TECHNISCHE UNIVERSITÄT DRESDEN

Computer Graphics and Visualization

Faculty of Computer Science

Institute of Software and Multimedia Technology
Chair of Computer Graphics and Visualization

Master Thesis

Out-of-distribution Detection for Microscopic Imaging

Erdem Ünal

Born on: 27th September 1996 in Düsseldorf
Matriculation number: 4870873

to achieve the academic degree

Master of Science (M.Sc.)

Referee
Prof. Dr. Stefan Gumhold

Supervisor
Dr. Thomas Neumann - Nishant Kumar

Submitted on: 25th October 2021

TECHNISCHE UNIVERSITÄT DRESDEN

Computer Graphics and Visualization

Faculty of Computer Science Chair of Computer Graphics and Visualization

Task for the preparation of a Master Thesis

Course:	Computational Modeling and Simulation - Visual Computing
Name:	Erdem Ünal
Matriculation number:	4870873
Matriculation year:	2019
Title:	Out-of-distribution Detection for Microscopic Imaging

Motivation

Microscopic imaging is a widely applicable tool for analyzing biological samples from many different fluids such as blood, urine, cell suspensions, etc. In-line, lens-free microscopic setups are particularly compact, making such analysis devices cheap to build and therefore easily accessible. Samples recorded with such devices can be efficiently and automatically analyzed using machine learning, especially deep learning, to recognize specific objects in samples such as cell types, sizes and states. However, it is well known that deep neural networks may yield unexpected outputs on inputs dissimilar from the training data. Such a situation can arise due to mistakes in sample preparation, due to sample contamination, or due to rare samples not included in the training data distribution. Recognizing such a situation have the following benefits:

1. The user can be informed about the uncertainty of an analysis so that the sample preparation routine can be checked or an additional investigation of the sample could be performed.
2. Rare samples could be automatically flagged, sent back for investigation by the manufacturer and finally annotated and fed back into the training pipeline in order to improve the efficiency of the preexisting classification based neural network.

In order to enable these applications, the master thesis aims to develop an out-of-distribution (OOD) detection method that robustly works on real-world microscopic data. To evaluate the method, real world data is provided by Anvajo GmbH, a manufacturer and developer of compact point-of-care devices featuring an in-line holographic microscope.

Goals

- Literature review on state-of-the-art Out-of-Distribution (OOD) and pre-processing approaches with focus on cell based datasets.
- Analysis of robust pre-processing steps such as cropping in order to standardize the microscopic data.
- Analyze and select the OOD detection approaches suitable for the microscopic data while fulfilling the requirements of the hardware.

- Train multiple OOD detection approaches to evaluate a fast and robust model that is capable of detecting anomalous microscopic image samples with high reliability.
- Evaluate the efficiency of the OOD detection methods and discuss the approaches on training as well as validation datasets.

Optional

- Optimize the meta-parameters of the evaluated OOD detection methods in order to obtain better efficiency results.
- Develop a new OOD detection approach and evaluate its efficiency with other methods.
- Compare the developed OOD detection model with the already existing classification based neural network model in terms of accuracy.

Referee:	Prof. Dr. Stefan Gumhold
Supervisor:	Dr. Thomas Neumann - Nishant Kumar
Issued on:	24th May 2021
Due date for submission:	25th October 2021

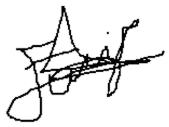
Prof. Dr. Stefan Gumhold
Supervising professor

Statement of authorship

I hereby certify that I have authored this Master Thesis entitled *Out-of-distribution Detection for Microscopic Imaging* independently and without undue assistance from third parties. No other than the resources and references indicated in this thesis have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the intellectual preparation of the present thesis. I am aware that violations of this declaration may lead to subsequent withdrawal of the degree.

Dresden, 25th October 2021

Erdem Ünal



Abstract

Deep Learning models are widely used in microscopic imaging. Unaware models can be easily fooled by unseen anomalous inputs and yield unexpected results. This thesis analyzes unsupervised out-of-distribution (OOD) detection methods to solve this problem. Baseline generative and classifier model-based OOD detection methods are elaborated and applied to a specific microscopic dataset. Results showed that Mahalanobis distance-based detector for pre-trained classifier provides an effective and applicable solution in terms of performance, efficiency and robustness. Generative models are concluded to be unfeasible to obtain distinguishable low dimensional feature representations. Reconstruction error-based anomaly detection method with generative models is shown to be ineffective regardless of used model and error metric. Statistical two-sample tests have a considerable potential to detect the shift between two image representations which can be obtained by various dimensionality reduction methods.

Contents

Abstract VII

Symbols and Acronyms XI

1 Introduction 1
1.1 Motivation 1
1.2 Key Components 2
1.3 Structure of the Thesis 3

2 Background 5
2.1 Preliminary Definitions 5
2.1.1 Definition of OOD Detection 5
2.1.2 OOD Detection vs Anomaly Detection 5
2.2 Device Information 6
2.2.1 Fluidlab 6
2.3 Urine Dataset 7

3 Related Work 9
3.1 Generative Model based OOD Detection 9
3.2 Discriminative Model based OOD detection 12

4 Methods and Implementation 15
4.1 Data Preparation 15
4.1.1 Train Dataset 15
4.1.2 Test Dataset 16
4.2 Unsupervised Anomaly Detection Basics 16
4.2.1 Principal Component Analysis (PCA) 16
4.2.2 Autoencoder 17
4.2.3 Reconstruction Methods for Anomaly Detection 18
4.2.4 Variational Autoencoder 20
4.2.5 Training AE&VAE 21
4.2.6 Likelihood Ratios Metric with an Autoregressive Model 22

4.3 Classifier based OOD Detection Approaches . . . 25
4.3.1 Maximum Softmax Probability (MSP) . . . 25
4.3.2 Calibrated MSP (ODIN) . . . 25
4.3.3 Mahalanobis distance-based OOD Detection . . . 28
4.3.4 Pre-trained Classifier . . . 30
4.4 Urine OOD Test . . . 31
4.4.1 Kolmogorov-Smirnov Test . . . 32

5 Results . . . 33
5.1 Evaluation Metrics . . . 33
5.2 Urine OOD Test Results . . . 33
5.2.1 Test Models . . . 34
5.3 Reconstruction Error Based OOD . . . 36
5.3.1 Likelihood Ratios Metric with PixelCNN++ Model . . . 37
5.4 Classifier Model Based OOD . . . 39
5.4.1 Why do MSP and ODIN fail? . . . 39
5.5 Mahalanobis Detector . . . 40
5.6 Kolmogorov-Smirnov Detector . . . 42
5.7 Proposed Solution: Double-threshold Mahalanobis Detector . . . 44
5.7.1 What makes Mahalanobis method effective and applicable? . . . 44

6 Conclusion and Further Work . . . 45
6.1 Conclusion . . . 45
6.2 Future Work . . . 46

Symbols and Acronyms

OOD	Out-of-distribution
IN	In-distribution
inlier	In-distribution sample
outlier	Out-distribution sample
Mahal	Mahalanobis Distance based Detector
KST	Kolmogorov-Smirnov Test
LLR	Likelihood Ratio
MSE	Mean Square Error
SSIM	Structural Similarity Index Metric
MSP	Maximum Softmax Probability
ODIN	Calibrated Maximum Softmax Probability
cgv-lab	Chair of Computer Graphics and Visualization
GPU	Graphics Processing Unit

1 Introduction

Machine learning approaches, especially deep learning, are now widely applicable in microscopic imaging to recognize and differentiate specific objects. A deep neural network (DNN) which is trained with noise-free, correctly labeled in large quantities of train images is capable to perform accurate classification when it is tested by test images sampled from the same distribution (in-distribution). However, it is ambiguous to estimate the possible outcomes when the test images sampled from a different distribution (out-of-distribution). Out-of-distribution (OOD) images do not belong to any class labels as in-distribution images and they are not intended to be tested for classification purposes. If such OOD images are mixed in the test dataset, the classifier DNN will try to label them into their class belongings and this can cause severe problems in real world data analysis such as microscopic imaging.

1.1 Motivation

DNNs do not know what they do not know. They try to classify any given input data without being aware if the input data belongs to their dataset distribution. Making DNNs aware of dissimilar looking OOD inputs is crucial to have in real medical sample tests. Considering the utmost importance of medical applications, a wrong prediction might cause misleading diagnosis and resulting a possible hazardous wrong treatment. This thesis aims to research Out-of-distribution (OOD) detection methods that can be implemented on real world microscopic data. The microscopic in-distribution and out-distribution real data is provided by Anvajo GmbH, a manufacturer and developer of in-line holographic microscopic devices called *fluidlab*. In short, these devices are composed of lens-free spectrometer and holographic microscope to capture images of cell culture of given sample fluids such as blood, urine etc. Then these images are fed into a machine learning segmentation model and the output of the segmented image patches are fed into a classifier DNN that classifies the given image patch into nine different classes of cells, crystals and urinary casts. However, some of these image patches are OOD inputs such as undefined artifacts in urine or unwanted contamination on the sample carrier. These devices are often used by veterinary doctors and the meticulousness of the analysis is highly dependent on the user. It is important to inform the user if there is an oddity in the analysis. An unaware classifier will not be able

to detect the abnormalities in the input image and more importantly, all the algorithms and classifications will be held in this unaware manner, thus the accuracy of the test will deteriorate reluctantly.

1.2 Key Components

Naive solutions to eliminate OOD images in the sample test, for instance, instructing users how to handle experiments neatly will not solve the problem even in theory. Artifacts and unexpected structures will always have the possibility to exist in urine samples. Therefore, a robust OOD detection algorithm is required to be added in the device that will be able to detect the OOD samples. The OOD detection methods in literature perform very well overall but they often use two completely different looking (different background, image distributions such as handwritten digits (MNIST [LeC10]) as in-distribution data and street view house numbers (SVHN [al11b]) as out-distribution data. However, it is not the case for the urine dataset. IN and OUT images are both sampled by the same image analysis algorithm. As a result, they share the same urine liquid background and some of them are not even semantically distinguishable by human eye. State-of-the-art (SOTA) OOD methods have never been tested in such noisy medical datasets in literature. To the best of our knowledge, this research study is going to be the first machine learning and OOD detection analysis have been done in microscopic images.

