space based methods for anomaly detection.

## 3.1 Taxonomies in the Field of Anomaly Detection

The terms outlier, anomaly and corner case are often used synonymously [42, 19] and there is no general definition for these terms. In literature [42, 69, 19, 87] one may find many different names for similar tasks like AD, Novelty Detection (ND), One-Class Classification (OCC), OOD detection, Corner Case Detection (CCD) or Open-Set Recognition (OSR). Chandola et al. [19] define anomalies as patterns that are different from the learned ones and Bolte et al. [12] describe a corner case as a "non-predictable relevant object/class in relevant location". Anomalies and corner cases by these definitions both describe an image which deviates from the norm but corner cases can also include other factors like the influence a situation has on the driving behavior. More complex scenarios which become anomalous only in their entirety but not consists of any anomalous objects can also be called corner case [37].

Breitenstein et al. [13] give a more structured overview by dividing corner cases into the categories:

**Pixel level:** errors in the data including local outliers(e.g. pixel errors, dirt on the camera) and global outliers (e.g. blinded camera).

**Domain level:** large constant shift in appearance but not semantics (e.g. different locations or weather conditions).

**Object level:** unknown objects (e.g. tiger on the street).

**Scene level:** unexpected patterns in a image. Includes collective anomalies with multiple known objects but in unknown quantity (e.g. crowd of people on the street) and contextual anomalies with a known object in a unknown location (e.g. tree lying on the street).

**Scenario level:** unexpected patterns observed over a image sequence and therefore requires a temporal understanding (e.g. person suddenly crossing the street).

Breitenstein et al. [13] followed the scene/scenario definition from [77]. Similar to their categorization of corner cases, Heidecker et al. [37] divide anomalies into different categories like sensor layer, content layer and temporal layer which each have different sub categories. Anomalies in the sensor layer can be divided into hardware level anomalies like pixel errors or broken lenses and physical level anomalies like dirt on the lens. Anomalies found in the image data itself can be very diverse because of the high information content that is included in one image and are represented by the content layer. The content layer includes the domain level (e.g. shifts between different countries), the object level which includes unseen objects and the scene level which contains new situations like a tree lying on the road. In the temporal layer, there are anomalies which are detected in video sequences and therefore need a temporal component. A pedestrian crossing a read light is an example for this category [37]. While there is no clear distinction between AD, OOD, ND, OSR and Outlier Detection (OD) and they are often used in different ways.

Yang et al. [87] provide a unified framework (see Fig. 3.1) for of generalized OOD detection which covers some of these terms and shows similarities and differences between them. By their framework, anomaly detection is the task of detecting anomalous samples which deviates from a pre-defined normality. They divide it into two subcategories depending on what causes the deviation from the normality. The two subcategories are semantic AD where a semantic shift occurs, like a object from a novel class and sensory AD which includes domain shift or style changes with no semantic shift. Their framework divides novelty detection in one and multiclass novelty detection. In one class novelty detection, normal In-Distribution (ID) images all belong to one class while test images which for example include the appearance of a new class are considered OOD. Multi-class ND differs from the one class case in the fact that ID belong to multiple classes. In this definition, ND is identical to semantic anomaly detection. Similar to this, Salehi at al. [69] looks at AD and ND as binary decision tasks where a given sample is classified either as normal or anomalous. OSR is often defined as a special form of multi-class ND where also the labels of the normal samples are taken into account. The difference between multi-class ND and OSR is that in OSR also includes ID classification [69, 87]. Following the framework of Yang et al. [87] OOD detection is similar to OSR but has a broader spectrum of learning tasks and solution space. The term outlier detection describes a setting where the majority distribution is considered as ID while outliers have distribution shifts from the majority. OD as defined in their framework doesn't follow a train-test scheme and all observations are provided [87].

The approach in this thesis follows the framework described by Yang et al. [87], and focus on semantic AD/ multi-class ND because the goal is to just detect anomalies in a setting where objects of multiple classes are present without a classification of the normal instances. In addition to this definition, already known objects were defined as anomalous if they appear in a completely different semantic context like a tree lying on the road or a car lying on its back. This matches the definition of corner cases on a object and scene level, provided by [13].

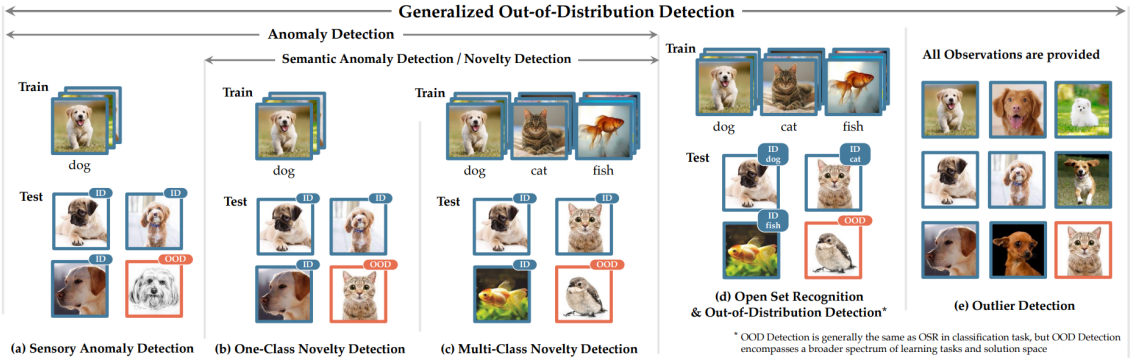



Figure 3.1: (a) shows sensory anomaly detection where images with covariate shift are considered OOD. In one-class ND (b), all normal images belong to a single class and a image with a semantic shift is considered anomalous. Multi-class ND (c) has ID images from multiple classes where images which belong to a different class are seen as OOD. ND (b)/(c) in this framework is identical to semantic anomaly detection. OSR (d) is similar to multi-class ND but with additional classification of ID images. OOD describes the same task as OSR but has a broader spectrum of learning tasks and solution space. In outlier detection (e), all observations are provided and the majority class is considered ID while outliers have some distribution shift from the ID data. Reprinted from [87]

## 3.2 Outlier Detection

The detection of outliers in data has been researched for many years and many different approaches have been developed. This chapter describes some non deep learning based approaches which focus on the detection of outliers. As described in the previous section, an outlier is an data point which deviates from the majority of the data. These approaches can be used in the input data space or in a lower dimensional space (e.g.created by Principal Component Analysis (PCA)/ latent space of VAE) to identify data points which differ from the majority and could potentially be anomalous. Popular approaches are for example statistical-based methods, distance-based methods or density-based methods [80]. Distance-based approaches (e.g. [22]) classify a data point as an outlier based on its distance (e.g. l2-norm) to other data points. Because the ability to discriminate between the nearest and farthest point decreases with increasing dimensionality [48], distance gets less meaningful as an outlier measurement for high dimensional data [9, 5]. Because of that, other approaches use angle-based methods to find anomalies. In angle-based approaches variance of vector angles from the point under investigation to all other datapoints are used as a indicator for an outlier. Kriegel et al. [48] propose, that in order to find patterns in high dimensional data you should look at angles between pairs of distance vectors in addition to distance. A low variance in the angles of vectors from a certain data point is a sign for an anomaly. Density-based approaches like Local Outlier Factor (LOF) [14] try to analyze how isolated a data point is with respect to its neighbors. Local in this context means that only a restricted neighborhood of the object is taken into account, or in other words its number of nearest neighbors. The authors of LOF propose that being an outlier is not a binary property but more a degree to which an object is isolated from its neighborhood. Other methods use decision trees for the task of outlier detection. A decision tree splits a given dataset into smaller subsets with the goal of maximizing homogeneity

within these subsets. Liu et al. [52] use the concept of decision trees for outlier detection. Their approach, called isolation forest, build a ensemble of decision trees where outliers are those with the shortest average paths. This means outliers are detected based on the number of splits required to isolate a certain data point. This method has the ability to handle large amount of data with high dimensionality and the advantage of linear time complexity and low memory requirement compared to other methods which are based on distance or density measurements.

## 3.3 Anomaly detection using Neural Networks

With big advances in deep learning in the last few years, deep neural networks have been widely used for a variety of different tasks. Because of their ability to extract rich and meaningful features from images, they became a popular method for anomaly detection tasks in many fields like object detection or classification. When thinking about anomaly detection with neural networks, a naive approach would be to train a neural network as a binary classifier to detect anomalies. Ruff et al. [68] follow an approach similar to this idea, they first transform the data using a neural net. The training objective here is to learn meaningful feature representation together with a one-class classification objective. The network is trained to map the input data to the output space with all data points lying inside of a hypersphere with minimal volume. They assume that the majority of the training data is from the one normal class, which is often the case for one-class classification tasks. The anomaly score for new input data is then computed by calculating the distance between the transformed representation and the center of the hypersphere, assuming anomalous data lies further away from the center. While such one-class classification methods may be effective when a single class is considered normal, they may fail in real world applications where the category of normal data often consists of heterogeneous semantic labels [64].

The authors in [40] propose a new outlier exposure objective. They create an outlier dataset from publicly available data and use it in addition to the normal data for model training. Applied to classifiers which are used to detect anomalies, outlier exposure results in a uniform softmax distribution for anomalies. Given an outlier dataset, the model learns to predict if a given sample is drawn from the inlier our outlier dataset. While some approaches like [39] use softmax confidence to detect unknown objects, this is not well suited for anomaly detection because it can produce high confidence scores for data far away from the training distribution. This is due to the fact that softmax is an rapidly growing function, which means that low confidence predictions can quickly become very high confidence predictions and the model is rarely unsure about a prediction. In [55] the authors propose to use an energy score which they claim to be superior to softmax based methods as well as generative based methods and can replace softmax confidence in any pre-trained network. The model maps each input to a scalar which indicates if its an OOD or ID object (low for known data, high for unknown data). They show that in contrast to the softmax score, the energy based one is theoretically aligned with the probability density of the input which means samples with higher energy have a lower probability of occurrence. A set of energy values can then be turned into a density function through the Gibbs distribution. Compared to outlier expo-

sure [40], which forces a uniform softmax distribution for outliers, the energy score [55] optimizes the energy gap between in and out of distribution directly, which results in better performance.
Liang et al. [50] use a different method to detect outliers which enhances performance and can be implemented without retraining the network. The input images are pre-processed by adding noise to them which increases the performance of the network because it has a stronger effect on the in distribution data then on the outliers which makes them more easily separable. Anomalies are detected by calculating the softmax score and compare it to a certain threshold which indicates if an image is a outlier or not. The softmax score is calculated as usual but with a temperature scaling parameter added to it. They claim a sufficiently large temperature parameter results in better detection. Papadopoulos et al. [62] adopt the idea of outlier exposure and achieve superior results. They use two regularisation terms, the first one minimizes total variation distance, in contrast to other methods using KL divergence [40], between the output distribution (given by softmax) and a uniform distribution. The second term minimizes the euclidean distance between the training accuracy of a neural network and its average confidence of its predictions. This leads to the model making average confidence predictions for known samples and highly uncertain ones for OOD samples. The drawback of some of the described approaches using outlier exposure is the need for a auxiliary dataset which needs to be diverse to cover as many cases as possible while making sure to not overlap with the ID-dataset. To tackle this issue, [25] synthesize virtual outliers in order to perform OOD detection for classification and object detection. To avoid the requirement of an outlier dataset, they assume the feature representations of the input images form a class conditional multivariate Gaussian distribution and sample outliers from low likelihood regions of this distribution. Generating virtual outliers from feature space is way more practicable than from the pixel space because of its lower dimensionality. A classifier is then trained to produce a higher uncertainty for the virtual outliers than for normal samples. The same approach can be used for object detection by replacing image uncertainty with object uncertainty.
A common problem for the approaches described here is that they produce high false positive rates. Even though VOS [25] reduced was able improve on this, it still remains an issue. Similar to VOS, Nitsch et al. [60] also make use of generated OOD training data but use a Generative Adversarial Network (GAN)[31] based method to generate it. Their approach therefore needs no anomalies for training and has low computational cost which makes it practicable for online scenarios. They use an auxiliary GAN to produce out-of-distribution training samples in order to train a classifier to assign low confidence predictions on these samples. In their setup the generator was trained using an additional loss component which forces him to produce outputs on the decision boundary of the classifier. The generator therefore tries not only to fool the discriminator but also forces the classifier to assign low confidence predictions to samples on, or outside the decision boundary. In addition to that they propose a post hoc component in which they compute the parameters of a class conditioned Gaussian distribution over the loggits of the classifier. During deployment, the distance (cosine similarity) to the Gaussian which is most likely, according to the softmax output, is used as an anomaly measurement. Generating synthetic anomalies can also be achieved using normalizing flows. Grcic et al. [33] claim that, compared to GANs, normalizing flows are better suited for anomaly detection because of better distribution coverage and more sable training. They

use a normalizing flow to generate negative patches of random size which then get pasted atop an in distribution training image. The training objective is for the discriminator to generate a uniform distribution on the generated samples and for the normalizing flow to maximize the likelihood of inlier patches. While the first objective moves the the generative distribution away from the inliers, the second one moves it towards them. This results in negative samples at the boundary of the training distribution. Synthetic anomalies can contain parts which look similar to scenes in normal images and therefore produce confident predictions by the discriminator. Because these predictions get penalized by the kl-divergence, the discriminator learns to lower its confidence on inlier pixels. To prevent this, the authors propose to use the Jensen-Shannon divergence which only mildly penalizes high confidence predictions. This approach outperforms several other techniques on different benchmarks and has state of the art performance in image anomaly detection [11].