Furthermore, the device constraints have to be taken into account. The OOD detection methods in literature which are exhaustively discussed in the related work chapter can be grouped in two categories *Discriminative Model Based OOD Detection* and *Generative Model Based OOD Detection*. Proposed robust OOD detection algorithm should be applicable for the fluidlab device without requiring immense amount of memory and performance cost. In addition, anvajo has a software application called *datalab*. Unfeasible OOD detection methods which are impossible to compute in real time through device, can be implemented offline through this software application.

It is favorable to use Discriminative Model based OOD methods since they are applied on pretrained classifier networks and tend to require low computation cost. Therefore, they are easily applicable to the already trained and functioning classifier in fluidlab. The classifier performance in fluidlab has a considerable effect on the OOD detection performance. The training dataset is unbalanced over the classes and some of the classes do not have sufficient amount of images to train a robust DNN. Since images have noise and are semantically hard to distinguish, image labels are not perfectly annotated. Therefore, the classifier in the fluidlab does not have high accuracy per classes compared to the classifiers used in the literature.

Generative Model Based OOD methods are generally composed of high level neural architecture and tend to have higher accuracy in literature studies compared to discriminative methods. However, they require training of large and complex generative models. Even if they might not be used directly in the current device, they can gain remarkable insights of urine dataset which is never tested before. Last but not the least, the OOD images are not available in abundant number as they are rarely found compared to blood cells and crystals

in urine samples. This causes limited number of test scenarios for OOD detection analysis and low number of test samples for out-of-distribution images.

1.3 Structure of the Thesis

The structure of the thesis is composed of Introduction, Background, Related work, Methods, Results and Conclusion chapters respectively. In the next chapter Background, preliminary definitions related to OOD detection and a detailed illustration of the urine dataset together with device information will be explained. In the Related work chapter, baseline OOD detection methods that achieved competitive results in literature will be elaborated in two main parts. In the Implementation part, generative model based OOD detection methods are introduced following with baseline classifier based OOD approaches. Algorithms and model details are explained as a story-line to solve the anomaly detection problem of urine dataset. In Results chapter, multiple OOD detection methods are grouped and named as *Urine OOD Test*. Methods are compared based on multiple evaluation metrics and confidence score plots. The applicability of methods are discussed one by one and the best method, in terms of its applicability and accuracy, is chosen. Finally, in the conclusion chapter, final remarks regarding the OOD analysis and proposed detector are shared and possible further works available in literature that might have better impact for the solution is discussed.

2 Background

In this background part, the concepts of Out-of-distribution detection, outlier detection, anomaly detection and novelty detection in DNNs will be discussed. In addition, information about the product and how urine image patches were obtained have been explained.

2.1 Preliminary Definitions

2.1.1 Definition of OOD Detection

Out-of-distribution (OOD) detection can be defined as binary classification problem where the binary classes are two distinct distributions *in-distribution* (P_X) and *out-distribution* (Q_X) where X is the defined image space. We can define a mixture distribution $\mathcal{P}_{X,Z}$[Shi17] as a mix of in and out-distributions where Z defines the label space $Z = \{0, 1\}$. Then the conditional probability distributions are $\mathcal{P}_{X|Z=0} = P_X$ and $\mathcal{P}_{X|Z=1} = Q_X$. Then the OOD detection problem can be reformulated as, detecting an image X, which is drawn from $\mathcal{P}_{X,Z}$ distribution, is from P_X or not[Shi17].

2.1.2 OOD Detection vs Anomaly Detection

The name of Out-of-distribution (OOD) detection can be seen substituted with *outlier detection*, *novelty detection* or *anomaly detection* even though they do not correspond to same application. Outlier detection and novelty detection can be seen as subsets of anomaly detection. In novelty detection the test data includes only samples from in-distribution dataset but in outlier detection the test data is polluted by outlier samples. Therefore, outlier detection is unsupervised anomaly detection and novelty detection is semi-supervised anomaly detection. The OOD detection on the other hand, differs itself from anomaly detection based on the dataset. The datasets used in OOD as out-distribution are often composed of multi-class datasets with numerous samples which are collected at a different time and different conditions. Usually the datasets in OOD are not related to each other and for instance, if a multi-class MNIST dataset is cosidered as the inlier dataset then any

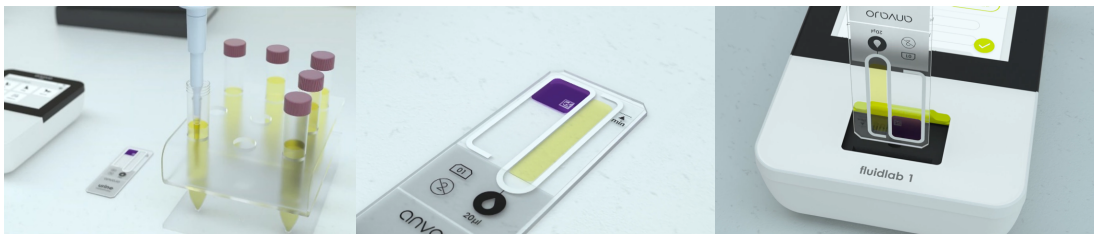
other distribution which is not seen by the trained neural network can be an outlier dataset such as FashionMNIST, SVHN, CelebA, random noise etc. However, the aim of this thesis is to differentiate between image patches that share a considerable amount of semantic pattern, in other words looking likely. Therefore, it is important to state the fact that urine IN and OOD data, compared to the IN and OOD data in literature, are the same as terminology but very different in practice. Time to time we will use Anomaly Detection and Out-of-distribution detection terms interchangeably as there is not any significant split meaning. Therefore, terms like anomolous data, OOD data, outlier data or simply outliers will be used interchangeably simply meaning image samples from out-of-distribution dataset. Similarly, normal data, IN data and inliers mean image samples from in-distribution dataset.

2.2 Device Information

This thesis work was made possible by collaboration with Anvajo GmbH. Anvajo is a Dresden-based technology company that develops, manufactures and sells fluid decoding devices called *fluidlab*. The company started working originally at TU Dresden 7 years ago and right now the team is working on one of the most compact device in the world for fluid analysis.

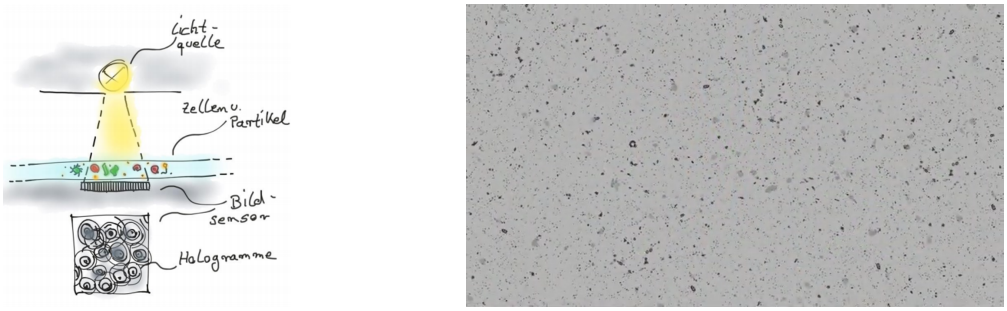
2.2.1 Fluidlab

The compact device Fluidlab can be interpreted as a combination of holographic microscopy, machine-learning-supported image analysis, mobile embedded hardware, function-integrated optics and sample carriers [Ste]. The device runs on a Raspberry Pi single-board computer. The embedded system runs on 1 GHz clock speed and includes 512 Mbs RAM where 256 Mbs reserved for CPU and 256 Mbs for GPU. It also offers on-board WiFi and Bluetooth for wireless communication. Aside from electronic specifications, sample carriers are the most important physical component of the device for OOD detection analysis. Sample liquids can be blood, urine, sperm or any liquid substance in room temperature. The workflow of the device is illustrated and can be interpreted as below. The sample liquid is dropped onto a very thin surface, replicating 2D plane- of sample carrier. The surface of the sample carrier has to be disinfected every time to avoid contamination. Such contamination is the main source of OOD data that made possible this thesis. Then the carrier is inserted into the fluidlab for the next image analysis part. Fluidlab has a small lens-less microscope that scatters light through on this very thin sample surface. The scattered light interferes with the surrounding light and this gives the hologram on the sensor. Reliable 3D reconstruction from the hologram is not possible as the sample is in a very thin surface that resembles 2D plane. Signal processing algorithms then reconstruct the actual 2D image (figure 2.2) from the shadows forming on the sensor. Reconstructed image is then fed into a segmentation network that detects semantic particles like cells and segments them by centering them into a square. The squared image patches are then transformed into a specified NxN format. The segmented and reshaped image patches are then fed into a classification network to label them. The labeled image patches then form the *urine dataset*.



(a) Urine sample obtained (b) Inserted onto sample carrier (c) Carrier inserted into Fluidlab

Figure 2.1: Step by step preparation of urine sample analysis



(a) Illustration of hologram microscopy (b) An example final obtained 1920 x 1080 resolution image

Figure 2.2: Image Analysis

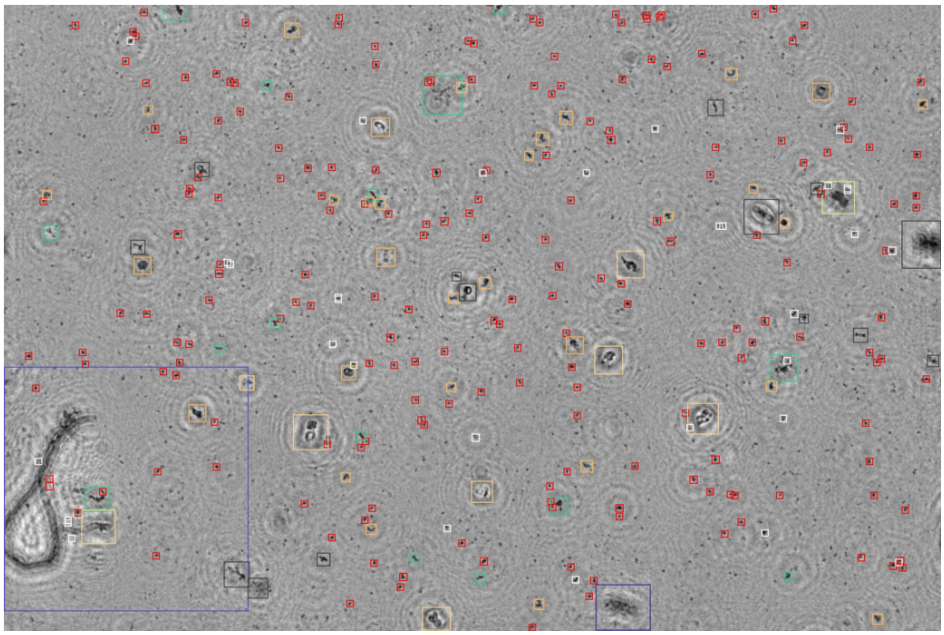


Figure 2.3: Segmented Image Patches where red squares represent red blood cell

2.3 Urine Dataset

The urine dataset is composed of in-distribution and out-distribution image patches. All samples are gray-scaled, 32x32 resolution images.

The in-distribution dataset is composed of 9 classes: red blood cells (RBC), white blood cells(WBC), squamous epithelial cell (sEC), non-squamous epithelial cell (nsEC), calcuimoxalat crystals (cCry), struvite crystals (sCry), unclassified crystals (uCry), hyaline casts (hCasts) and non-hyaline casts (nhCasts). The out-distribution dataset is *so far* composed of 17 classes

that have to be detected. It is still convenient to separate them into classes as OOD detection methods can detect one significantly better than the other class. The outlier classes are named based on what kind of contamination is added to the sample carrier. The outlier classes are artifact, blank urine, bubbles, cathair, condensation, dirt, dust, feces, fingerprint, human hair, lipid droplets, lotion, pollen, semifilled, void, wetslide and yeast.

Urine dataset is quiet imbalanced, as you can see in figure 2.4, there are 33 times more blood cell images available than casts or crystals. In addition, there are not many outlier samples available; however, this does not pose any problem as we are only using OOD data during test analysis. The semantic patterns of inliers and outliers sometimes really look like each other. For instance, some LD images look exactly like a blood cell or cCry. The annotations of the classes are expected to be fully accurate with minimal error.

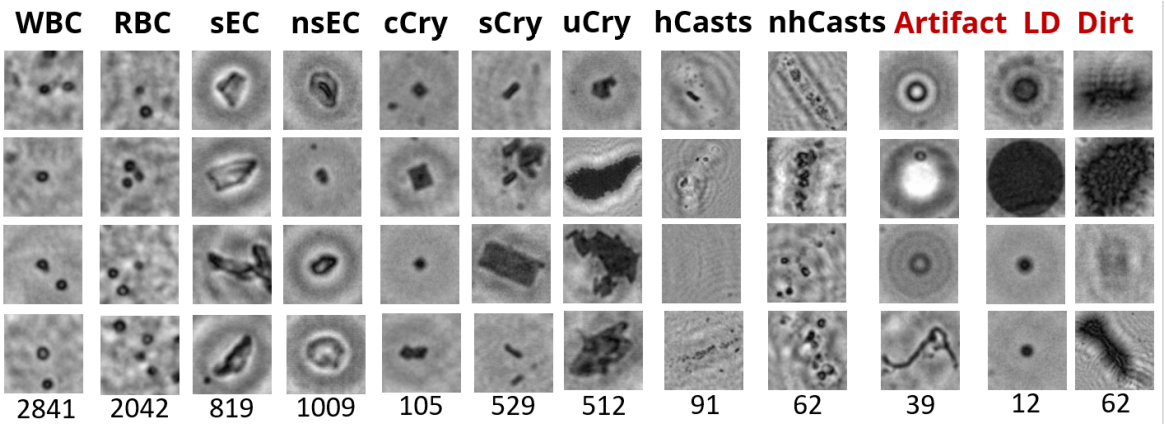


Figure 2.4: Inliers (black) and some outliers (red) are exemplified, numbers under each class represent the amount of available samples

3 Related Work

3.1 Generative Model based OOD Detection

The problem of detecting out-of-distribution examples in low-dimensional space such as density estimation on time-series of data, clustering analysis and nearest neighbor problem has been well-studied [Shi17]. The general idea behind reconstruction based methods is that every inlier sample should be reconstructed accurately using a basis function[Deh20] but anomalous data should get larger reconstruction errors. Nearest neighbor algorithm [al02], K-means [HW79], low-rank PCA [Jol11][al11a] can be used as reconstruction basis to establish the statistical distance between other components and density estimation approach is based on locating the low-density areas between two probabilistic distributions. In recent years, generative models have been used as alternative for the previous density estimation methods[al20a].

[al15a] claimed that PCA method can be extended by estimating the background covariance and the influence of background statistics can be mitigated to detect outliers during detection. [May14] showed that autoencoders are able to detect specific anomalies which PCA fails, presented that autoencoders learn the inliers much more properly and can detect anomalous inputs in lower dimensions. [al20c] showed that autoencoders are better in anomaly detection in time-series data than baseline density estimators. [Cho17] demonstrated Robust Deep Autoencoders (RDA). The idea is splitting the input dataset into two parts $X = L_D + S$ where L_D is effectively reconstructed data by the autoencoder and S is outliers or reconstructions with noise in the original data.

[Sun15] proposed to use variational autoencoder for anomaly detection purposes instead of using autoencoders or PCA. He presented VAEs as probabilistic graphical model (DPGM) and demonstrated that the probabilistic characteristics of variational autoencoder are incorporated by the reconstruction probability. The experiments show that probabilistic approach via VAEs outperforms the deterministic approach as VAE generalizes the characteristic of the trained data and therefore better for anomaly detection tasks.

[al04] proposed an alternative metric, *Structural Similarity Index Metric* (SSIM). SSIM is resembled to human visual system as our vision is highly adapted to extract the main

structural information from a visual input. To detect the distortion between two images SSIM can extract valuable information and be an alternative metric than common metrics like Mean Square Error (MSE) to evaluate reconstruction outputs of autoencoders and variational autoencoders. [Ber18] proposed using a perceptrual loss function based on SSIM metric instead of using a pixel independent error. By taking luminance, contrast and structural information of the reconstructed output, feasible comparison and significant performance can be achieved.

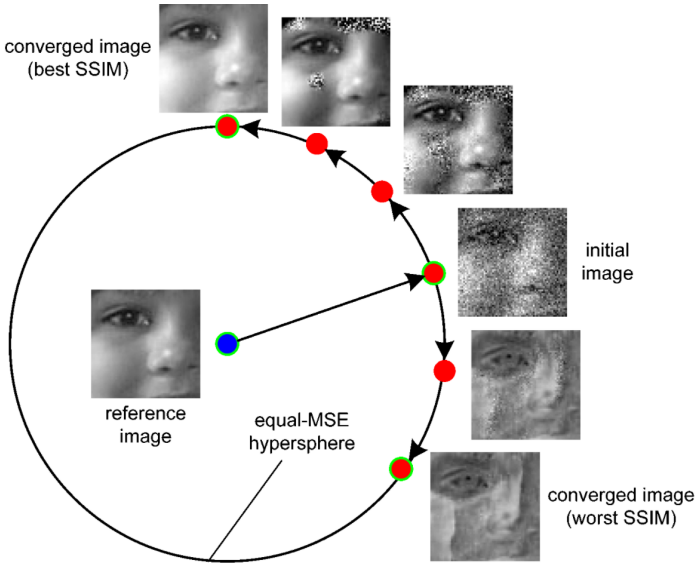


Figure 3.1: How SSIM score differs where MSE score continues returning the same error, image source[al]

[DB16] proposed deep perceptual similarity metric (DeepSiM) that computes the distances in the extracted feature space of deep neural network instead of computing distances in the image space. Using this metric as a loss function improves the generated image quality in terms of sharpness and resembling natural images.

[al20b] used a discrete probability model, called *VQ-VAE*, to estimate the latent space of an autoencdoer. With this probability model, outliers can be detected in the latent space and undesirable reconstructions can be avoided. VQ-VAE model is depicted below. Different than regular VAEs, VQ-VAE encodes the normal inputs to the latent space with a deterministic mapping like an autoencoder. Then the latent vector is collected and the probability model of the latent space is estimated by a deep autoregressive model during the training phase. After training the model, during testing, input samples are reconstructed two times, one via directly by the deterministic latent space and second by resampling the latent cvector via trained autoregressive model. Abnormal samples are detected by comparing two reconstructions and obtaining anomaly heat score map.