Due to the advances in semantic segmentation networks and their usage in safety critical applications like autonomous driving [27], it becomes more important for these models to detect anomalies in order to prevent the model from making wrong predictions with high confidence. Classical semantic segmentation networks consist of a feature extractor and a classifier to assign a class to every pixel. Closed set semantic segmentation models are based on the assumption that all classes are included in the training data which in reality is not the case [15]. Because the classifier has no class for unknown objects [15] developed a different approach using a segmentation system which detects objects of all kinds and gradually incorporates unknown objects to its knowledge base. They state that using a feature extractor and a classifier is problematic because the classifier can only assign known labels. They instead train the feature extractor to map the input to a feature vector which has the same euclidean length as their prototype representation. This prototype is from a given set where each class has a prototype representation. The euclidean distance between the feature representation of the input and the prototype representations is used as a classification method. Their open world semantic segmentation module consists of a closed-set segmentation module which assigns in-distribution labels to all pixels, and an anomaly segmentation module responsible for detecting OOD pixels. The anomaly segmentation module uses two criteria to identify anomalous pixels, the euclidean distance between the feature vector of the input and the prototype representations and a metric-based maximum softmax probability. During inference, samples are pulled towards prototypes of their own class (if existing) and get repelled by prototypes of different classes which leads to clusters of similar data around their prototype representation. Objects which are detected as anomalies are then learned by a incremental few shot learning module. Di Biase et al. [23] combine model uncertainty with image re-synthesis in order to prevent semantic segmentation models from not detecting or wrongly classifying an unknown object. The first component of their model, the segmentation network, creates a semantic map for a given input image and calculates softmax entropy and softmax distance for each pixel. The Synthesis module, which is trained as a conditional GAN (cGAN) then tries to reconstruct the input image from the semantic map. Because the semantic map doesn't provide any information about color or appearance of the individual objects, the perceptual difference, which focuses more on the semantic similarity, of the original image and the re-synthesized one is used to detect differences

instead of a pixel by pixel comparison. The last part, the dissimilarity module then takes the original image, the semantic map, the re-synthesized one and the uncertainty maps (softmax entropy/ distance, perceptual difference) as input to detect anomalies. The dissimilarity module consists of an encoder to extract features from the inputs, a fusion module to guide the network to focus on high probability areas and decoder to produce the anomaly map in which unknown objects are highlighted. This approach doesn't constrain the segmentation network and can therefore be used with state-of-the-art semantic segmentation models.
Golan at al. [29] propose to detect anomalies using different geometric transformations. They train a multi-class neural classifier over a self labeled dataset. This dataset is created by performing different types of geometric transformations on normal data. The goal here is that for each image, the model can predict which transformation was applied to it. For test images all the transformations are applied and the classifier outputs a softmax response for each of the resulting images. Comparing the combined log-likelihood of these vectors to the distribution of softmax vectors from normal data results in the final anomaly score. Differentiating between different geometric transformations should encourage the model to detect new geometrical features in the input image which often represent unknown objects.

This section showed that there is wide variety of different approaches which try to detect anomalies using deep learning methods. While there have been significant improvements over the last years, the existing methods are far from perfect. The next chapters will focus on methods using AE/ VAE to perform anomaly detection

## 3.4 Anomaly detection using AE/ VAE

A popular tool for anomaly detection are autoencoder or variational autoencoder. While there are many different approaches which use these kinds of models for anomaly detection, most methods can be categorized in reconstruction-based or latent space-based methods. Reconstruction-based methods rely on the assumption that if a model is trained on normal data only, it only learns to reconstruct normal data and fails to do so for data not represented in the training data. These methods therefore use the difference between the original image $x$ and its reconstruction $\hat{x}$ as an indicator for anomalies. Latent space based methods try to detect outliers in the latent space of the model. These methods assume that the latent representation $z$ of an image contains all necessary information displayed by the image and therefore anomalies should have a different latent representation than normal instances. Although the boundary between these two methods is not always clear, with some approaches using latent representations and reconstruction ability, the following chapters are intended to provide an overview of different approaches using these two methods.

### 3.4.1 Reconstruction-based

In order to detect anomalies using reconstruction error, Vu et al. [79] developed a GAN-based method in which the generator and discriminator both are made up of AE. They define the training

objective as the pixel-wise error between the reconstruction of the data produced by the generator and the generated image produced by the discriminator. This forces the discriminator to produce bad reconstructions with a high pixel-wise error if it thinks the data doesn't belong to the original data distribution. To furthermore improve training stability they model the learning rates of generator and discriminator after a sigmoid centered around zero which allows the weaker model to catch up to the better performing one. The goal of the network is for the discriminator to accurately reconstruct data, sampled from the original distribution, and fail to do so for out of distribution data which results in a higher reconstruction error. An et al. [8] claim that for VAEs anomaly detection using reconstruction probability $\mathbb{E}_{q_\phi(z|x)}[logp_\theta(x|z)]$, which takes the variability of the distribution of variables into account, is more suitable compared to methods using reconstruction error (e.g. $||x-\hat{x}||$). Somepalli et al. [74] also claim, metrics like the l2-norm are not suitable for anomaly detection with AE via reconstruction error because it only compares pixel level errors but doesn't capture high level structures in the image. Some works [58] try to solve this issue by introducing adversarial loss which is capable of capturing high level details. This fixes the problem of blurry reconstructions in low diverse settings like faces but remains poor for more diverse datasets like CIFAR-10 [1]. The authors of [74] posit that this issue arises because some loss functions with adversarial loss compare distributions for batches of samples but not individual samples themselves. They instead propose to minimize the wasserstein distance between the distributions $\mathbb{P}_{x,x}$ and $\mathbb{P}_{x,\hat{x}}$. Furthermore they propose latent space regularization by performing Simplex interpolation of normal samples in the latent space and by sampling synthetic negatives and optimizing the latent space to be far away from them. Negative examples are sampled from an atypical set of the latent space distribution. Based on the assumption that in a d-dimensional space a typical set, which is a set where the information content of the elements is similar to the expected information, lies with high probability at a distance of $\sqrt{d}$ from the origin [74]. They propose to sample negative outliers from the region between $\sqrt{d}$ and $\sqrt{d} \pm \delta$. In their experiments they showed that regularization on the convex combination of latent codes of training samples works better if some negative examples are also provided. With these negative examples the model can be trained to reconstruct normal data while making sure that anomalies are reconstructed poorly.

Anomaly detection using reconstruction error can also be applied to online scenarios. Bolte et al. [12] perform online corner case detection utilizing semantic segmentation and video-based prediction error. Their detection method is based on predictability and the assumption that a situation, even if it is novel or abnormal, which is technically predictable poses no threat to an autonomous driving system. Their method consists of 3 parts. The first part is a prediction module which consists of an AE that outputs the prediction error (based on Mean Squared Error (MSE)) for each new frame based on the previous frames. In the second part, a semantic segmentation module detects and classifies objects. Lastly a detection module processes the information output by the other two parts. Because only moving objects are considered relevant, the prediction error for non moving classes is set to zero and errors are weighted based on their distance to the bottom of the image (objects at the bottom are assumed to be closer and therefore more relevant) and summed up. The final error score is normalized and based on a threshold $t$, classification into corner cases

or normal data can be performed.

Some approaches like [7] focus on an end to end autonomous driving scenario where in addition to the anomaly detection task, the model is trained to output a steering command. The steering command is generated from the latent space of a VAE. In contrast to a standard VAE they explicitly supervise a single dimension of the latent space, which represents the steering output and gets trained using ground truth human data. For the detection of novel images, a pixel wise uncertainty estimation between the input x and its reconstruction $\hat{x}$ is performed. Based on a threshold for this loss, they classify an image as normal or anomalous. Furthermore they perform model debasing during training in which they sub-sample the dataset to reduce over represented samples which show faster and more data efficient training.

Other approaches [4, 81] combine reconstruction based detection methods with latent space analysis. Based on how "surprised" a model is for a given input, Abati et al. [4] try to detect anomalies using an autoregressive model in addition to encoder and decoder. Autoregressive models are generative models which perform sequential predictions where every prediction is based on the previous observations [30]. The latent space is generated by the AE which was trained to reconstruct anomalous free data. Besides the reconstruction error they try to minimize how surprising the produced latent representation is (low for normal data). This is done by the autoregressive model which learns the distribution of the latent vectors by maximum likelihood principals. For an anomaly, besides a higher reconstruction error, the model should produce uncommon latent features which are detected by the autoregressive model due to their lower log likelihood. These two measurements are combined into an anomaly score. Similar to this [81] uses a Vector Quantized - Variational Autoencoder (VQ-VAE) which is a special network architecture designed for image compression with a discrete latent space. The model gets trained on anomaly free data. With the use of an autoregressive model (PixelSNAIL [20]) the distribution of all the latent representations is learned in a second step. If an anomaly is passed through the network during the prediction stage, the latent code is out of the distribution learned by the network PixelSNAIL. PixelSnail then performs re-sampling on the latent code and passes it through the decoder to generate an image from it. The image which got reconstructed using the latent code generated by the encoder is then compared to the image which got generated from the re-sampled latent variables generated by PixelSnail. The difference in these two images is used as a anomaly score.

Another approach which combines feature representation and reconstruction is proposed by [63] who developed a memory module where items in the memory represent prototypical patterns of normal data. They assume that there is no single prototypical feature which represents all normal instances and try to capture the diversity of normal patterns through the memory module which gets updated during training on normal video frames. To reduce intra-class variance, they train the model to map a given input as close as possible to its nearest item in the memory. To prevent a degenerate solution, where all items are mapped closely to one memory item, they include the objective of minimizing the distance between a feature and its nearest item while maximizing the

distance to its second nearest. This ensures diversity in the memory. The model mainly consists of an encoder, decoder and memory module with a overall loss consisting of reconstruction, feature compactness and feature separateness loss. The abnormality score for an input of video frames is given by the $l^2$ distance between the input and the nearest item in the memory and the reconstruction of the video frame using the memory items. Motivated by the idea of using vehicular trajectories rather than video frames, Santhosh et al. [71] extract a color gradient representation of these trajectories from video sequences recorded by a stationary camera in order to detect anomalies in traffic scenes. They obtain trajectories by a tracking algorithm, cluster them to identify different patterns and map them into a color gradient form. Their approach consists of a VAE to identify anomalies and a Convolutional Neural Network (CNN) to classify trajectories. The CNN predicts the class of a given trajectory and the VAE detects anomalies based on the resulting reconstruction error. The classification of trajectories is necessary because in a given traffic scene a set of flows can be allowed to happen at the same time but other movements, which by it self are normal and can occur at a different time, are anomalous in this specific scenario. The VAE which is trained on normal data then fails to detect this movement because it is unknown.

### 3.4.2 Latent space-based

Detecting unknown or untested scenarios is crucial for scenario based testing. Because infrastructure is a big component of a scenario, [84] tries to find traffic scenes which are different than the ones already known to an autonomous driving model. In their approach they use a triplet based AE which maps road infrastructure to the latent space where novelty detection is performed. Triplet learning enforces similarity in the latent space based on data triplets. Because of that, neighboring points in the latent space have visually similar inputs. In order to measure the similarity of road infrastructure, a connectivity graph is generated for each scenario which is then used to compare different scenes. Similar to this approach, Harmening et al. [34] uses the latent space representation of traffic scenes in order to cluster the latent space and find different scenarios. They developed two approaches to generate the latent space. The first one consists of an AE which gets a 4-dimensional grid as input. This grid represents the 2 image axis with the color channel and a time axis. In the second method, each frame of the scenario is described by a set which contains features (e.g. position, velocity) of all traffic participants. Each element of a set gets encoded to an embedding vector which gets fed into an RNN-based AE where the resulting latent space is clustered. Because many real world datsets have multiple labels assigned to each image, Sundar et al. [75] propose to synthesize a multi-label dataset into smaller partitions who each have labels of one generative factor fixed while labels of the other factors can change. In their approach they train a $\beta$-VAE on each partition to generate the latent space. The VAE is trained as one class classifier to learn information about the fixed factor. During testing, each $\beta$-VAE should identify changes in the generative factor it was trained on. They deploy a chain of detectors with each detector using only a single latent variable which is the one that shows highest sensitivity to variations in the assigned factor. This VAE chain can then be used for online detection.

While many approaches focus on comparing original and reconstructed images to identify anomalies, Akcay et al. [6] propose a different method which compares the latent representation of the original image with the one from a reconstructed image. For that, they train an AE in an adversarial setting. The network contains two encoders, a decoder and a discriminator. First the encoder decoder pair are trained in a classical autoencoder setup and learn to reconstruct the input image. In this setup the AE takes the role of the generator. The reconstructed image is then passed to another encoder to generate its feature representation and a discriminator who decides between real or generated. The intuition behind this approach is, that if the model is trained only on normal samples, the AE has problems to reproduce anomalies. The anomalous input $x$ gets compressed to its latent representation $z$ and reconstructed to $\hat{x}$ which misses the anomalous parts of the image. It is then compressed by the second encoder to $\hat{z}$. The dissimilarity between $z$ and $\hat{z}$ is used to classify images as normal or anomalous. Chalapathy at al. [16] follow a similar approach to the one class classification network proposed by [68] with the additional use of encoder as feature extractor. They integrate a one-class-Support Vector Machine (SVM) equivalent object into a neural network architecture. This combines the ability of Neural Networks to learn meaningful representations of the data with the one class classification objective which separates in distribution data from out of distribution data. For that, they use a SVM like loss function for training the neural network. In the first step they train a deep AE to learn the features of the input image. After that, the encoder of this pre-trained network is used to generate the latent space of the input image. This representation of the original data is then passed through a simple feed forward network with one hidden layer and a scalar output which classifies the input data as normal or anomaly. The encoder and classifier weights are learned simultaneously in this second step. This differs from other methods which use an anomaly detector on generic features produced by an AE without training the model to produce features which are especially well suited for anomaly detection tasks [26]. These approaches train the feature extractor and anomaly detector independently from another which results in a less accurate performance in anomaly detection tasks.

Park et al. [65] look at VAEs from the perspective of rate-distortion theory and show that VAEs can be explained with the trade-off between the rate and the distortion. They propose that using only the encoder to calculate the marginal log-likelihood of the data is more effective and simpler than to use the reconstruction loss of a VAE to calculate the anomaly score. They follow the standard VAE architecture with a Gaussian $N(0, 1)$ as prior and let the encoder output the mean and standard deviation of a multivariate Gaussian but without the use of a decoder. The network basically learns to approximate the prior without the additional reconstruction task. Their prior generating network (PGN) generalizes methods like [68] described earlier.
Liu et al. [53] propose a method which creates an attention map using the latent representation of the data directly and therefore not needing an extra classification module. For every element $z_i$ in the latent vector $z$ an attention map $M^i$ is generated by performing backpropagation to the last convolutional feature map. In detail $M^i$ is computed by performing ReLU on $\sum_{k=1}^{n} \alpha_k A_k$ where $A_k$ is the k'th feature channel and $\alpha_k$ is a scalar value given by performing global average pooling on $(\frac{\partial z_i}{\partial A_k})$. For an d-dimensional latent space, this results in d attention maps with the mean of

them resulting in a single overall attention map. For a anomalous input they sample z from the normal difference distribution given by the overall $\mu$ and $\sigma$ which represent all normal images and the $\mu$, $\sigma$ inferred from the anomalous sample and calculate the anomaly attention map for this z. The resulting attention map highlights anomalous parts in the input image. The results can be improved by adding attention disentanglement loss which forces the high response regions of two attention maps generated from $z$ to be as separable as possible. Another method similar to the one used in this thesis was proposed by [24]. They combine Gaussian Mixture Model (GMM) with VAEs to form the latent space. The Gaussian Mixture Variational Autoencoder (GMVAE) uses a mixture of Gaussians as a prior. Their method is based on the assumption that the data is generated by multiple distributions. The Gaussian mixture prior results in a latent space in which multiple classes are more clearly separated from each other and therefore improves classification. The main difference between this and the approach used in this thesis is that in the GMVAE there is no conditioning on a predefined data label (normal/anomaly).

# 4 Approach

This section explains the approach that was used to in this thesis to detect anomalies in real world driving scenarios. For that, a VAE was trained on normal data and anomalies. The anomaly detection is then performed through clustering in the latent space of the VAE. The clustering assigns a binary label to every data point in the latent space and classifies it either as anomalous or normal. This approach looks at real world cases where the input data consist of many different objects from different classes

## 4.1 Discrepancy Images

This section will explain the generation of the discrepancy maps that were used as an additional input together with the RGB-images. This method was proposed by Lis et al. [51] and generates images where unknown objects are supposed to be highlighted by the model. The resulting image is added to the original one as a $4^{th}$ color channel and passed through the VAE. This should train the model to better detect anomalies by providing some additional information about the relevant parts of the image. The idea behind this is, to encourage the model to pay more attention to the highlighted regions in the image and especially the objects behind these regions. The goal is that the VAE learns that the objects behind the marked regions are of interest and should be labeled as anomalous if not known. By training on normal and anomalous images, the model should learn to differentiate between normal and anomalous images even if some normal parts are highlighted. It therefore should learn that not the marking is the relevant factor for a image to be classified as anomaly but the novelty of the object behind it. These discrepancy maps are created by the process used in [51]. The pipeline for creating these discrepancy maps consists of three parts. First a semantic segmentation network, second an image resynthesis module and a module for anomaly detection. As shown in figure 4.1, the pipeline first creates a semantic mask of the original input image. Then the resynthesis module tries to recreate the original image from the semantic mask. Because the semantic segmentation network is only trained on normal data it cannot classify anomalies correctly which leads to a wrong reconstruction by the resynthesis module. The anomaly detection part of the model then tries to pick up differences in the original and the resynthesized version of the image. The separate parts of the model will now be explained in more detail.

### 4.1.1 Semantic segmentation module

The semantic segmentation network used to create the semantic masks is the pyramid scene parsing network (PSPNet) [89]. The model architecture is shown in figure 4.2. For a given input, this model uses a pre-trained ResNet [35] model as a feature extractor. The resulting feature map

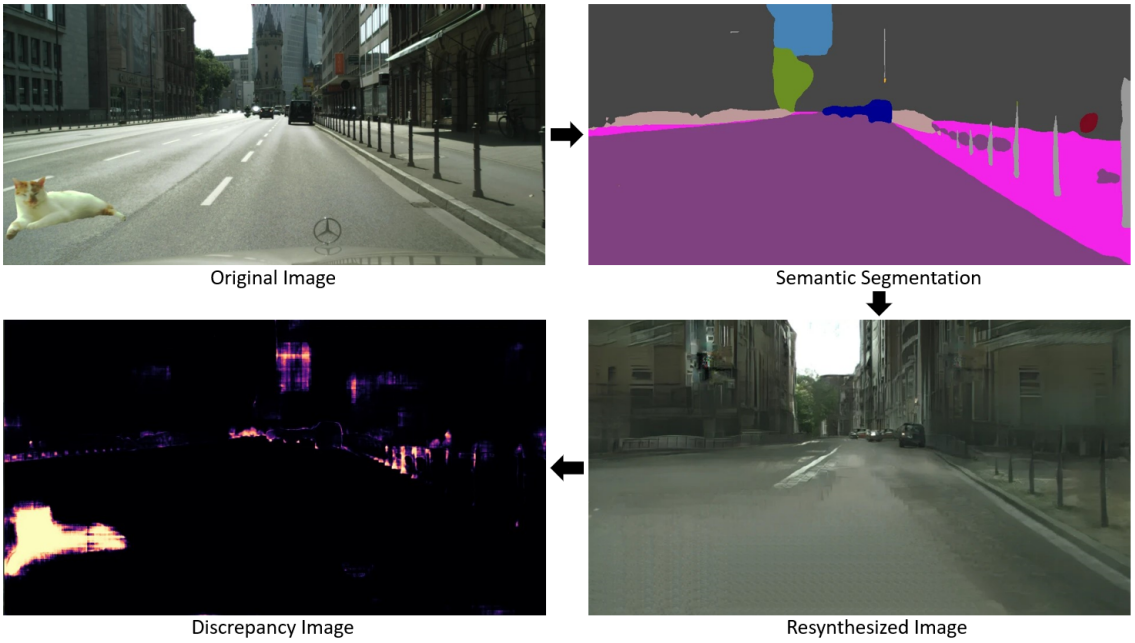



Figure 4.1: Overview of the pipeline which creates the discrepancy maps. The original image gets passed through a semantic segmentation module to create a semantic mask. From this a GAN tries to reconstruct the original image. The discrepancy map gets generated by comparing the original image to the reconstructed one.

has 1/8 the size of the input image and gets passed to a pyramid pooling module. This pooling module is used to gather information on different subregion representations and get global context information. It consists of four different levels, each pooling at different sizes in order to capture different portions of the image. The coarsest level is global pooling and generates a single output for each feature map. The lower level pooling modules focus on different subregions of the feature map and generate pooled outputs for different locations. For every level of the pyramid, a 1 x 1 convolution is performed to reduce the depth and get a 2 dimensional output. The low dimensional feature maps are then upsampled, in order to make them the same size as the original feature map. All these upsampled feature maps are then concatenated with the original feature map output by the ResNet feature extractor. This combination, which fuses features collected at different scales to better capture the overall context of the image, is then passed to another convolutional layer to generate the final class prediction for each pixel.

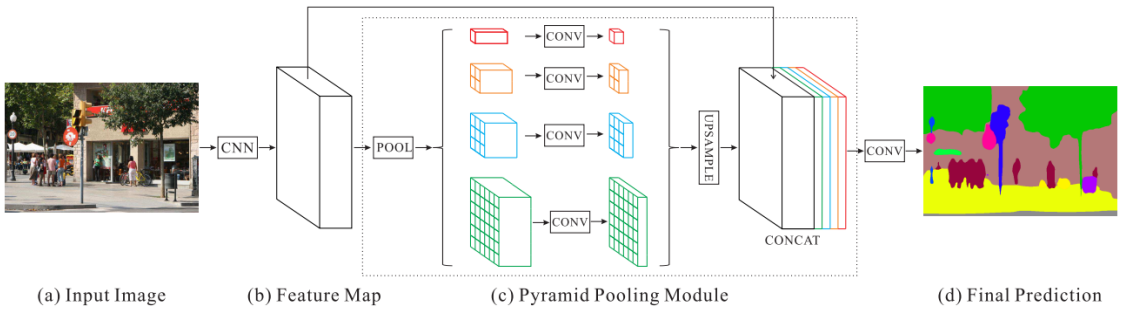



Figure 4.2: Overview of the PSPNet architecture. Reprinted from [89]

### 4.1.2 Resynthesis module

For image resynthesis the pix2pixHD [82] model was used. This model is a conditional GAN which can create a RGB image out of a semantic map. The objective of the generator $G$ in this case is to produce a real looking image from a semantic map while the discriminator $D$ tries to distinguish real image from the synthesized ones. The generator in this network consists of two sub-networks, a global generator $G_1$ and a local enhancer $G_2$ (see Fig. 4.3). The global generator consists of a convolutional front-end followed by a set of residual blocks and a transposed convolutional back-end. It gets a semantic map as input and produces an output image of the same size. The local enhancer network has the same architecture as the global generator but in- and output is twice the size in each image dimension. The input to the residual blocks of the local enhancer $G_2$ is not the output of the convolutional front-end like in the global generator $G_1$ but a element wise sum of its convolutional front-end output and the last feature map of the back-end of $G_1$ which integrates global information from $G_1$ to $G_2$. During training first the global generator gets trained and after that the local enhancer is appended to it and trained simultaneously. Because high resolution image synthesis is challenging for the discriminator and would require a larger network or larger convolutional kernels, which brings problems of memory usage and overfitting, the authors of [82] propose a multi-scale discriminator. They use three discriminators $(D_1, D_2, D_3)$ with identical network structures and use them on different image scales. The real and synthesized image gets down scaled by a factor of 2 and 4 to get images of 3 different sizes for the discriminators. They then try to distinguish between the real and the synthesized version while the coarsest one focuses more on global features and the finest more on the details of the image. The multiple discriminators transform the typical GAN objective

$$\min_G \max_D \mathcal{L}_{GAN}(G,D) \tag{4.1}$$

with $\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{(s,x)}[\log D(s,x)] + \mathbb{E}_s[\log(1 - D(s,G(s))]$ to a multi-task learning problem

$$\min_G \max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G,D_k). \tag{4.2}$$

In addition to this, the authors propose a feature matching loss which is added to the GAN loss in order to stabilize training and improve the generation quality. For this loss, features from different layers of the discriminator were extracted and features from real and fake images compared. This results in the final objective

$$\min_G((\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G,D_k)) + \lambda \sum_{k=1,2,3} \mathcal{L}_{FM}(G,D_k)) \tag{4.3}$$

where $\mathcal{L}_{FM}$ denotes the feature matching loss and $\lambda$ balances both terms.

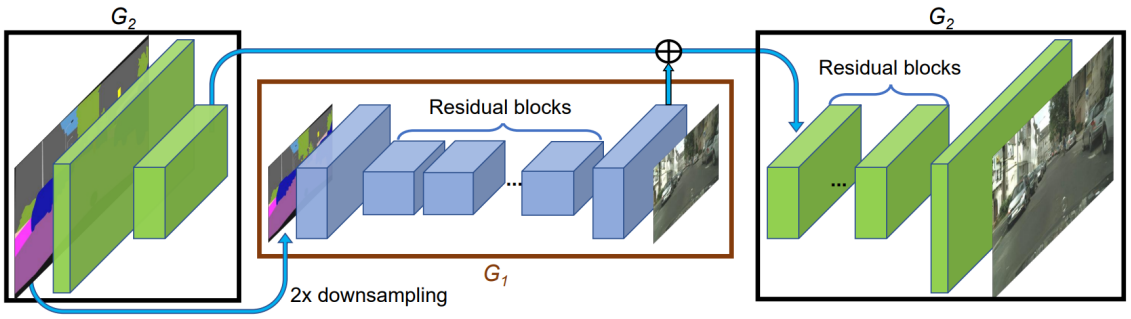



Figure 4.3: Pix2PixHD architecture. The residual network G1 gets trained on images with lower resolution. Later, another residual network gets appended and the networks are trained together while the input of the second part of G2 is the sum of the last feature map from G1 and the feature map from G2. Reprinted from [82]

### 4.1.3 Discrepancy module

The discrepancy module introduced by Lis et al. [51] detects anomalies by comparing the synthesized image to the original one. If unknown objects are not recognized or wrongly classified by the semantic segmentation network, the resynthesized version should be semantically different in these parts of the image. A simple pixel by pixel comparison would not give any information about anomalies and can therefore not be used as an anomaly detection measurement. The reason for this is that the segmentation module only captures the semantics of individual objects, it doesn't give any information on features like color or texture, so the resynthesized version will not match the original one in these aspects and therefore can have large pixel value differences even if the object is detected and classified correctly. The discrepancy network is build upon a three stream architecture (see Fig. 4.4). A pre-trained VGG network is used to extract features from the original image and the resynthesized version of it. A custom CNN is used to process the predicted semantic labels produced by the segmentation module. At each level of the feature pyramid the features from the 3 images are concatenated and combined by a 1 x 1 convolution. In addition, point wise correlations between the features obtained from the original and the resynthesized image are calculated. These pointwise correlations and concatenated features are then passed to to a upconvolutional pyramid which returns the final discrepancy map. In order to train this network the the authors used a synthetic anomaly dataset. This makes it possible to train the network without the need of real anomalies. The synthetic training set mimics the occurrence of unknown objects which get classified incorrectly by the segmentation network. For that, they propose to replace the label of a randomly-chosen object instance with a different random label. This process is shown in figure 4.5. By passing these modified semantic maps to the image resynthesis module reconstructs these objects according to the new label. This results in pairs of real images and fake images to train the discrepancy network. By using this method to create a synthetic anomaly dataset, no anomalies are necessary for training. The problem with this method of creating synthetic anomalies is that it chooses a random class to replace to selected object, without considering how often this class is seen normally in the dataset. Because not all classes have the same likelihood of appearing in an image, this leads to some rarer classes appearing more often as a replacement (which should

be detected as anomalous) than as normal instances. This results in the model associating these rare classes with anomalies and learning to classify them as such because it encounters them more often as a replacement class, which symbolizes an anomaly, than as a normal instance. To avoid this problem, the process of creating the synthetic anomaly dataset was modified. For a selected object, the replacement class is not chosen randomly but according to its probability of appearing in the dataset. This means that rare classes are chosen less often as a replacement then common ones, which prevents the model from learning to classify less frequent classes as anomalies and helps it to differentiate between rare and unknown objects.