In order to apply detection algorithms in image space, suitable dimensionality reduction (DR) techniques are needed such as Principal Component Analysis (PCA), Autoencoders etc. DR techniques each yield a representation of the given image and such representations can be compared using suitable statistical drift detection methods. [al19b] applied various statistical drift detection methods like Maximum Mean Discrepancy (MMD) for multivariate test, Kolmogorov-Smirnov for multiple univariate test and Chi-Squared for categorical shift test between two sample distributions. These two sample distributions are obtained by

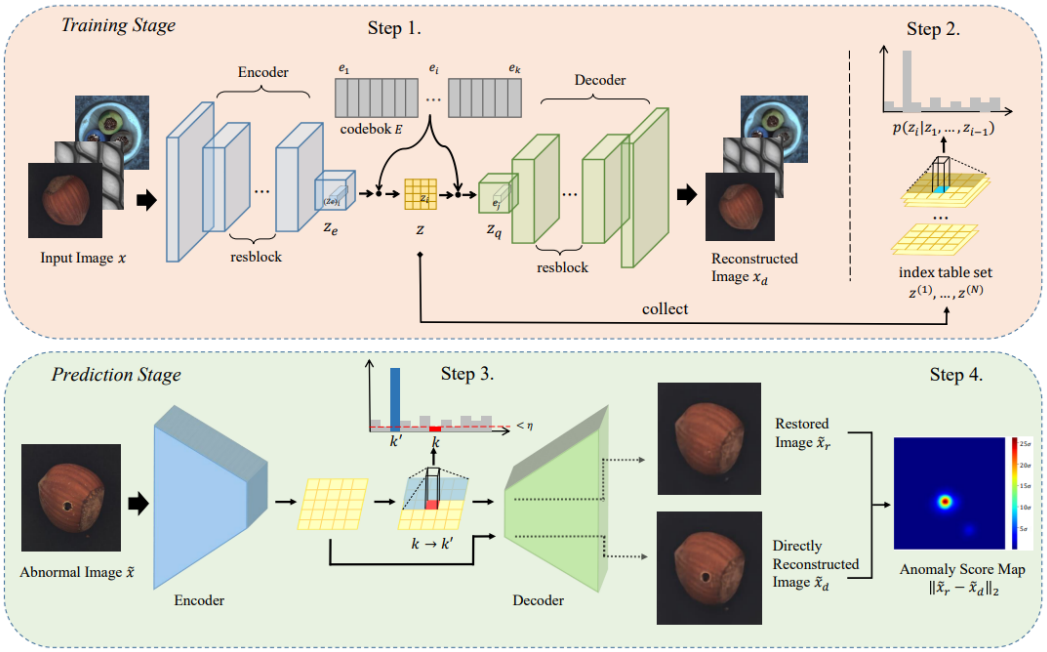


Figure 3.2: Pipeline of the VQ-VAE framework, image source[al20b]

applying various dimensionality reduction methods such as Sparse Random Projection (SRP), Autoencoder, PCA, label classifiers and domain classifiers to a base dataset and its shifted versions. Multiple shifts are added to the base dataset such as turning a fraction of samples into adversarial samples, removing some of the samples from a specific class (creating class imbalance), adding Gaussian noise, random rotations, using other datasets etc. Exhaustive amount of experiments with various combinations of statistical detection

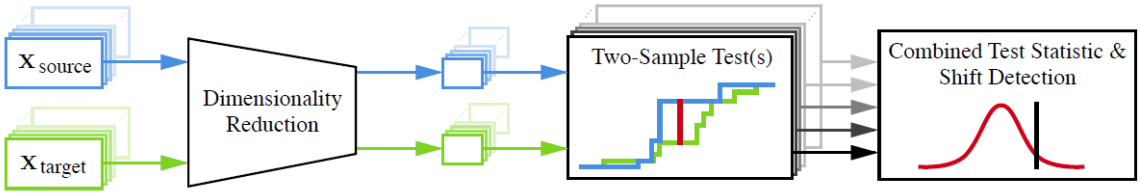


Figure 3.3: Rabanser's pipeline to detect shift between two dataset using DR methods. Image from [al19b]

methods yielded that untrained autoencoder on multivarite test and label classifiers with *black box shift detection (BBSD)* [al18b] method on univarite test performed the best.

On the other hand, drift detection methods can be unreliable when the drift is not continuous or obvious in high-dimensional space such as image analysis [Was06][LB15]. [al18g] showed that generative models might assign high likelihood to the OOD data and it has been later shown that by using alternative metrics, generative models can be an effective method to detect out-of-distribution samples. [al18c] presented Deep Autoencoding Gaussian Mixture Model (DAGMM) where the encoded features of the autoencoder is further fed into a Gaussian Mixture Model (GMM) to jointly optimize the parameters of the autoencoder and mixture model simultaneously. DAGMM is composed of two components, a compression network and an estimation network. Compression network can be a deep autoencoder or variational autoencoder. The squeezed latent vector (z_c) and reconstruction error features (z_r) is fed to the compression network. The compression network predicts the *energy* of the

inputs in the GMM structure. It is claimed[al18c] that this joint optimization balances the reconstruction outputs and regularize the latent space and outperforms state-of-the-art anomaly detection techniques.

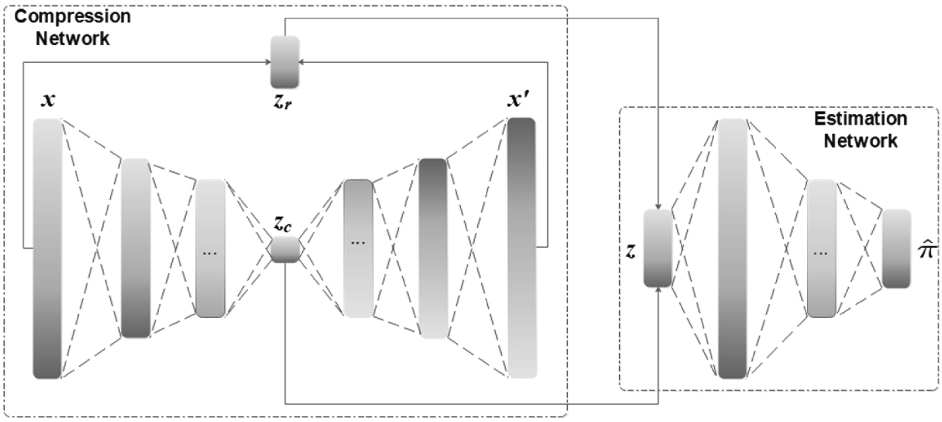


Figure 3.4: An overview of Deep Autoencoding Gaussian Mixture, [al18c]

It has been shown that deep generative models such as GLOW [al18f] and PixelCNN [al16c][al17c] sometimes assign higher likelihoods to OOD than IN inputs. [al18g] and [al18d] showed that GLOW model which is trained on CIFAR-10 as in-distribution dataset obtained higher likelihood for out-distribution dataset SVHN. [al18g][al18h][al18e] also showed similar failure cases with PixelCNN and PixelCNN++[al17d] model for OOD detection. [al19a] claimed that likelihood score is heavily influenced by the background statistics and proposed training two generative models where one is trained with the original inlier data and second with perturbed version of the inlier data. This way the second generative model is trained to capture the background statistics [Ltdb] since the semantic features are no longer available due to perturbations. By using the likelihood ratios between these two generative models, misleading background staitstics are corrected [al19a] and Likelihood Ratio (LLR) provides state-of-the-art performance.

Generative models require enlarged neural networks and due to the memory and performance constraints of the *fluidlab* device, generative models might not serve the optimal solution. *fluidlab* has already a trained classifier that classifies and segments in-distribution image patches. Therefore, such baseline OOD approaches that do not require training an extra neural network are more favorable.

3.2 Discriminative Model based OOD detection

In machine learning, kernel machines are used for pattern analysis. Kernel methods are *instance-based learners* where they learn particular weight per instance instead of learning fixed set of parameters based on features of training samples. The best known member of kernel machines is support-vector machine (SVM). [al00] uses one-class SVM to train a classifier to separate between normal and anomalous regions.

Neural network models outputs *logit* vectors in classification probems. Urine dataset is composed of inlier and outlier images and both inliers and outliers are composed of multi-classes. Classifier based OOD approaches should not be mixed with distinguishing among

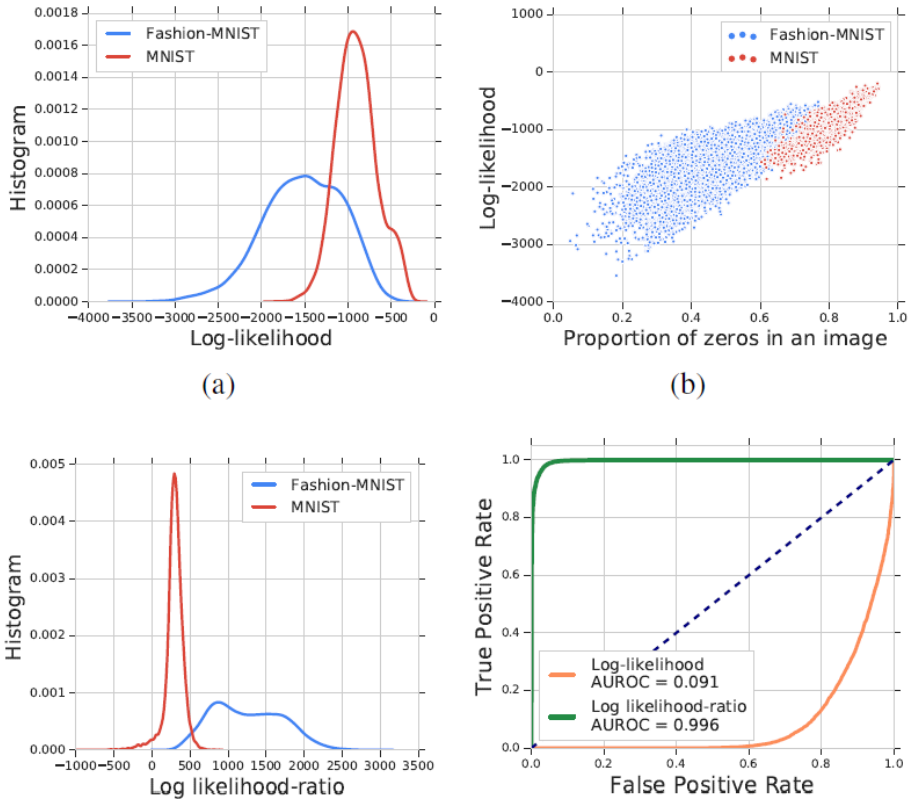
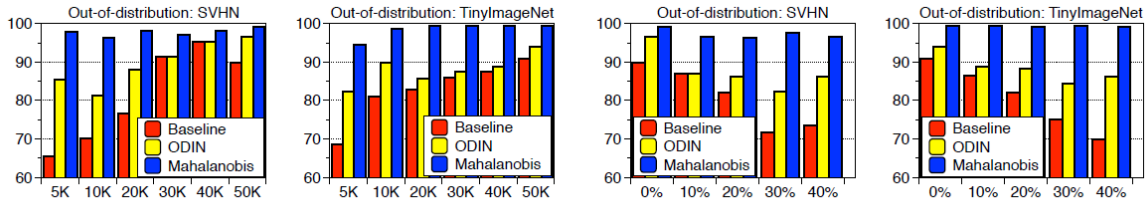


Figure 3.5: Likelihood of outliers is higher that in-distribution dataset (Fashion-MNIST) but likelihood-ratio is higher for in-distribution samples than OOD samples(MNIST), image from [al19a]

inlier classes or outlier classes. [Anh14] reported the initial evidences that OOD inputs make discriminative neural networks produce overconfident predictions. [Dan16] proposed Maximum Softmax Probability (MSP) approach, demonstrated that a maximum threshold value can be assigned to the softmax outputs of the DNN where the values passing that threshold will be detected as outliers.

Following the baseline softmax OOD detection method, [Shi17] proposed ODIN (calibrated MSP), claimed that the softmax outputs of the DNN have to be calibrated. The baseline MSP approach is calibrated by adding temperature scaling and input perturbations to the test inputs.

[al18a] obtained class conditional Gaussian distributions. It has been shown [al18a] that Mahalanobis OOD detector outperforms ODIN and Baseline MSP OOD detectors consistently on various vision datasets CIFAR [KH09], SVHN [al11b], ImageNet [al09] and LSUN [al15b]. Moreover, Mahalanobis OOD achieves higher AUROC(%) when the number of training data is scarce or the network is trained with randomly labeled data. This approach is also suitable to class-incremental learning without further training the neural network. Class-incremental learning also enables the detector to classify out-of-distribution samples into classes as whenever a new out-of-distribution samples are detected, it gets distanced closer to its own looking-like (same class) distributions. Therefore, Mahalanobis distance-based OOD detection algorithm works robust in harsh conditions and suitable to use in adversarial cases where ODIN or MSP catastrophically fail [al20a].



(a) Small number of training data (b) Training data with random labels

Figure 3.6: Comparison of AUROC (%) between classifier-based OOD detectors where the x-axis represents the number of training data. (a) small number of training data, (b) Random label is assigned to training data [al18a]

[al18e] introduced Outlier Exposure (OE) method where a DNN is taught to detect between in-distribution samples and small number of out-of-distribution samples. Then the network is tested with an OOD data disjoint from previously trained OOD samples. It is observed that network generalize on unseen OOD distributions so performs better with this semi-supervised learning method.

4 Methods and Implementation

The aim of this thesis is differentiating between inliers and outliers. The main challenge while achieving this goal is that potential algorithm should be unsupervised, meaning that outliers will only be seen during testing.

The general setup of an OOD detection algorithm includes urine in-distribution dataset, a dimensionality reduction model and an OOD algorithm. In this section, various unsupervised OOD detection methods that are applicable to urine dataset will be elaborated exhaustively until a robust method is achieved.

4.1 Data Preparation

All of the OOD detection algorithms shown in this paper include an unsupervised neural network architecture, meaning that only in-distribution images are used as the reference dataset during training and inference. These architectures are trained by the same group of inlier images called *Train Dataset*.

4.1.1 Train Dataset

As we have seen in the Background part, inlier classes are unbalanced regarding the number of available image patches per class are not equal. In order to avoid overfitting and ensure higher balanced accuracy, balanced detection rate per class. By this way, we give equal priority and opportunity to each class in structuring the weight distributions of the deep learning model.

Augmentations

Usually it is difficult to find enough training data for deep learning models since collecting, labelling and preprocessing the image take a lot of time and effort. Sometimes it is even impossible to collect particular class images as the case in urine dataset. Simple urine image

analysis results in obtaining tons of white or blood cells but some class members like casts are not easily found in the sample test.

This problem can be solved or alleviated via augmentation. Augmentation of the data is simply adding features or transforming some features of the data to create more data. The microscopic image patches can be reproduced easily by applying rotation, flip transformations, scale and crop augmentations. Such transformations like crop or scale will shift the semantic pattern of the image away from the center. Image patches are patched in squares via centering the semantic pattern. Such augmentations might cause distortion in the train dataset and influence the learning algorithm. We aim to add augmentations without manipulating the overall data rather just balancing the samples per class. Therefore, transformations like rotation and oversampling is applicable for urine dataset. By oversampling and adding rotations to the classes with low number of samples we extended the amount of available total train samples to 312987 from 23918.

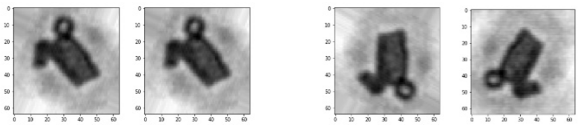


Figure 4.1: Example of a rotated and oversampled crystal image

4.1.2 Test Dataset

The test dataset is used in order to evaluate and compare the performances of different OOD methods. The test dataset is composed of 636 outlier and 636 unseen inlier image patches. Since outliers are not often observed during image analysis, the availability of outlier images often lack in numbers. Therefore, the number of 636 has been chosen based on the available number of out-of-distribution images.

4.2 Unsupervised Anomaly Detection Basics

A variety of ML approaches have become increasingly popular for anomaly detection. Some basic approaches to identify the *normal* area or plane in the dimensional space in which the data is spread out and the anomalous samples that lies outside of the marks filled with inliers can be detected as outliers[al02]. Let us reduce the dimensions of 32x32 sized inlier and outlier samples to 2 components and observe whether we can differentiate them in 2D space.

4.2.1 Principal Component Analysis (PCA)

PCA is a simple linear transformation on the input space to the maximum variation [Mun] and the projected data into orthogonal dimensions provide zero correlation. The idea of principal component analysis (PCA) is reducing the dimensionality of a data consisting of a large number of interrelated variables while retaining most of the variation present in the

data set by using eigenvalues. [Mat18] showed that PCA method can be applied to images by observing eigenvectors as eigenimages. This way we might achieve informative low dim representations of inliers per class. The mathematical process of applying PCA can be briefly summarized as we take *d+1* dimensions of array for every image where *d* is the flattened version of 32x32 image matrix and the last term represents the label of the inlier class. Then the mean of every class dimensions and then covariance matrix of whole dataset is calculated. Corresponding eigenvecors and the corresponding eigenvalues are computed from the covariance matrix since the direction of an eigenvector remains unchanged when a linear transformation is applied. Eigenvectors with the lowest eigenvalues provide the least information; therefore, only 2 eigenvectors with the highest eigenvalue is chosen and the image is projected into these 2 components resulting in 2 principle components. In the below, you can observe the two dimensional PCA of inlier and some outlier images in the normal region.

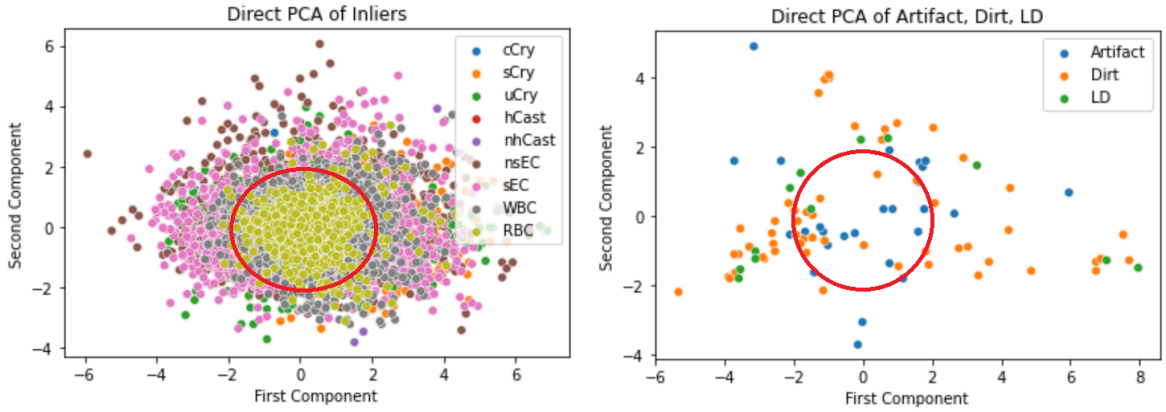


Figure 4.2: Inlier and outlier image analysis with 2 principle components. The red circle represents the same region between two plots.

As you can see, PCA of inliers and some outliers in 2 dimensional space does not really provide any possible threshold to detect outliers. Inliers of the same class are hardly grouped into similar regions and outliers cannot be detected in the normal region. There are many possible reasons why PCA fails. One of them is that PCA ignores the semantic relation between adjecent pixels as image matrix are considered as one long array and feature independent orthogonal projections may not yield distinguishable 2D representations.

4.2.2 Autoencoder

Let us now discuss Autoencoders, another dimensionality reduction method using neural networks. Autoencoders are composed of encoder and decoder scheme composed of neural networks. They learn the best composition of weights through iterative optimization process. The encoder part take the input image, downsamples it through a bottleneck and decodes the downsampled features into 32x32 reconstruction version of the input image. Decoded output is compared with the input through a reconstruction error metric and the error is backpropagated through the architecture to update the weights. Just like PCA, autoencoders look for the best subspace to project the data but they do not have just one best composition as PCA, they can learn several optimal subspaces by gradient descent. By choosing a bottleneck (encoding dim) as 2, we can obtain two dimensional distributions of

inlier and outlier samples similar to PCA approach. The literature also supports that[May14] autoencoders are able to detect specific anomalies in which PCA fails in lower dimensions.

Encoder and decoder networks are composed of 3 hidden layers and 2 dimensional bottleneck. By using multiple layers, it is assured that autoencoder will follow a nonlinear approach at finding best subspaces. In additon, a deep autoencoder is prone to overfitting; therefore, regularization and drop out methods are used to avoid overfitting. However, it is observed that it was impossible to obtain good reconstructions with a bottleneck size of 2. However, our aim was never to obtain good reconstructions for now. The aim is to observe whether two dimensional reduced inlier and outlier inputs are differentiable or not.