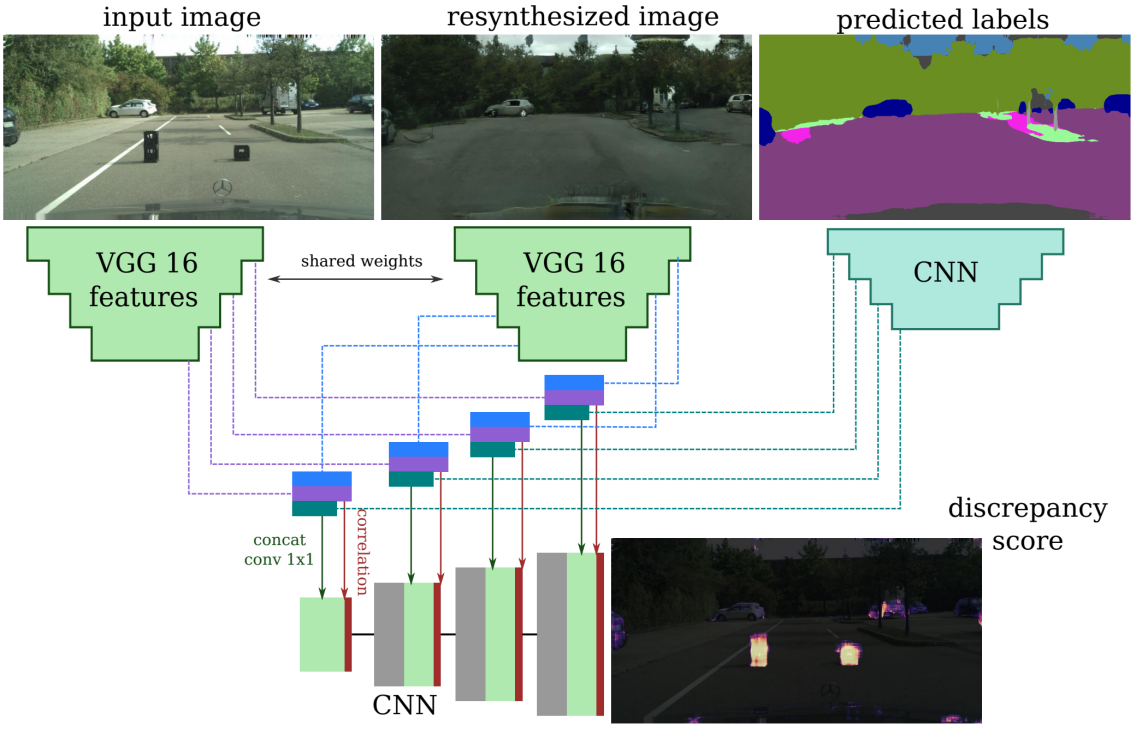



Figure 4.4: Architecture of the discrepancy module [51]. The VGG and CNN extract features from the three input images. Features and correlations are passed to a decoder which generates the final discrepancy map. Reprinted from [51]

### 4.1.4 Training

For the semantic segmentation module and the image resynthesis part of the model we used the weights provided in the implementation of [51]. They use the Cityscapes dataset as the "normal" dataset without anomalies. For the semantic segmentation module they used the BDD100k [88] dataset for training because of its high diversity and large number of training samples. The BDD100k dataset has the the same classes as the Cityscapes [21] dataset which represents the "normal" data for the discrepancy detector. The segmentation network therefore can only assign objects to the known classes and wrongly classifies objects from unknown classes. The image resynthesis module was trained on Cityscapes in order to learn how to reconstruct normal data. For the discrepancy module, the model had to be retrained on the new synthetic anomaly dataset which was generated from Cityscapes, while the rest of the training was kept like in the original

Figure 4.5: Creation of the training data for the discrepancy network as proposed by [51]. The left image shows the ground truth labels of a Cityscapes image. For some objects, the labels are replaced by labels from a different class which simulates that an object gets classified incorrectly. This is shown by the image in the middle. The image on the right shows the objects with wrong labels which should be detected by the discrepancy network.

implementation from [51]. For the both, the resynthesis and the discrepancy module the training images were down scaled to 1024 x 512 while the original version of Cityscapes has 2048 x 1024 images.

## 4.2 Latent Space Conditioning

A lot of anomaly detection methods focus on the reconstruction error/ reconstruction probability using AE/ VAE. This has some limitations because powerful models which can generalize well are able to reconstruct anomalous data just as good as normal one with no significant difference in the reconstruction error [54]. This is especially true if a majority of the anomalous image is normal [63]. This is often the case in traffic scenes where one unknown object alongside many known ones already can cause a scene to be anomalous. This method doesn't use reconstruction quality as an indicator of anomalies but focuses on anomaly detection in the latent space of a VAE. Because of the regularization term (most of the time kl-divergence), the latent space of VAEs can be more structured compared to the basic AE. This makes VAEs better suited for this approach and is the reason they were chosen. As already mentioned, VAEs are generative models using latent variables from which they reconstruct data or generate new data. In order to reconstruct a image from its latent variables, they need to store the relevant information contained in the image. The latent space of a VAE typically has way less dimensions than the input data. This means that not all information in the input data is relevant for capturing the overall semantic of the image. The idea is that in this compressed space some latent representations of anomalies differ from the ones produced by normal data. In order to create and structured latent space which makes it easier to detect outliers in it, anomalies and normal data need to be separated as much as possible. To achieve this, the model used in this thesis, uses a method proposed by Norlander et al. [61] which conditions the latent space on class labels, in this case anomaly and normal. Compared to a standard VAE which typically forces the latent space to be similar to a Gaussian prior the Conditioned Latent Space Variational Autoencoder (CL-VAE) [61] uses one Gaussian per class label. By using the additional information which is given by the labels, it is possible to more accurately express the data with several Gaussians. The CL-VAE assigns a distribution to every data point in the latent space. This makes it different from other approaches which use GMM for training a VAE. Similar methods using GMM like [24] don't condition the latent space on

a specific pre-defined label but use Monte Carlo methods to estimate the prior distribution. The CL-VAE assumes that not all data can be mapped to the same Gaussian and therefore uses one for each class. This should lead anomalous inputs to gather around a different Gaussian than normal images which makes it easier to separate them later. The VAE basically acts as a preprocessing step to transform the data into a shape where anomalies can easily be separated from normal data using clustering. Similar to a normal VAE, this variant tries to estimate $p_\theta(x)$ with a Gaussian in the latent space. But unlike the normal VAE the Gaussian gets conditionally chosen depending on the kind of data (normal/anomaly). This follows the idea of some other approaches (e.g. [40]) which showed that providing some outliers during training can help to train the model to differentiate better between normal and anomalous data. For the CL-VAE [61], the ELBO becomes

$$\mathcal{L}_{ELBO}(\theta,\phi;x) = \mathbb{E}_{z_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x,y)||p_\theta(z|y)) \tag{4.4}$$

with a weight parameter $\beta$ to balance the 2 terms. Because y is discrete the KL-divergence can be expressed as

$$D_{KL}(q_\phi(z|x,y)||p_\theta(z|y)) = \mathbb{E}_{z|y}[\log \frac{q_\phi(z|x,y)}{p_\theta(z|y)}] = \sum_{y=1}^{N} p_\phi(z|y) \log \frac{q_\phi(z|x)}{p_\theta(z|y)} \tag{4.5}$$

to produce a closed form solution of this, [61] used $q_\phi(z|x)$ as a Gaussian conditioned on $x$ and $p_\theta(z|y)$ as Gaussian with $\mu_y$ and variance of 1

$$q_\phi(z|x) = \frac{\exp\{-0.5||\frac{z-\mu(x)}{\sigma(x)}||^2\}}{\prod_{i=1}^{d}\sqrt{2\pi\sigma_i^2(x)}}, \tag{4.6}$$

$$p_\theta(z|y) = \frac{\exp\{-0.5||z-\mu_y\}}{(2\pi)^{d/2}} \tag{4.7}$$

together with the reparameterization trick to simplify the expression above to

$$\log \frac{q_\phi(z|x)}{p_\theta(z|y)} = -0.5\sum_{i=1}^{d}\log\sigma_i^2(x) + \frac{1}{2}||z-\mu_y||^2. \tag{4.8}$$

This form of the KL-divergence is used in the CL-VAE to condition the latent space to form multiple clusters around the assigned Gaussians. As shown in [61], the CL-VAE is capable of producing a latent space where the data forms multiple clusters, depending on the class label. For the reconstruction part of the ELBO, the MSE between the input $x$ and its reconstruction $\hat{x}$ was used. This is given by taking the mean of the squared l2-distance between $x$ and $\hat{x}$.

$$MSE = \frac{1}{N} l(x, \hat{x}) \quad [4.9]$$

$$l(x, \hat{x}) = \{l_1, ..., l_N\},\ l_n = (x_n - \hat{x}_n)^2. \quad [4.10]$$

The variational autoencoder used in the implementation of the CL-VAE unfortunately only consists of one hidden fully connected layer for encoder and decoder and one layer for the parameters $\mu$ and $\sigma$. While such an architecture may be enough for small 28 X 28 grey-scale images like MNIST it is not suited for larger more complex images. The next chapter will go into more detail about the VAE architecture used in this thesis.

### 4.3 VAE Architecture

For the VAE, multiple architectures have been tried out. While models with discrete latent space like VQ-VAE [78, 67] show good reconstruction and compression capabilities, they are based on a uniform prior distribution over the discrete latent codes, this results in a constant kl-divergence which therefore doesn't play a role in the loss function. Because the kl-divergence in this approach is the part which conditions the latent space to form separate clusters, a VAE as proposed by [46] without a qunatized latent space was chosen. For the VAE i tried out multiple architectures and decided to use one with residual blocks for the encoder and decoder. As shown in previous works, residual networks are easier to train and optimize especially if the network gets deeper [36]. The encoder/ decoder architecture consists of multiple residual blocks which reduce the dimensionality of the input and extract features from it. In comparison to other network architectures, which often use a series of convolutional layers followed by batch-normalization and an activation function (most of the time ReLU), residual networks have shortcut connections in their individual building blocks. These shortcut connections connect the input to the output of the residual block and therefore skip the layers in between. A residual block can be formally defined as

$$y = \mathcal{F}(x, \{W_i\}) + x \quad [4.11]$$

where $x$ is the input and $y$ the output of the block. $\mathcal{F}(x, \{W_i\})$ describes the residual mapping to be learned and $\mathcal{F} + x$ describes the shortcut connection with element wise addition which means their dimensions must be equal. If the input and output dimension are not equal, there needs to be a transformation in the shortcut to make sure $x$ and $\mathcal{F}(x, \{W_i\}$ have the same number of dimensions in order to perform element wise addition of the two resulting feature maps [36].

$$y = \mathcal{F}(x, \{W_i\}) + W_s x \quad [4.12]$$

The residual blocks in the encoder are made out of a 2D convolutional layer, batch-normalization layer, randomized leaky ReLU, an average pooling layer and another 2D convolutional layer followed by batch-normalization. After these layers the skip connection gets added to the output

and a final randomized leaky ReLU is applied (see figure 4.6). Because of the pooling layer, the skip connection also passes through a convolution and average pooling layer in order to reduce its dimensions to perform the addition. The convolutional layers have a 3 x 3 kernel with stride 1 and padding 1 and are used to extract features from the image. The batch-normalization layer normalizes the activation values which speed up training and makes it more stable [44]. The randomized leaky ReLU is a special version of the normal ReLU which for a given input $x$ is defined as $x$ if $x \geq 0$ and 0 otherwise. The randomized leaky ReLU for a input $x$ is defined as $x$ if $x \geq 0$ and $ax$ otherwise where $a$ is sampled from a uniform distribution. This version can have some advantages over the standard ReLU [85]. The decoder architecture is similar to the encoder with the difference that the pooling layers are replaced by upsamling layers. Because the input data is in the range between 0 and 1, the decoder output is passed through a final 3 x 3 convolution followed by a sigmoid activation function to generate outputs that are also in the range 0-1. For generation of the bottleneck parameters $\mu$ and $\sigma$, a 2 x 2 convolution with stride 2 was used for each parameter. Compared to fully connected layers, convolutional layers in the bottleneck resulted in a better reconstruction which indicates that more information is stored in the latent variables. Furthermore, using fully connected layers to generate the distribution parameters makes it harder to increase the number of latent variables due to the huge amount of parameters that comes with them. This can cause memory problems if the latent space gets too large and therefore also limits the information which can be stored in it. From the $\mu$ and $\sigma$ feature maps, the latent space $z$ gets sampled using the reparametrization trick described earlier. The encoder consists of five residual blocks where every block doubles the number of channels and the resulting feature map is passed to the 2 convolutional layers which generate the distribution parameters. The decoder consists of six residual blocks followed by the final output layer (conv + sigmoid). The model has 2 more parameters $\mu_1$ and $\mu_2$ which are the means of the two prior distributions for the normal and anomaly data. These two learnable parameters are initialized random and lie in the interval [0,1). The basic structure of the VAE, like the residual blocks, was taken from the implementation provided by [2].