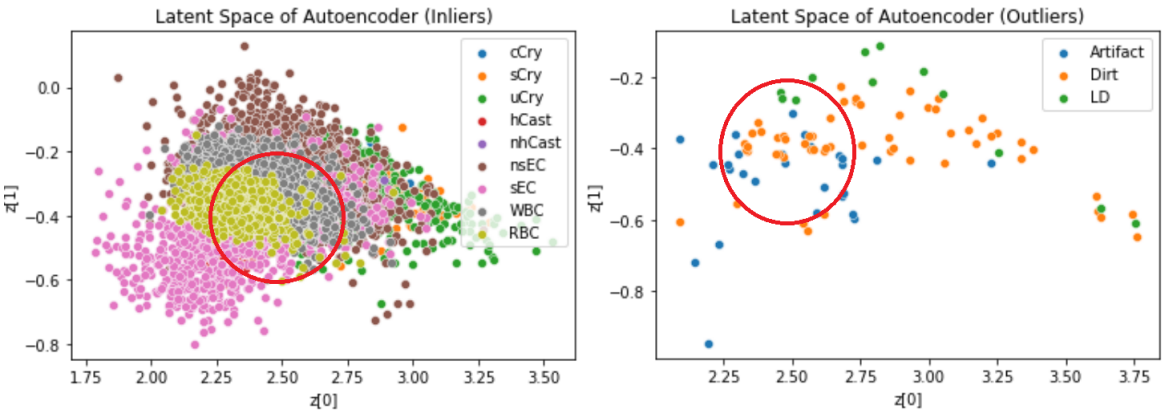


Figure 4.3: Encoded inlier and outlier images in 2 values. The red circle represents the proportional region between two plots.

From the figure 4.3 we can observe that two dimensional encodings of inliers and outliers also are not seperated in 2D space and a possible threshold to detect outliers cannot be drawn. Encodings are grouped together on top as it was the case in PCA. The most important purpose of dimensionality reduction methods for anomaly detection is decreasing dimensions while keeping encodings **exploitable and interpretable**. Using a low dimensional encodings forces the autoencoder to generalize on the inputs and both the reconstruction output and encoded space are generalized, resulting anomaly detection impossible to apply.

4.2.3 Reconstruction Methods for Anomaly Detection

Previously, we could not obtain useful encoded information and good reconstructions on decoded outputs with a low encoding dimension. However, choosing a suitable good encoding dimension is very tricky. If a large encoding dimension is used, the model simply overfits and the autoencoder achieves very good reconstruction outputs even with the anomaly data. Another popular approach which is widely observed in deep learning literature[Sun15][al15a][al20c] is based on *reconstruction methods*. The main idea is based on an assumption that if a model is trained to encode and reconstruct a given data, it should fail to reconstruct on unseen or anomalous data. Some argues [al18c] that this approach is not very feasible in microscopic data; however, it is shown that by using alternative metrics[al19a] a good comparison can be achieved.

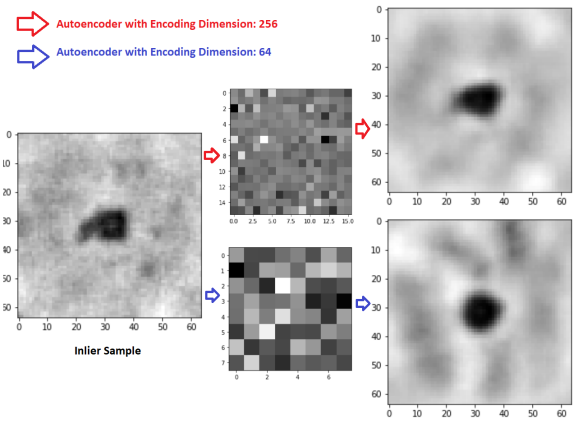


Figure 4.4: Encoding and reconstruction of a sample inlier by two different autoencoders

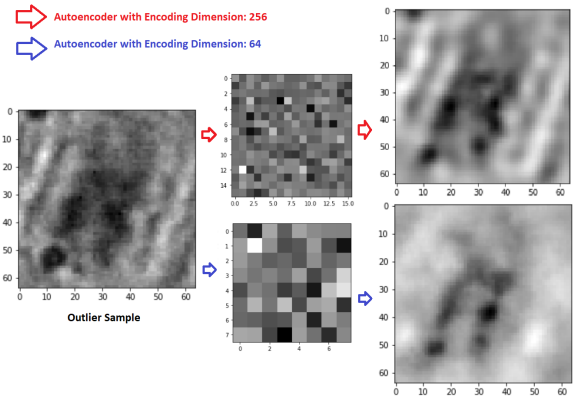


Figure 4.5: Encoding and reconstruction of a sample outlier by two different autoencoders

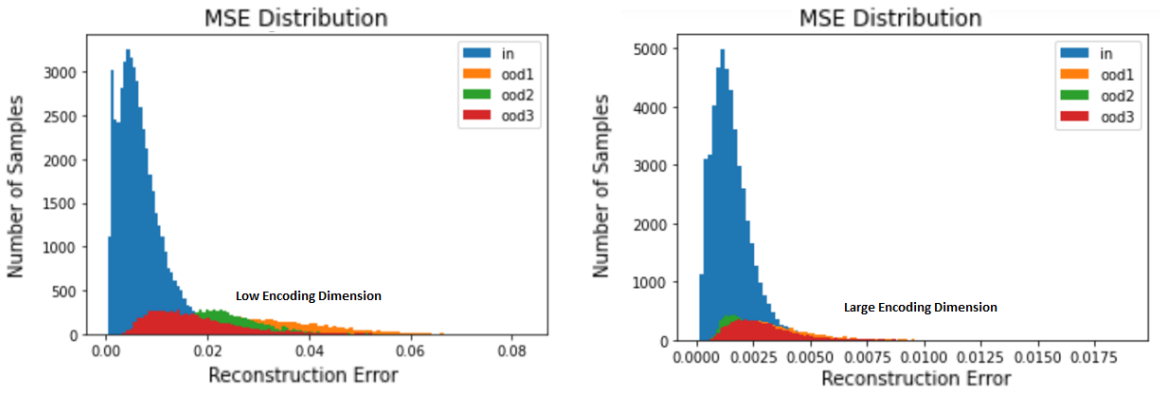


Figure 4.6: Importance of choosing low encoding dimension for anomaly detection via error in reconstructed outputs

In order to observe the reconstruction quality *Mean Square Error (MSE)* and *Structural Similarity Index Measure (SSIM)*[al04] metrics are used. MSE metric is one of the most common metric to evaluate the quality of the reconstruction outputs of generative models. MSE measures the average of squares of errors and the errors are measured by taking the differences between particular input pixels and reconstructed output pixels. MSE can be defined as $MSE = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ where y_i, $\hat{y}_i$ and N are input pixel, output pixel and total number of pixels respectively.

SSIM

MSE metric particularly may not be a convenient metric for urine images as most of the background images are noise. Therefore, another metric that penalizes especially the structural dissimilarity is needed. SSIM metric extracts 3 key features from an image: *Structure, Contrast and Luminance*. Luminance is basically the average of all pixel values, $\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$. The contrast is measured by taking the standard deviation of all pixel values, $\sigma_y = (\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2)^{\frac{1}{2}}$. Finally, the structure is calculated by $(y - \mu_y)/\sigma_y$. In order to compare an input image x and output image y *comparison functions* are used again for each feature. Luminance comparison function is defined by a function $l(x, y) = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C1}$ where C_1 is a constant to avoid the cases where the denominator becomes 0. C_1 is also calculated as $C_1 = (K_1 L)^2$ where L is the range for pixel values (255 or 1 for normalized case) and K_1 is a normal constant. Contrast comparison function is defined again by a function $c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{{\sigma_x}^2 + {\sigma_y}^2 + C_2}$ where $C_2 = (K_2 L)^2$ and K_2 is again a normal constant. Structure comparison function is again defined by a function $s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y 9C_3}$ where $\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$. And finally, SSIM score is computed by

$$SSIM(x, y) = [l(x, y)]^{\alpha}[c(x, y)]^{\beta}[s(x, y)]^{\gamma}$$

where α, β, γ can be chosen based on the importance of the metric. During experiments, we used $\alpha, \beta = 0$ and $\gamma = 2$ to increase the contribution of the structure comparison function.

Limitations of Autoencoders

Two autoencoders which are architecturally identical except having different encoding dimensions have been trained using MSE loss function (detailed information about the architectures and results are provided in the results section). The autoencoder with a large encoding dimension (256) obtained better reconstructions with both inlier and outlier samples as expected. The second autoencoder with 64 encoding dimensions obtained proportionally worse decodings for both inliers and outliers as you can see in figure 4.4 and 4.5. It is also realized that encoded image space gets noiser as the encoding dimension increases. After many attempts of experimenting different hyperparameter such as encoding dimension, number of layers, adding regularization and drop out layers, autoencoders could not achieve higher score with both SSIM and MSE metrics for inlier samples and lower score for outlier samples. The autoencoder is trained to encode and decode given training data with a few MSE loss as possible no matter how the encoded dimension is organized. Therefore, it is natural that autoencoders lead to either severe overfitting or underfitting for both inlier and outlier samples unless we explicitly *regularize* the latent space.

4.2.4 Variational Autoencoder

[al14] first suggested that probabilistic variational autoencoders (VAEs) can be a better approach for reconstruction based anomaly detection problem. VAEs similarly learn an encoding-decoding scheme like traditional AEs. However, instead of learning how to generate a latent vetor, VAEs learn how to generate mean (μ) and variance (σ) vectors

that can represent a normal distribution from which the latent vector z can be sampled by $z \sim \mathcal{N}(\mu, \sigma^2)$. VAE encodes inputs as distribution rather than points and the constrained Gaussian distributions returned by the encoder prioritize the latent space organization. This way VAE learns a general representation of training images and this enables VAEs to generate sample images from its latent space without any input. We are not interested in content generation but probabilistic property enables[Sun15] to detect outliers better than simple AE. The trained VAE will learn the μ and σ parameters based on inlier samples and it will fail more at reconstructing outlier samples [al14].

Formulation and Updated Loss Function

Given x as the input image and multivariate latent vector z, if we assume that $p(z)$ is a standard Gaussian distribution and $p(x|z)$ is a Gaussian distribution as well whose mean and variance is defined by the VAE. Then, $p(z|x)$ can be obtained via Bayesian inference problem:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Unfortunately, computation of $p(x)$ is expensive, in order to further speed the computation $p(z|x) \approx q(z|x)$ is assumed. Then the conditional likelihood $p(z|x)$ is calculated by the decoder of VAE and $q(z|x)$ is computed by the encoder of VAE.

Updated Loss Function

In VAE, the aim is jointly minimize the reconstruction error while approximating $q(z|x)$ to $p(z|x)$. To compute the loss between two distributions reverse Kullback-Leibler divergence $D_{KL}(q(z|x)||p(z|x)$ is used. Previously in AE, MSE loss function was used to tame the reconstruction output error. In MSE, loss is computed by adding two loss terms namely *reconstruction loss (MSE)* and *KL loss*. The final loss is defined as $TotalLoss = ReconLoss + \beta * KL_Loss$[al16b] where $\beta = 10$ to further increase the influence of Kl loss.

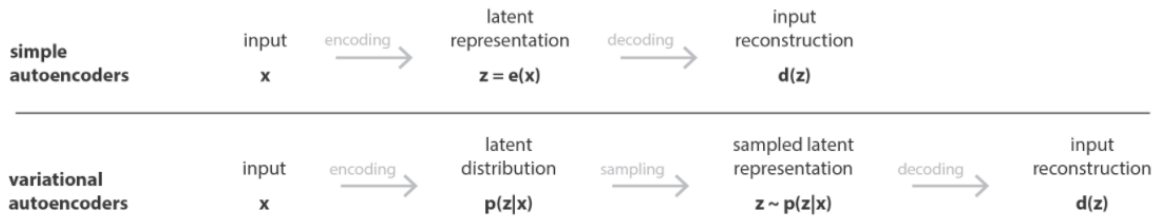


Figure 4.7: Difference between Traditional (Deterministic) Autoencoder and Variational (Probabilistic) Autoencoder

4.2.5 Training AE&VAE

Both AE and VAE architectures accept (32x32x1) image array as input and composed of 3 hidden layers with convolutional neural networks with 3x3 kernel with stride equals to 2. The final flattened layer of AE's encoder returns a latent array dimension of 16x16. VAE's

encoder includes the sampling function and returns three 16x16 arrays namely latent space, mean and standard deviation. The decoder network follows the same inverse structure to reconstruct the (32x32x1) shape output. The final autoencoder is composed of 527297 trainable params with a total size of 2.1 Mbs (encoder and decoder networks both weight 1.034 Mbs) and VAE with a total size of 2.9 Mbs.

Both networks are trained for 500 epochs, with a learning rate of 1e-4. One fifth of the train images are used as validation data. Mean squared error (MSE) loss function is used during training.

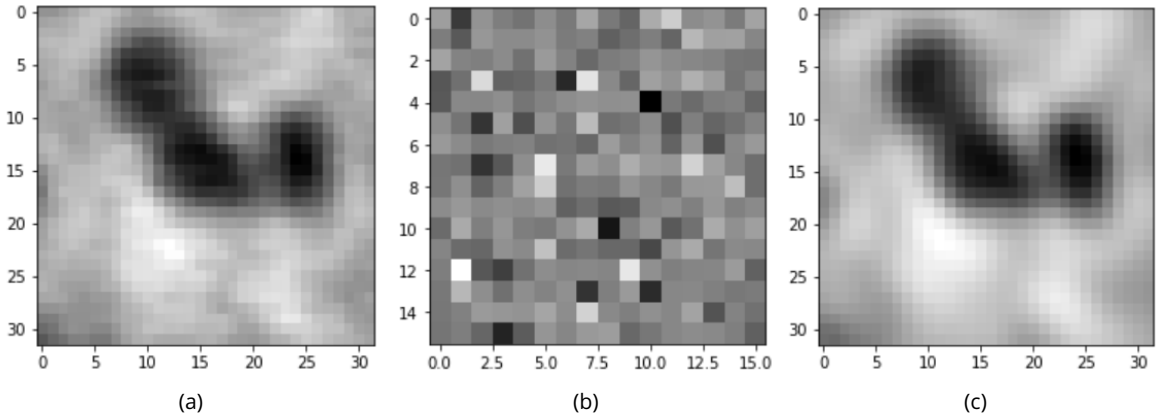


Figure 4.8: Example of an input image, encoded version and decoded reconstruction image

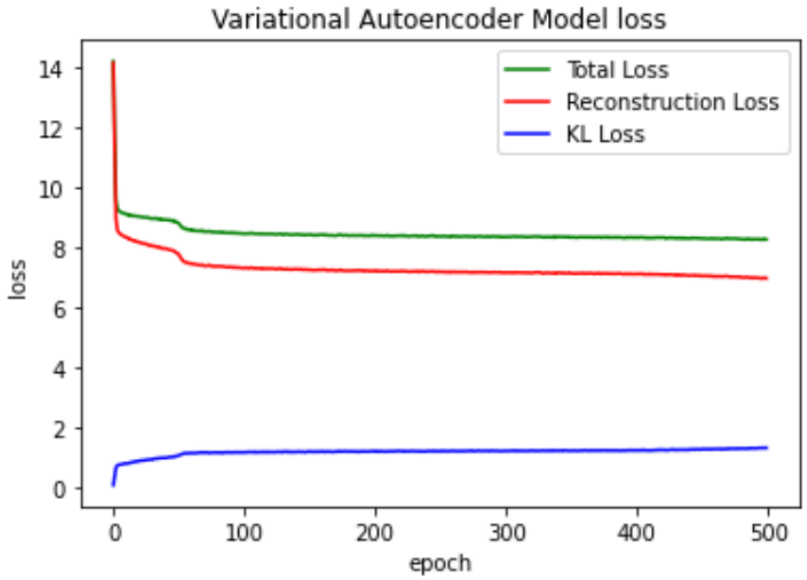


Figure 4.9: VAE Model Loss

4.2.6 Likelihood Ratios Metric with an Autoregressive Model

An advanced autoregressive generative model is also needed to be trained and applied for reconstruction based OOD detection. The VAE described earlier was a very straightforward neural network model and it might fail at fitting the in-distribution dataset. Autoregressive model can be defined as *a model that predicts future based on past behaviour*. Here

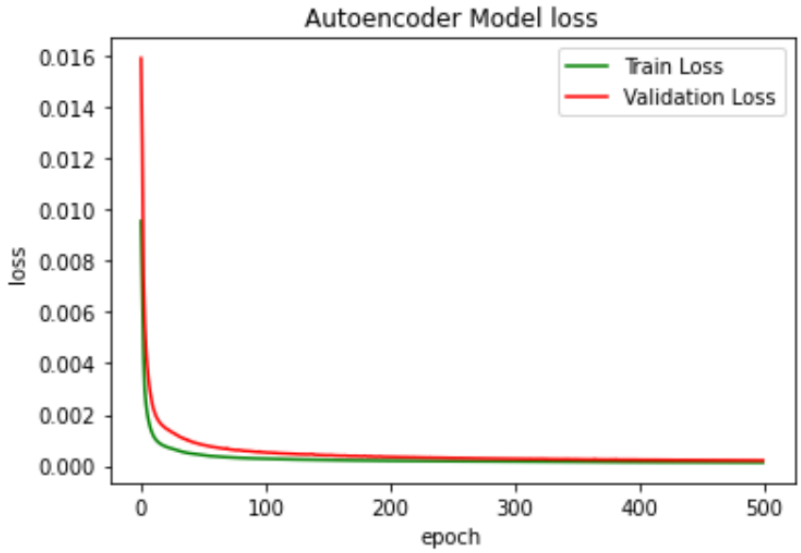


Figure 4.10: AE Model Loss

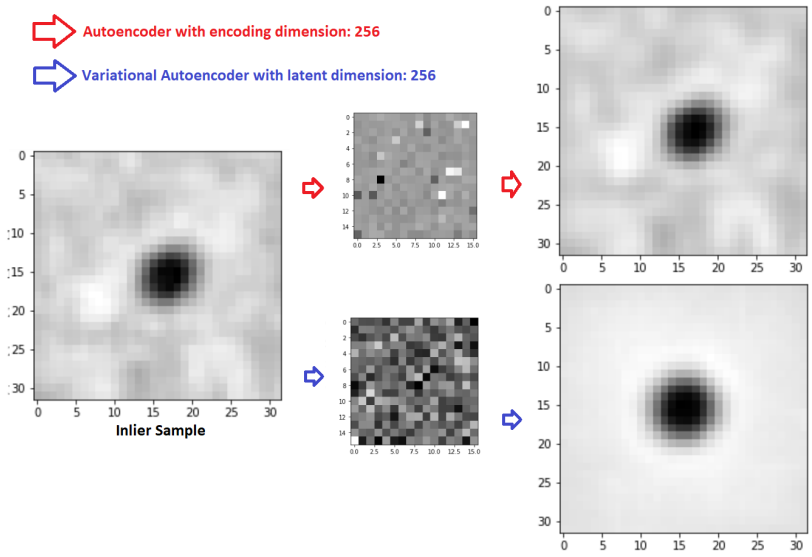


Figure 4.11: Encoding and reconstruction of a sample inlier by Autoencoder and Variational Autoencoder

autoregressive means "regressed against itself automatically". The main contribution of autoregressive models is that they provide a way to calculate the likelihood due to *conditional independence* between pixels. The autoregressive networks scan the image one pixel at a time and predicts the conditional probabilities over the possible pixels and these possible values are shared across all other pixels. The objective is assigning likelihood probability $p(x)$ to every pixel of nxn sized images:

$$p(x) = \prod_{i=1}^{n^2} p(x_i|x_1, ..., x_{i-1})$$

In addition to conditioning on whole pixels, PixelCNN++ apply downsampling to compute long range dependencies between CNN blocks. Residual blocks are used to skip the layer information to the next layers. We have seen that downsampling reduces the input size significantly and improves the receptive field. Skip connections are used in the middle layers

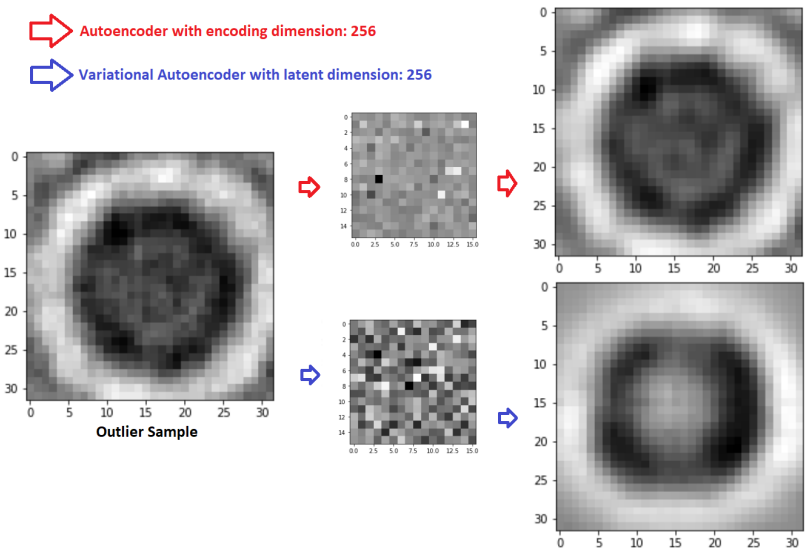


Figure 4.12: Encoding and reconstruction of a sample outlier by Autoencoder and Variational Autoencoder

to apprehend the convolutional connections. Droput is also used since the model is very powerful and likely to get overfit. Overall, the PixelCNN++ is a high level network that we can apply to our urine dataset; however, if we consider the inference time and amount of memory and computation it will use, it is likely that such a model will not be feasible to use in *fluidlab* for anomaly detection in real time detection purposes. However, it might provide useful insights in *datalab*.