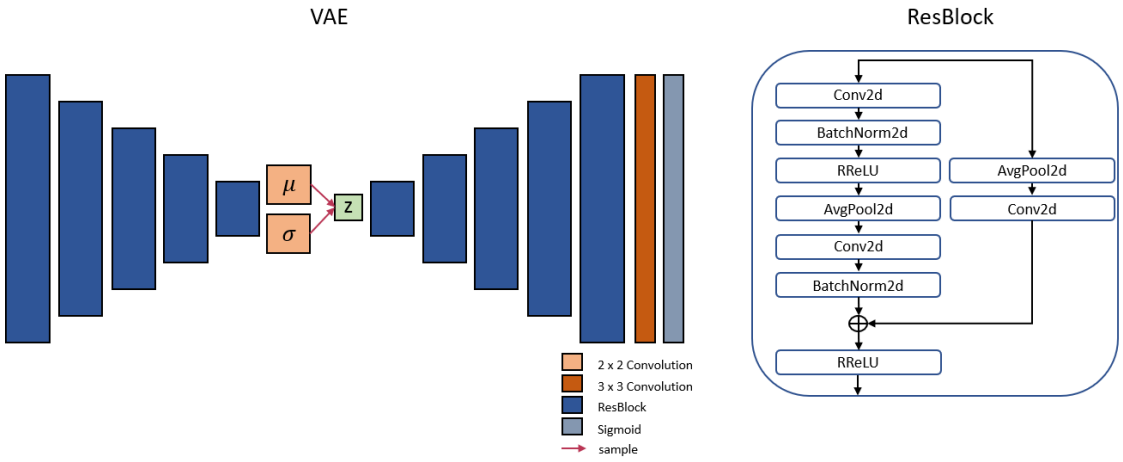



Figure 4.6: The left image shows the overall architecture of the VAE used in this thesis. The image on the right shows the components of the ResBlock.

To furthermore increase the separation of normal and anomalous data, experiments with two addi-

tional loss terms were carried out.
The first one is a distance loss $\mathcal{L}_{distance}$ which is used to maximize the distance between the two mean parameters in order to push the clusters away from each other. The negative l1-norm

$$\mathcal{L}_{distance} = -\|\mu_1, \mu_2\|_1 = -|\mu_1 - \mu_2| \quad [4.13]$$

is therefore added to the loss function. If the mean values of the two distributions are further away from each other clustering the latent space into normal and anomalous data should be easier. The second term is similar to the method proposed by Yang et al. [86]. They propose an optimization criterion for DNN based dimensionality reduction which trains the model to produce a clustering-friendly latent representation. They include a term in their loss function which minimizes the distance between every data point and its assigned cluster centroid given by the k-means algorithm. Because the mean parameters of the prior distribution can already be seen as the cluster centroids, using k-means to generate the centroids and cluster assignments is not necessary for the method used in this thesis. To generate a clustering friendly latent space, the term was added to the loss function which trains the model to minimize the squared l2-distance between each point in the latent space and the mean of its assigned distribution which is given by its label (normal/anomalous). For every point $z_i$ in the latent space, the loss is defined as

$$\mathcal{L}_i = (\mu_i - z_i)^2 \quad [4.14]$$

where $z_i$ is the latent representation of the data and $\mu_i$ the mean from its corresponding distribution. The overall loss is given by

$$\mathcal{L}_{cluster} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i \quad [4.15]$$

and added to the loss function of the VAE. The effects of these two terms on the anomaly detection task are examined later in the evaluation chapter.

## 4.4 Feature Loss

As mentioned by others [74, 43], measuring the reconstruction of an image by the pixel-wise error between the original and the reconstructed image (e.g. MSE-loss) is not an ideal measurement for reconstruction capability. The reason for this is that only pixel errors are detected but the overall structure of the image is not taken into account. The results of such reconstruction measurements are extremely blurry reconstructions. To deal with this problem, Hou et al. [43] propose a special feature loss for training VAEs which was also used in this thesis in addition to the CL-VAE objective. For the experiments in this thesis, an implementation of the feature loss provided by [2] was used. Hou et al. [43] claim that the perception of an image is more important than a pixel-wise error to measure reconstruction quality because the same image offset by a few pixels has little perceptual difference to a human but can have a high pixel-wise error [43]. Even though the reconstruction of the image is not used as an anomaly measurement, it is important that the VAE

is capable to reconstruct the input image. The ability to reconstruct the input data assures that enough relevant information is captured by the latent variables. Because CNNs are able to capture rich feature representations of images they can be used to mimic human perception and are used to calculate the perception loss between two images. The authors in [43] propose to use a VGGNet [73] which was pre-trained on the ImageNet dataset for classification. The VGGNet consists of multiple blocks of convolutionnal layers with a ReLU activation function which are followed by a max pooling operation as shown in figure 4.8. The goal of these operations is to extract features from the image which are then passed to a series of fully connected layers with ReLU activation functions and a sigmoid at the end to perform the classification task. For the feature loss however, only the convolutional layers are useful to generate feature maps and there is no need for classification. Because of this, only the first part of the VGGNet without the fully connected layers at the end was used. The idea behind this method is to compare the hidden representations from the VGGNet given two images. Because the hidden representations can capture features like spatial correlations, small differences between the hidden representations of two images can show similar perception. The feature loss between two images $x$ and $\hat{x}$ is defined as the squared Euclidean distance between the feature representations of each layer (see Fig. 4.7). For the $l^{th}$ layer the feature loss $\mathcal{L}^l_{feat}$ is given by

$$\mathcal{L}^l_{feat} = \frac{1}{2C^l W^l H^l} \sum_{c=1}^{C^l} \sum_{w=1}^{W^l} \sum_{h=1}^{H^l} (\Phi(x)^l_{c,w,h} - \Phi(\hat{x})^l_{c,w,h})^2 \quad [4.16]$$

where $\Phi(x)^l_{c,w,h}$ is the representation of the $l^{th}$ layer generated from image $x$ and $C^l, W^l, H^l$ represent the number of filters, the height and the width of the feature map produced by the $l^{th}$ layer. The overall feature loss is then given by combining the loss of the individual layers.

$$\mathcal{L}_{feat} = \sum_l \mathcal{L}^l_{feat} \quad [4.17]$$

Because the VGGNet was pretrained on RGB-images, only the three RGB-channels of the output image, without the discrepancy image, were used to calculate the feature loss. The overall objective of the model is to minimize $\mathcal{L}_{VAE}$, which is given by:

$$\mathcal{L}_{VAE} = \mathcal{L}_{ELBO} + \mathcal{L}_{feature} \quad [4.18]$$

## 4.5 Data Selection

In order to select the right data to train and evaluate this approach normal and anomaly data is necessary. The anomaly set has to contain anomalous objects which are not included in the normal data. Lately there have been some datasets using synthetic data like StreetHazards [38] which consists of normal training data and test data where anomalous objects are placed in the generated environment. Unfortunately StreetHazards lacks image quality and synthetic datasets are often

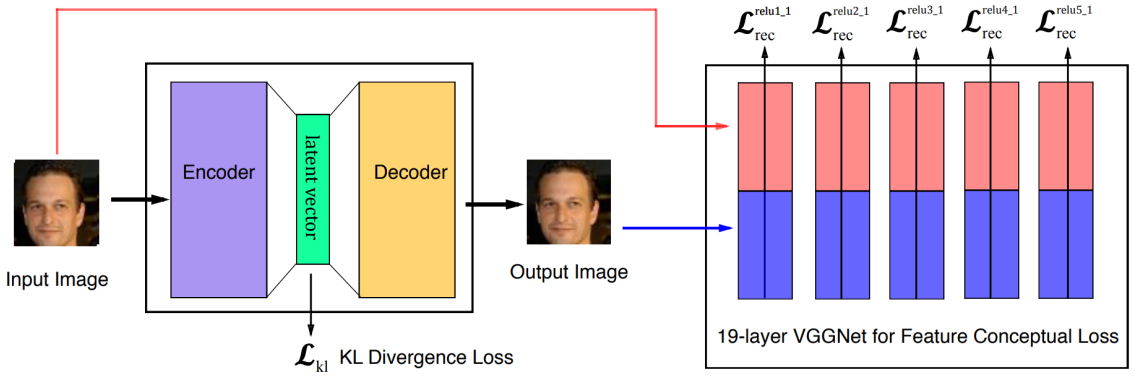



Figure 4.7: Overview of the feature loss. The feature loss gets calculated after every layer of the CNN (VGGNet) by taking the squared distance between the feature maps generated from the input and the reconstructed image. Reprinted from [43]

simplified versions of real live driving environments. While there are many datasets with real traffic images for autonomous driving, the Cityscapes dataset [21] offers a large collection of images with high diversity and was therefore used as the normal data in this thesis. The Cityscapes datasets consist of 5000 images which are recorded in 27 different cities. The dataset contains 2048 x 1024 images recorded from the ego perspective of a car driving in a mostly urban environment. The scenes were recorded in different cities at different time (month and daytime) and with different weather conditions to ensure a high variety in the data. While there are different weather conditions in the dataset, it doesn't contain extreme weather conditions like heavy rain or snow because as claimed by the authors such scenarios require special techniques and datasets. These images were manually selected to ensure a high diversity and capture real live driving scenarios with a high variety of background, objects and lighting and overall scene layout which makes it well suited to represent the normal class of data in this approach. The data is divided in a train, validation and test set. The data wasn't split randomly but based on criteria such as balanced distribution of location and population size as well as time of the year to ensure that every split captures a wide variety of different scenarios. The train, validation and test sets contain 2975, 500 and 1525 images and are kept in this split during the training procedure. Because Cityscapes has a wide variety of images from real life driving scenarios and doesn't contain any unusual situations or objects it is well suited to represent the normal data in this approach.

Because the method used in this thesis requires some outliers during training in order to condition the latent space of the VAE, some anomaly datastes were used in addition to Cityscapes. These anomaly datasets should contain images with objects different to the ones in Ciyscapes or images where the appearance of the scene differs from the normal images. The first anomaly dataset chosen is the LostAndFound dataset [66]. LostAndFound contains images from 13 different street scenes with different road obstacles. The obstacles vary in size and type and are placed in the driving corridor of the vehicle. The objects are considered anomalous and so an image containing these should be classified as anomaly. Because the images are individual frames from video sequences, there are frames where the anomalous object is very far away and barely or not at all

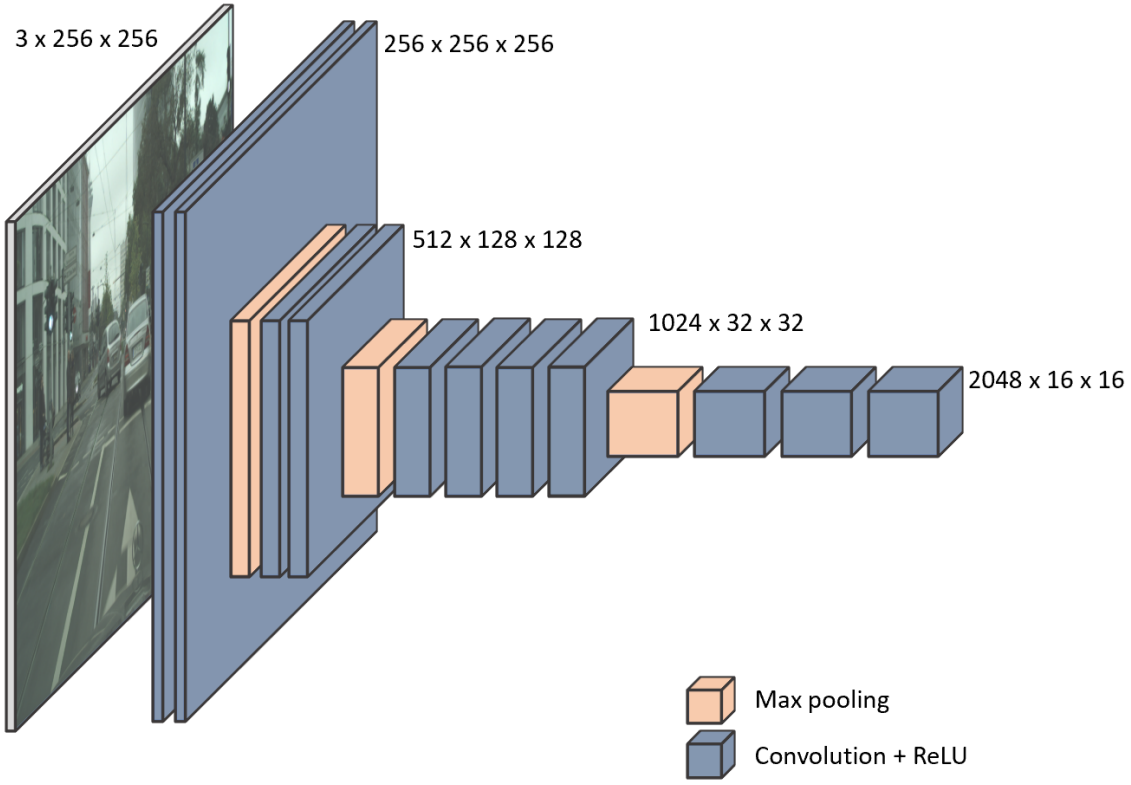



Figure 4.8: Architecture of the VGGNet to calculate the feature loss of a 256 x 256 rgb-image. The weights were taken from a network which was pre-trained on ImageNet and the the feature loss gets calculated after every convolution + ReLU block

visible. To ensure that each image has an anomalous object in it which can be detected, the dataset gets filtered and only images where the amount of anomalous pixels is greater than 3000 are used for training, validation and testing. Because the dataset contains multiple images from the same environment where only the anomalous object changes, i further manually filtered the dataset so that it contains only a few images with different anomalies from every environment. This is done in order to prevent the model from overfitting to the environment which stays the same in multiple images. This could lead to the model just recognizing the surroundings without paying attention to the anomalous objects in the image. In the LostAndFound dataset children crossing the road or bicycles lying on the road are also considered anomalous. The scenes where children are the anomalies were removed because they are also present in Cityscapes and considered normal. The scenes with bicycles are kept as anomalous even though there are also bicycles in Cityscapes. The reason for this is that a person riding a bicycle is common in real world driving scenes but a bicycle lying on the street is unusual and is therefore considered anomalous because it appears in a different context than in the normal Cityscapes images. After manually selecting only a few images from every scene the dataset contains a total of 172, 99 and 64 images for the train, validation and test set. To assure that the model does not only learn a certain type of anomaly or recognize the environment of the anomaly dataset it is important to provide a wide variety of anomalies and scenes in the training data to prevent overfitting.

To get more diversity, the RoadAnomaly21 from the SegmentMeIfYouCan benchmark [18] was used in addition to LostAndFound as anomaly data. RoadAnomaly21 contains 110 images of resolution 2048 x 1024 or 1280 x 720. Each of these images contains at least one anomalous object like an unknown vehicle or animal which is not present in the normal data (Cityscapes). These anomalous objects can appear everywhere in the image and vary in size. Because these images were selected from web sources, they contain a wide variety of environments making it more diverse and less likely that the model just learns to recognize a certain environment from the anomaly data. The combination of LostAndFound and RoadAnomaly21 provides a wide variety of objects which are not included in the Cityscapes dataset and therefore considered anomalous. The RoadAnomaly21 dataset is split into a train validation and test set with the train set containing 70%, the validation set 20% and the test set 10% of the images. The LostAndFound dataset has a train and test split which is predefined. The train split is used for training and the half of the test images are used for testing and the other half for the evaluation set.

## 4.6 Model Training

For the training of the VAE, the generated discrepancy images were used together with the rgb-images. The model was trained on the training data described in chapter 4.5 for 100 epochs and evaluated on the validation set. The model weights with the lowest validation loss were saved for later evaluation. For training a batch size of 12 was used and the images were down scaled to 256 x 256 in order to avoid memory problems. The VAE training was performed on a NVIDIA GeForce RTX 3090. I experimented with different learning rates and used a learning rate of 1e-4 with the ADAM [45] optimizer. Higher learning rates resulted in exploding gradients and a kl-divergence which is increasing rapidly to infinity. While different methods like gradient clipping were tried out, choosing a lower learning rate between 1e-4 and 1e-5, depending on the size of the model and the number of parameters, solved this issue. During the training, the learning rate was decreased linearly to allow the optimizer to make smaller steps later in the training and converge to a minimum. Furthermore this reduces the risk of exploding gradients later in the training.

# 5 Evaluation

In this chapter, the approach described in chapter 4 will be evaluated. For that, a description of the evaluation data is provided as well as an evaluation of the VAE, discrepancy module and the ability to detect anomalies in the latent space of the VAE.

## 5.1 Test Data

A diverse collection of anomalous images is necessary to test the capability of the model to detect anomalies in real world driving driving scenes. For testing, the test data from the dataset introduced in section 4.5 was used. This data consists of parts of the LostAndFound dataset [66] and the RoadAnomaly21 dataset from the SegmentMeIfYouCan benchmark [18]. These images all contain an anomalous object and should therefore be detected and classified as anomaly. Images from these datasets are also used in the model training as anomalies which could lead to the model just memorizing the overall scenery of the different datasets and not the anomalous object in them. While the RoadAnomaly21 dataset has a lot of different images and environments due to the images in it being collected from the web and therefore not having a specific environment where all images are recorded, the images still look a little bit different than the normal Cityscapes [21] images. A similar issue arises for the LostAndFound dataset where the images are frames from a video and show different anomalous objects but in similar environments. Compared to Cityscapes, which consists of metropolitan images, the images in LostAndFound are recorded in more rural areas. This could lead the model to just learn to detecting the domain shift and not the anomalies itself.

To evaluate the detection of unknown objects, the publicly available FS Static images from the Fishyscapes dataset [10] were used in addition to the two other datasets. These images are based on the Cityscapes validation set and therefore have the same environment as the normal data which prevents the detection of a distribution shift. These images are created by inserting novel objects. These objects are from classes that cannot be found in Cityscapes like airplane, boat or dog. The objects are randomly sized and positioned in the image. The positioning is dependent on the class of the object, while mammal classes have a higher probability of appearing on the lower half of the image, classes like birds have a higher chance to be placed in the upper half. In order for the objects to match the characteristics of their environment the authors of [10] used several processing steps to e.g. adapt shadows and lighting. Furthermore they used different degrees of blending, which makes the objects slightly transparent, to make the detection harder and provide some normal images where objects from known classes are inserted in order to check if the model just detects the insertion mechanism. Unfortunately the publicly available evaluation set only consists

of 30 images where 20 are anomalies and 10 are normal. The small size of the dataset makes it not suitable for training and it therefore was only used for testing (see 5.1 for example images from the datasets).

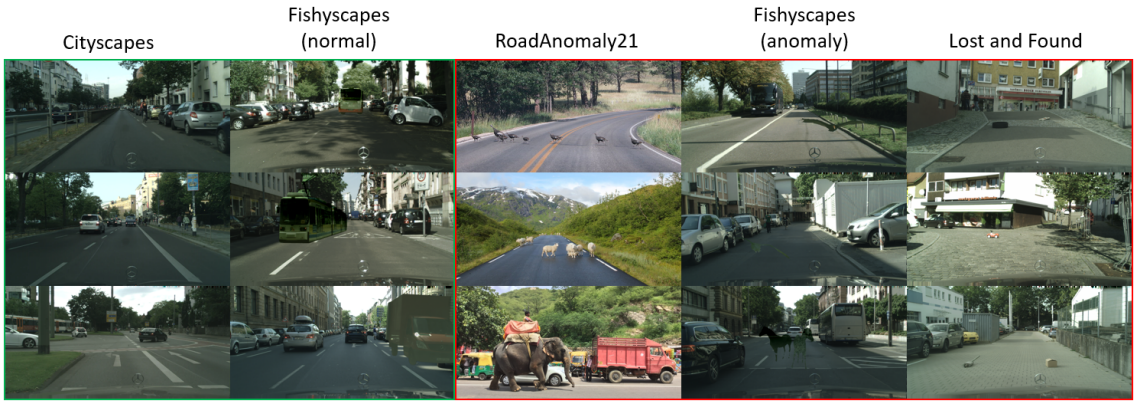



Figure 5.1: Example images from the different datasets

## 5.2 New Discrepancy Variant

This section examines the effect the newly created synthetic anomaly dataset has on the discrepancy module proposed by [51]. The discrepancy module was trained on this new dataset and used to create the discrepancy maps which were used together with the RGB-image as input to the VAE. As described in chapter 4.1.3, in the original version of the synthetic anomaly dataset the class labels of randomly selected objects are replaced by a random class which leads to the model detecting rare classes as anomalies. In the new version of the dataset the replacement class is not selected randomly but depending on the number of images in the Cityscapes dataset which contain this class. If a class appears in only a few images it has a lower chance of being selected as the replacement class than a class which appears in more images. The discrepancy module trained on the new dataset is compared with the version used in [51]. For comparison, the provided weights of the original version are used. For the discrepancy module Cityscapes is used as normal data where the model should not detect anything. The model generates a per pixel anomaly score which lies between 0 and 1 for a gray scale discrepancy map. If the model works perfectly all images from the normal dataset Cityscapes should have the value 0 for every pixel. To compare the new model to the original one, discrepancy maps from the Cityscapes test set were created and the mean pixel value of every image was calculated. Figure 5.2 shows that the average pixel value decreases for the variant trained on the new synthetic dataset which shows that less pixels are wrongly detected as anomaly. Figure 5.3 shows the effect the new synthetic anomaly training set has on rare classes like "Bus". Choosing the new variant, results in lower anomaly scores for rare but normal classes in the Cityscapes dataset.

The evaluation on the anomaly datasets LostAndFound and RoadAnomaly was kept like described in [51]. For the LostAndFound test set the evaluation was performed on downscaled images (1024x512) where the ego vehicle was excluded from evaluation. As shown in figure 5.4, the new

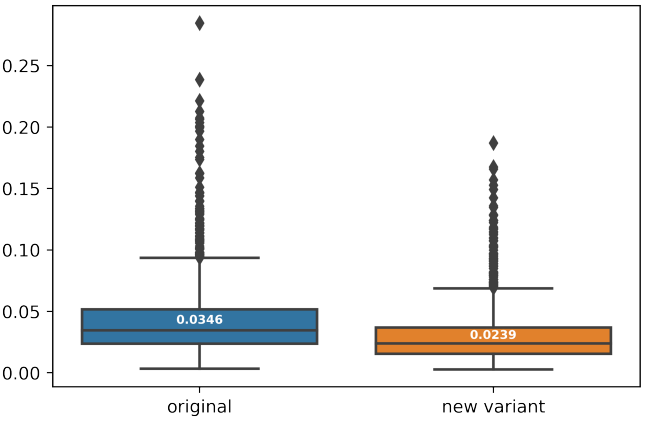



Figure 5.2: Box plot of the mean pixel value for every discrepancy map generated from the Cityscapes test set. The median decreases from 0.0346 to 0.0239 for the new variant which shows that less pixels are wrongly detected as anomaly. A perfect model should have the value zero for all images.

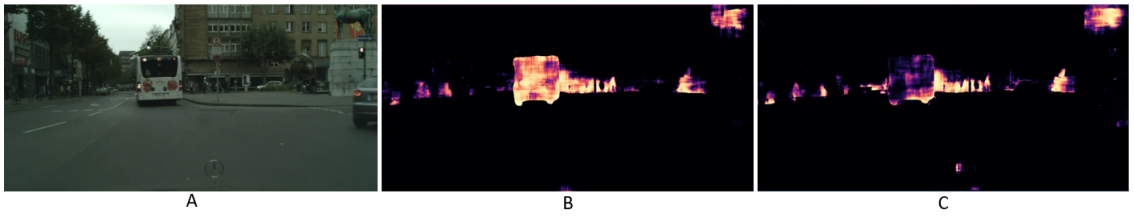



Figure 5.3: A: original image from a dataset where "Bus" is a rare class, B: discrepancy map with a model trained on a synthetic anomaly dataset with random replacement class, C: discrepancy map with a model trained on a synthetic anomaly dataset where the replacement class is selected based on its rarity.

version of the discrepancy module has similar performance in the anomaly detection task as the original version. On the LostAndFound test set it shows improved anomaly detection capabilities while the detection of anomalies on the RoadAnomaly dataset slightly decreases.

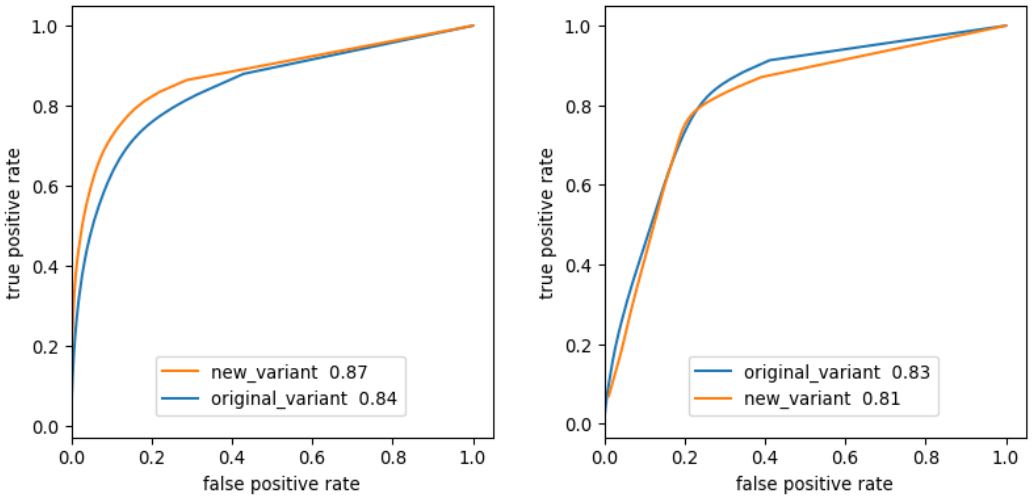



Figure 5.4: ROC curves for the LostAndFound test set (left) and the RoadAnomaly datset (right).

## 5.3 VAE

This section covers the evaluation of the VAE and looks at the influence of specific parameters on the overall model performance. Important parameters of the model are the latent space size and the $\beta$ weight for the kl divergence. The size of the latent space limits the amount of information that can be stored in it. If the model has only a few latent variables, it stores only the most relevant information about the input image and small details can be lost. This results in less accurate reconstructions the smaller the latent space gets (see Fig.5.5). To evaluate the reconstruction capabilities the Fréchet Inception Distance (FID) [41] was used. FID is used to evaluate the quality of generated samples by calculating the Fréchet distance between two Gaussians of the real and generated data. First, the feature representations of the data are generated. The Gaussians for the real and generated data are obtained by estimating the mean and covariance of these feature representations. This score is more consistent with human judgment compared to other metrics like the Inception Score [70] and therefore used in the evaluation [56]. It is worth noting that in the experiments described in this thesis, the PyTorch implementation [72] of the Fréchet Inception Distance (FID) score was used which has slightly different results than the original tensorflow implementation. A lower FID score means the images are closer to the real ones.

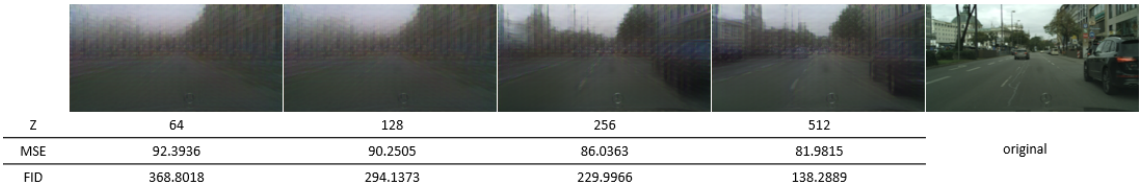



Figure 5.5: Reconstructions of Cityscapes test images. The latent feature map has the size $z*4*4$. Below each image, the FID score and the average MSE for the Cityscapes test dataset are given (lower means better).

The weight of the kl-divergence also plays an important role. A larger weight leads to less accurate reconstructions as more focus is put on the regularization part of the loss function. The effects, the $\beta$-factor of the kl-divergence (see equation 4.4) on the reconstruction capabilities of the VAE are shown in figure 5.6.

The effects that these parameters have on the latent space and the anomaly detection task in it are

| $\beta$ | 1 | 0.1 | 0.01 | 0.001 | |
|---|---|---|---|---|---|
| MSE | 81.9815 | 69.8002 | 65.8637 | 64.0007 | original |
| FID | 138.2889 | 48.2382 | 43.2463 | 52.8207 | |

Figure 5.6: Reconstructions of Cityscapes test images. The latent feature map has the size $512 * 4 * 4$. $\beta$ denotes the factor that is used to up- or downscale the kl-divergence. Below each image, the FID score and the average MSE for the Cityscapes test dataset are given.

examined in the next chapter.

In order to ensure that the VAE encodes all relevant information about the image in the latent variables, the VAE needs to be able to reconstruct the input data from the latent space. The feature loss explained in chapter 4.4 was therefore chosen to create a more accurate reconstruction of the input image. As mentioned by others [74, 43], a pixel by pixel loss between input and output data is not suitable for measuring perceptual difference and often results in extremely blurry reconstructions. As shown in figure 5.7 the VAE, trained with an additional feature loss term, produces sharper and higher quality images than a VAE trained with only the MSE as a reconstruction loss.

Figure 5.7: Reconstruction from a VAE trained with (right) and without (left) the feature loss. The VAE used for these images has a latent space size of 512*4*4 and a $\beta$ value of 0.01

The images in figure 4.7 were created from the Cityscapes test data and have a FID score of 263.9314 for the VAE without, and 65.8637 for the one with feature loss. Even though the right image in figure 5.7 is noticeable sharper and closer to the original one based on a human perception, the left one still has a slightly better average MSE compared to the right one (left: 61.0244, right: 65.8637). This shows that the MSE is less suited to evaluate the perceptual quality of generated images.

## 5.4 Anomaly Detection

This chapter looks at the effects the individual model parameters have on the latent space and the ability to detect anomalies in it. In order to classify a point in the latent space as anomalous or normal the k-means algorithm was used for clustering. All the latent space visualizations in this chapter are created by using PCA to reduce the number of dimensions to two. As already mentioned, a larger latent space can store more information and therefore results in a more accurate reconstruction. The size of the latent space also has influence on the anomaly detection capabilities in it. Experiments with different sizes of the latent space showed that not only the reconstruction capabilities improve but the separation of normal and anomalous data also increases and makes clustering more feasible. During most of the experiments, the latent feature map has the size $512 * 4 * 4$ as it gave better results compared to small ones like $64 * 4 * 4$ (see Fig. 5.9). Because k-means is a distance based algorithm and distance measurements get less meaningful in higher dimensions [9, 5], reducing the number of dimensions by applying PCA improved the detection of anomalies when using a larger latent space (2 dimensions for the experiments shown below). As shown in figure 5.9 this approach was able to detect most of the data from the anomaly dataset. For a VAE with a latent space size of $512 * 4 * 4$ and a $\beta$ factor of 1, clustering the latent space resulted in 93 true positives (TP), 1 false negative (FN).

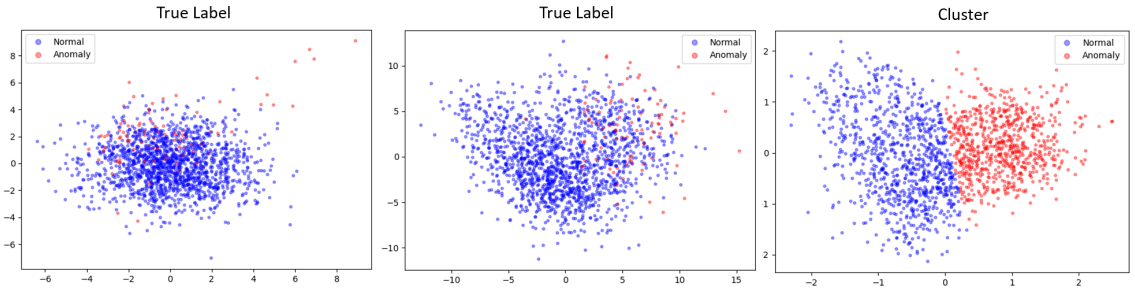



Figure 5.8: The left image shows the latent space of a VAE with a latent feature map of size $64 * 4 * 4$. This latent space is significantly less suited to detect anomalies by clustering compared to the one in the middle which is produced by a VAE where the latent space has the size $512 * 4 * 4$. The image on the right shows the latent space from the middle after clustering.

even though the anomalous data points can be detected by clustering, it also produces 665 false positives (FP) and 870 true negatives (TN). When looking at the $\beta$ factor which is the weight of the kl-divergence, different values gave slightly different results. A Kl-weight smaller than 0 (e.g. 0.01) resulted in a slightly lower false positive rate.

| $\beta$ | 1 | 0.1 | 0.01 | 0.001 |
|---|---|---|---|---|
| FPR | 0,4332 | 0,3231 | 0,3557 | 0,4065 |
| TPR | 0,9894 | 0,9681 | 1 | 1 |

Adding other terms like the described cluster loss (see equation 4.15) did not reliably reduce the number of false positives. This could be due to the fact that the cluster loss minimizes the squared Euclidean distance between every point in the latent space and the mean of the prior

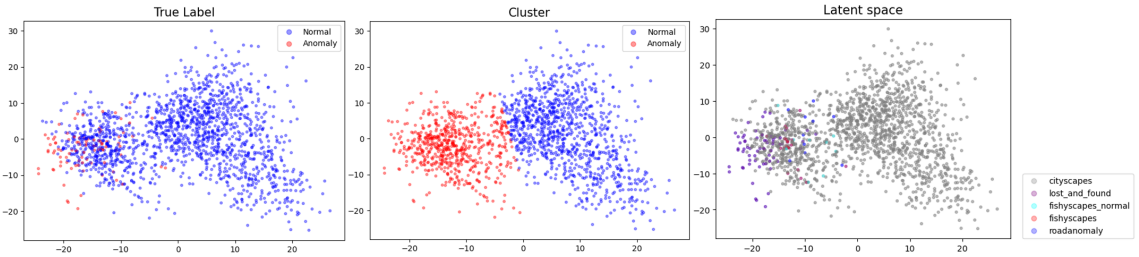



Figure 5.9: Latent space of a VAE with $\beta$ of 0.01. The left image shows the true labels of the data and the middle one the cluster assignments. The image on the right shows to which dataset the individual points belong

Gaussian to which it belongs. Because of the high number of dimensions this distance can become more meaningless which reduces the effectiveness of this term. When looking at the effect of the distance loss (see equation 4.13), it increased the distance between the two clusters to an extend where most of the anomaly data is closer to the normal cluster than to the anomalous. As shown in figure 5.10 the distance between the clusters has drastically increased but this structure of the latent space is not suited when trying to form an anomaly and a normal cluster. As seen in the figure, only the images from the RaodAnomaly dataset would be classified correctly while the Fishyscapes and LostAndFound data is closer to the normal cluster. This is probably due to the fact that images from LostAndFound and especially Fishyscapes are visually closer to Cityscapes images than the ones from the RoadAnomaly dataset and therefore get mapped closer to the Cityscapes data if the means of the two Gaussians are that far apart. Because of that, including that loss is not suited for detecting anomalous objects in images.

A lot of normal images which have to many highlighted regions in their discrepancy maps could

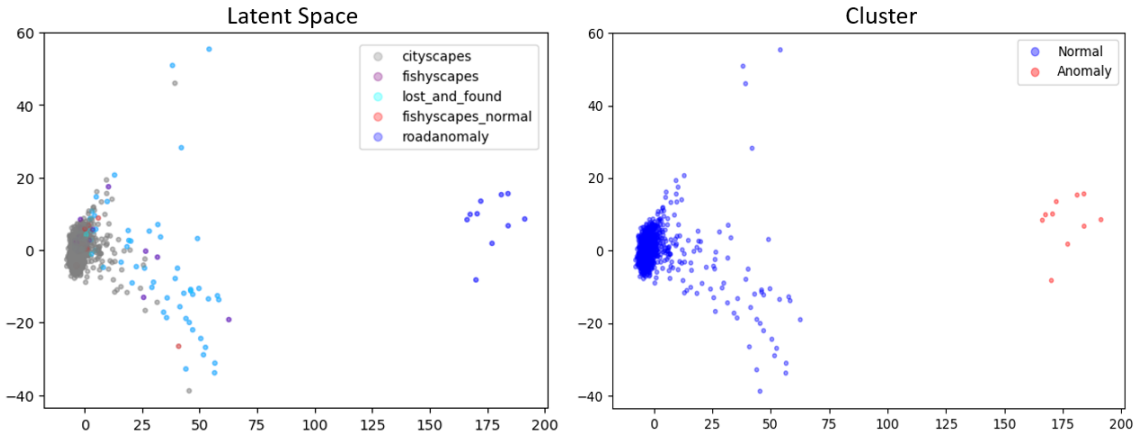



Figure 5.10: Latent space of of a VAE trained with additional distance loss

be an explanation for a high false positive rate. In order to evaluate the influence of the discrepancy maps, the mean pixel value for all images can be calculated and compared. A higher value means that more pixels are classified as anomalous by the discrepancy module. High false positive rates could be explained by poor results of the discrepancy module, if false positives have a higher mean pixel value than true negatives. When comparing the average mean pixel value, it turns

out that there is no significant difference between false positives and true negatives which means that their misclassification is probably not caused by the discrepancy maps. In fact, when trying out experiments without the discrepancy maps as additional VAE input, it showed that they have little influence on the anomaly detection task and the structure of the latent space, even though the average mean pixel value of the anomaly data is significantly higher than from the normal data.

# 6 Conclusion and Outlook

As shown, the approach described in this thesis is able to detect anomalous images in the latent space of a VAE. By clustering the latent space most anomalous images were detected even from datasets like Fishyscapes without a distribution shift. Similar to other approaches[25] this method produces high false positive rates. Additional model components like the distance or cluster loss did not reliably improve the anomaly detection capabilities in the latent space. The same goes for the discrepancy maps which made little difference in my experiments. In safety critical applications like autonomous driving, detecting as many anomalies as possible is far more important then receiving a low FPR because an anomalous sample which is classified as normal can have more dangerous consequences as a normal sample classified as anomalous. Because nearly all anomalies of the test data could be detected, this approach can be used to pre-select possible anomalies which then have to be further checked by human experts to sort out the false positive. Even though the anomaly data, which was selected for testing and training, has multiple anomalous objects appearing in different environments, only the data from the Fishyscapes dataset can ensure that the model detects the anomalous object itself and not the distribution shift. Because the publicly available Fishyscapes dataset is quite small, more research needs to be done on larger anomaly datasets without distribution shifts to further evaluate the generalization and anomaly detection capabilities of this model. Possible next steps to improve this approach could be to generate synthetic anomalies like in [25] to train the model or to use a method like the one used to train the discrepancy module to generate a synthetic anomaly dataset. This could eliminate the need for anomaly training data and could be a big improvement because it leaves more of the often rarely available anomaly data for testing the model. Furthermore more powerful VAE architectures can be tried out to improve the generation of the latent space and the detection capabilities in it. Because the discrepancy network makes little difference in the approach described above, a possible improvement could be to replace individual parts like the semantic segmentation network or the image resynthesis network with more powerful models. This could improve the highlighting of anomalous objects and decrease the highlighting of other image parts. By this, the discrepancy maps could potentially reduce the number of false positives while improving the detection capabilities if it can reliably only mark objects which are truly anomalous. In summary, anomaly detection is still a challenging filed where more research needs to be done to reliably detect anomalies in image data which is crucial for the safe deployment of autonomous vehicles

## Acronyms

**DNN** Deep Neural Network

**AD** Anomaly Detection

**CCD** Corner Case Detection

**ND** Novelty Detection

**OD** Outlier Detection

**OCC** One-Class Classification

**OSR** Open-Set Recognition

**OOD** Out-of-Distribution

**ID** In-Distribution

**CNN** Convolutional Neural Network

**AE** Autoencoder

**VAE** Variational Autoencoder

**VQ-VAE** Vector Quantized - Variational Autoencoder

**GAN** Generative Adversarial Network

**ELBO** Evidence Lower Bound

**PCA** Principal Component Analysis

**MSE** Mean Squared Error

**SVM** Support Vector Machine

**GMM** Gaussian Mixture Model

**GMVAE** Gaussian Mixture Variational Autoencoder

**CL-VAE** Conditioned Latent Space Variational Autoencoder

**FID** Fréchet Inception Distance

# A List of Figures

# B Bibliography


[1] CIFAR-10 and CIFAR-100 datasets.

[2] LukeDitria/CNN-VAE: Variational Autoencoder (VAE) with perception loss implementation in pytorch.

[3] MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges, May 2013.

[4] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara. Latent Space Autoregression for Novelty Detection. pages 481–490, 2019.

[5] C. Aggarwal, A. Hinneburg, and D. Keim. On the Surprising Behavior of Distance Metric in High-Dimensional Space. *First publ. in: Database theory, ICDT 200, 8th International Conference, London, UK, January 4 - 6, 2001 / Jan Van den Bussche ... (eds.). Berlin: Springer, 2001, pp. 420-434 (=Lecture notes in computer science ; 1973)*, Feb. 2002.

[6] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. GANomaly: Semi-supervised Anomaly Detection via Adversarial Training. In C. V. Jawahar, H. Li, G. Mori, and K. Schindler, editors, *Computer Vision – ACCV 2018*, Lecture Notes in Computer Science, pages 622–637, Cham, 2019. Springer International Publishing.

[7] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus. Variational Autoencoder for End-to-End Control of Autonomous Driving with Novelty Detection and Training De-biasing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 568–575, Oct. 2018. ISSN: 2153-0866.

[8] J. An and S. Cho. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. *undefined*, 2015.

[9] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When Is "Nearest Neighbor" Meaningful? *ICDT 1999. LNCS*, 1540, Dec. 1997.

[10] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena. The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation. *International Journal of Computer Vision*, 129(11):3119–3135, Nov. 2021.

[11] D. Bogdoll, M. Nitsche, and J. M. Zöllner. Anomaly Detection in Autonomous Driving: A Survey, Apr. 2022. Number: arXiv:2204.07974 arXiv:2204.07974 [cs].

[12] J.-A. Bolte, A. Bar, D. Lipinski, and T. Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 438–445, June 2019. ISSN: 2642-7214.

[13] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt. Systematization of Corner Cases for Visual Perception in Automated Driving. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1257–1264, Oct. 2020. ISSN: 2642-7214.

[14] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 93–104, New York, NY, USA, May 2000. Association for Computing Machinery.

[15] J. Cen, P. Yun, J. Cai, M. Yu Wang, and M. Liu. Deep Metric Learning for Open World Semantic Segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15313–15322, Oct. 2021. ISSN: 2380-7504.

[16] R. Chalapathy, A. K. Menon, and S. Chawla. Anomaly Detection using One-Class Neural Networks. *arXiv:1802.06360 [cs, stat]*, Jan. 2019. arXiv: 1802.06360.

[17] M. Chamseddine, J. Rambach, D. Stricker, and O. Wasenmuller. Ghost Target Detection in 3D Radar Data using Point Cloud based Deep Neural Network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10398–10403, Jan. 2021. ISSN: 1051-4651.

[18] R. Chan, K. Lis, S. Uhlemeyer, H. Blum, S. Honari, R. Siegwart, P. Fua, M. Salzmann, and M. Rottmann. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1, Dec. 2021.

[19] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009.

[20] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel. PixelSNAIL: An Improved Autoregressive Generative Model, Dec. 2017. Number: arXiv:1712.09763 arXiv:1712.09763 [cs, stat].

[21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, June 2016. ISSN: 1063-6919.

[22] T. T. Dang, H. Y. Ngan, and W. Liu. Distance-based k-nearest neighbors outlier detection method in large-scale traffic data. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pages 507–510, July 2015. ISSN: 2165-3577.

[23] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena. Pixel-Wise Anomaly Detection in Complex Driving Scenes. pages 16918–16927, 2021.

[24] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders, Jan. 2017. Number: arXiv:1611.02648 arXiv:1611.02648 [cs, stat].

[25] X. Du, Z. Wang, M. Cai, and Y. Li. VOS: Learning What You Don't Know by Virtual Outlier Synthesis, May 2022. Number: arXiv:2202.01197 arXiv:2202.01197 [cs].

[26] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58:121–134, Oct. 2016.

[27] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, Mar. 2021. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[28] S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier, T. Weißgerber, K. Bengler, R. Bruder, F. Flemisch, and H. Winner. Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. *IET Intelligent Transport Systems*, 8(3):183–189, 2014. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2012.0188.

[29] I. Golan and R. El-Yaniv. Deep Anomaly Detection Using Geometric Transformations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[30] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[32] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples, Mar. 2015. arXiv:1412.6572 [cs, stat].

[33] M. Grcić, P. Bevandić, Z. Kalafatić, and S. Šegvić. Dense anomaly detection by robust learning on synthetic negative data, Dec. 2021. Number: arXiv:2112.12833 arXiv:2112.12833 [cs].

[34] N. Harmening, M. Biloš, and S. Günnemann. Deep Representation Learning and Clustering of Traffic Scenarios, July 2020. arXiv:2007.07740 [cs, stat].

[35] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition, Dec. 2015. arXiv:1512.03385 [cs].

[36] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. ISSN: 1063-6919.

[37] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick. An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 644–651, July 2021. arXiv:2103.03678 [cs, eess].

[38] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song. Scaling Out-of-Distribution Detection for Real-World Settings. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8759–8773. PMLR, June 2022. ISSN: 2640-3498.

[39] D. Hendrycks and K. Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. July 2022.

[40] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep Anomaly Detection with Outlier Exposure. Sept. 2018.

[41] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[42] V. J. Hodge and J. Austin. A Survey of Outlier Detection Methodologies. page 42.

[43] X. Hou, L. Shen, K. Sun, and G. Qiu. Deep Feature Consistent Variational Autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141, Mar. 2017.

[44] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Mar. 2015. arXiv:1502.03167 [cs].

[45] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, Jan. 2017. arXiv:1412.6980 [cs].

[46] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes, May 2014. Number: arXiv:1312.6114 arXiv:1312.6114 [cs, stat].

[47] D. P. Kingma and M. Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, Nov. 2019. Publisher: Now Publishers, Inc.

[48] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 444–452, New York, NY, USA, Aug. 2008. Association for Computing Machinery.

[49] S. Kumano, H. Kera, and T. Yamasaki. Are DNNs fooled by extremely unrecognizable images?, Mar. 2022. arXiv:2012.03843 [cs].

[50] S. Liang, Y. Li, and R. Srikant. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. Feb. 2018.

[51] K. Lis, K. K. Nakka, P. Fua, and M. Salzmann. Detecting the Unexpected via Image Resynthesis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2152–2161, Oct. 2019. ISSN: 2380-7504.

[52] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, Dec. 2008. ISSN: 2374-8486.

[53] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps. Towards Visually Explaining Variational Autoencoders. pages 8642–8651, 2020.

[54] W. Liu, W. Luo, D. Lian, and S. Gao. Future Frame Prediction for Anomaly Detection – A New Baseline. pages 6536–6545, 2018.

[55] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based Out-of-distribution Detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475. Curran Associates, Inc., 2020.

[56] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are GANs Created Equal? A Large-Scale Study, Oct. 2018. arXiv:1711.10337 [cs, stat].

[57] M. Maurer. EMS-vision: knowledge representation for flexible automation of land vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, pages 575–580, Oct. 2000.

[58] P. Munjal, A. Paul, and N. C. Krishnan. Implicit Discriminator in Variational Autoencoder. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2020. ISSN: 2161-4407.

[59] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, June 2015. ISSN: 1063-6919.

[60] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena. Out-of-Distribution Detection for Automotive Perception. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2938–2943, Sept. 2021.

[61] E. Norlander and A. Sopasakis. *Latent space conditioning for improved classification and anomaly detection*. Nov. 2019.

[62] A.-A. Papadopoulos, M. R. Rajati, N. Shaikh, and J. Wang. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 441:138–150, June 2021.

[63] H. Park, J. Noh, and B. Ham. Learning Memory-Guided Normality for Anomaly Detection. pages 14372–14381, 2020.

[64] J. Park, J.-H. Moon, N. Ahn, and K.-A. Sohn. What is Wrong with One-Class Anomaly Detection?, Apr. 2021. arXiv:2104.09793 [cs].

[65] S. Park, G. Adosoglou, and P. Pardalos. *Interpreting Rate-Distortion of Variational Autoencoder and Using Model Uncertainty for Anomaly Detection*. May 2020.

[66] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester. Lost and Found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1099–1106, Oct. 2016. ISSN: 2153-0866.

[67] A. Razavi, A. van den Oord, and O. Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[68] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4393–4402. PMLR, July 2018. ISSN: 2640-3498.

[69] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. H. Rohban, and M. Sabokrou. A Unified Survey on Anomaly, Novelty, Open-Set, and Out-of-Distribution Detection: Solutions and Future Challenges, July 2022. Number: arXiv:2110.14051 arXiv:2110.14051 [cs].

[70] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[71] K. K. Santhosh, D. P. Dogra, P. P. Roy, and A. Mitra. Vehicular Trajectory Classification and Traffic Anomaly Detection in Videos Using a Hybrid CNN-VAE Architecture. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2021. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[72] M. Seitzer. pytorch-fid: FID Score for PyTorch, Aug. 2020.

[73] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, Apr. 2015. arXiv:1409.1556 [cs].

[74] G. Somepalli, Y. Wu, Y. Balaji, B. Vinzamuri, and S. Feizi. Unsupervised Anomaly Detection with Adversarial Mirrored AutoEncoders. *arXiv:2003.10713 [cs, stat]*, Jan. 2021. arXiv: 2003.10713.

[75] V. K. Sundar, S. Ramakrishna, Z. Rahiminasab, A. Easwaran, and A. Dubey. Out-of-Distribution Detection in Multi-Label Datasets using Latent Space of -VAE. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 250–255, May 2020.

[76] P. Tabacof and E. Valle. Exploring the space of adversarial images. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 426–433, July 2016. ISSN: 2161-4407.

[77] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 982–988, Sept. 2015. ISSN: 2153-0017.

[78] A. van den Oord, O. Vinyals, and k. kavukcuoglu. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[79] H. S. Vu, D. Ueta, K. Hashimoto, K. Maeno, S. Pranata, and S. M. Shen. Anomaly Detection with Adversarial Dual Autoencoders. *arXiv:1902.06924 [cs]*, Feb. 2019. arXiv: 1902.06924.

[80] H. Wang, M. J. Bah, and M. Hammad. Progress in Outlier Detection Techniques: A Survey. *IEEE Access*, 7:107964–108000, 2019. Conference Name: IEEE Access.

[81] L. Wang, D. Zhang, J. Guo, and Y. Han. Image Anomaly Detection Using Normal Data Only by Latent Space Resampling. *Applied Sciences*, 10(23):8660, Jan. 2020. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute.

[82] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. pages 8798–8807, 2018.

[83] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun. Identifying Unknown Instances for Autonomous Driving. In *Proceedings of the Conference on Robot Learning*, pages 384–393. PMLR, May 2020. ISSN: 2640-3498.

[84] J. Wurst, L. Balasubramanian, M. Botsch, and W. Utschick. Novelty Detection and Analysis of Traffic Scenario Infrastructures in the Latent Space of a Vision Transformer-Based Triplet Autoencoder. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 1304–1311, July 2021.

[85] B. Xu, N. Wang, T. Chen, and M. Li. Empirical Evaluation of Rectified Activations in Convolutional Network, Nov. 2015. arXiv:1505.00853 [cs, stat].

[86] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3861–3870. PMLR, July 2017. ISSN: 2640-3498.

[87] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized Out-of-Distribution Detection: A Survey, Aug. 2022. arXiv:2110.11334 [cs].

[88] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642, June 2020. ISSN: 2575-7075.

[89] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, July 2017. ISSN: 1063-6919.