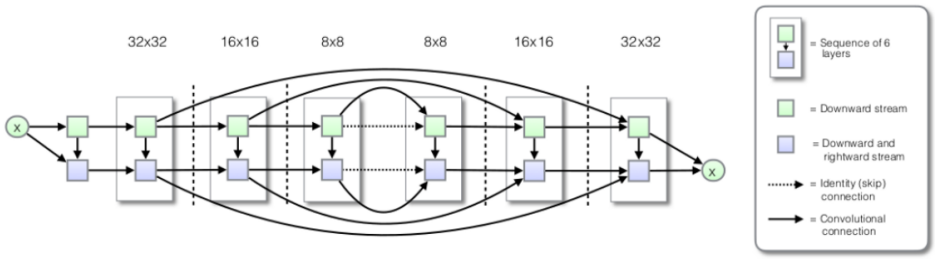


Figure 4.13: Architecture of PixelCNN++

In addition to a new model, [al19a] provided a high level OOD metric that is claimed to solve higher likelihood problem[al18g][al18h][al18e] for OOD inputs. Idea of the metric assumes two components for an input image *semantic component* and *background component*. Background component is mostly characterized by population level background statistics[al19a] and the semantic component is the specific structural pattern of the image. If one generative model is trained on the original data and a second generative model is trained on perturbed version of the original data, the second model will capture the background information only since the semantic structures will no longer be present due to perturbations. A simple sample from urine dataset includes a noisy background and a semantic structure often as a cell shape. Usually the inliers, especially blood cell images, are mostly composed of background statistics compared to other outlier images like Dirt or Artifact where they are mostly composed of unique semantic structures. Therefore, using a usual reconstruction error metric like MSE or SSIM might assign higher error for inlier samples than it's supposed to be. Therefore, eliminating the background statistics by Likelihood Ratios (LLR) metric

can uncover important insights for our OOD detection purposes. Given an input x can be factored as $x = \{x_B, x_S\}$ if the background and semantic components are assumed to be generated independently, the likelihood can be decomposed as

$$p(x) = p(x_B)p(x_S)$$

If we assume $p_\theta(.)$ and $p_{\theta_0}(.)$ as the likelihood of model trained with in-data and perturbed in-data respectively, the likelihood ratio (LLR) can be defined as

$$LLR(x) = log\frac{p_\theta(x)}{p_{\theta_0}(x)} = log\frac{p_\theta(x_B)p_\theta(x_S)}{p_{\theta_0}(x_B)p_{\theta_0}(x_S)}$$

By assuming[al19a] both generative models capture the background information equally, then $p_\theta(x_B) \approx p_{\theta_0}(x_B)$ then $LLR(x) \approx logp_\theta(x_S) - logp_{\theta_0}(x_S)$

4.3 Classifier based OOD Detection Approaches

4.3.1 Maximum Softmax Probability (MSP)

[Dan16] demonstrated a softmax prediction probability as a baseline approach for the OOD detection problem. Assume a pretrained classifier network f that classifies inlier classes. Given an input x the maximum softmax score of the classifier is computed by taking the maximum output of the softmax layer. Then, confidence score $S(x; f) = max_i F_i(x)$ is calculated per images and the images that have higher threshold γ detected as outliers. A good threshold value can be determined manually. It is claimed that confident predictions which have greater maximum softmax probability tend to be outliers [Dan16] rather than in-distribution sample.

$$F(x) = \frac{e^{f_i(x)}}{\sum_{j=1} e^{f_j(x)}}$$

The detector G_{MSP} can be defined as below:

$$G_{MSP}(x; \gamma, f) = \begin{cases} 0 & \text{if } S(x; f) \leq \gamma \\ 1 & \text{if } S(x; f) > \gamma \end{cases}$$

4.3.2 Calibrated MSP (ODIN)

Calibration Problem

[Shi17] claimed that MSP confidence scores might be overconfident and lack calibration. Confidence scores should match the true expected likelihood of the given test case. If a given neural network outputs %87 accuracy then it is expected that 87 of the test cases are inlier samples and 13 of them are outliers per 100 samples. If this is the case, it can be said that the network is *perfectly calibrated*. Let f be the neural network with $f(x) = (y, p)$ where

y is the prediction output and p is the probability of correctness. Then perfect calibration can be defines as,

$$P(\hat{Y} = Y|\hat{P} = p) = p, \forall p \in [0, 1]$$

Perfect calibration is almost impossible to achieve; however, it should be pursued. Networks without calibration tend to output higher confidence scores [al17a] and they require calibration to have trustworthy probability outputs. In the below, you can see a reliability diagram that represents the effect of model calibration. Examples that have the same expected confidence are grouped in the same blue bin and their accuracy is calculated by the mean accuracy of the bin. *Expected Calibration Error (ECE)* is used to calculate the calibration error where M, B_m, acc and $conf$ represents the number of bins, m^{th} bin, accuracy and confidence of the selected bin respectively. One calibration method, *Temperature Scaling*, is an extension of *Platt Calibration* which is transforming classification model outputs into a probability distribution of classes [Pla]. Temperature scaling is determined by a single scalar parameter T.

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$

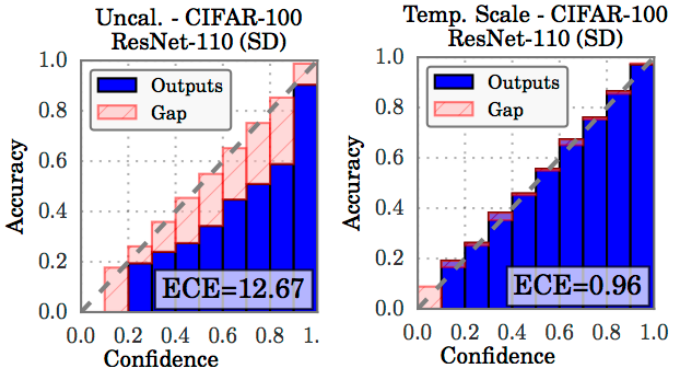


Figure 4.14: Sample Reliability Diagram, Left uncalibrated Resnet110 on Cifar100, Right calibrated version with Temperature Scaling calibration method

[Shi17] proposed ODIN for detecting out-of-distribution samples. The detector uses a pretrained network f as MSP baseline approach but additionally uses temperature scaling to calibrate the prediction results. Therefore, the method is also referred as *calibrated confidence score* $S(x; T, f)$. In addition to the temperature scaling, ODIN also applies noise perturbation to the input images. These small perturbations decrease the softmax score [Shi17] and force the detector to make wrong predictions. It is observed that [Shi17] perturbations work better at differentiating in-distribution images than out-of-distribution images. In other words, the softmax score on any given input is decreased more with inlier images than outlier images. Therefore, by combining calibration techniques like temperature scaling and perturbation, a softened softmax function is obtained.

ODIN approach shows almost better results in any combinations of in and out-of-distribution datasets compared to the baseline MSP approach [Shi17][al20a][al19a]. ODIN does not require any change to the pre-trained classifier as the MSP approach.

Input Preprocessing

The test dataset is perturbed by gaussian and saltpepper noise. This input preprocessing method is inspired from [Ian15], claimed that adding small perturbations to the inputs decreases the overall softmax scores and force the classifier to make wrong prediction [Shi17]. It has been observed [Shi17] that perturbation have stronger effect on inliers, meaning that the confidence score gap between inliers and outliers increases even more. The calibrated confidence score is computed by

$$S(x;T,f) = max_i \frac{e^{\frac{f_i(x)}{T}}}{\sum_{j=1} e^{\frac{f_j(x)}{T}}}$$

$\hat{x} = x - \eta.sign(-\nabla_x log S(x;T,f))$ where the perturbation magnitude η can be optimized based on the network and dataset. The temperature scaling value T and perturbation noise η is chosen from a set of values by re-running experiments in a grid optimization format where all combination of values are tried. Optimization spaces are defined as $T = [1, 10, 100, 1000]$ and $\eta = [0.001, 0.01, 0.1, 1, 10]$. The final structure is obtained with $T = 100$ and $\eta = 0.01$ values. The final ODIN detector (also known as calibrated maximum softmax probability) (G_{ODIN}) is obtained by

$$G_{ODIN}(x;\gamma,T,\eta,f) = \begin{cases} 0 & \text{if } S(\hat{x};T,f) \leq \gamma \\ 1 & \text{if } S(\hat{x};T,f) > \gamma \end{cases}$$

However, the parameters of T and η are needed to be well-tuned. This additional task requires additional computation power but also might not be possible if the target out-of-distribution dataset does not exist in large scale, as it is the case in this thesis. Urine dataset has samples of OOD data but they lack in numbers and only sufficient for the test runs. The network can be still tuned by a random noise such as Gaussian or uniform distribution images. Such an approach may yield unexpected results when the OOD distribtuion is composed of completely different image samples but this method is the only way to make ODIN approach applicable to the Urine dataset.

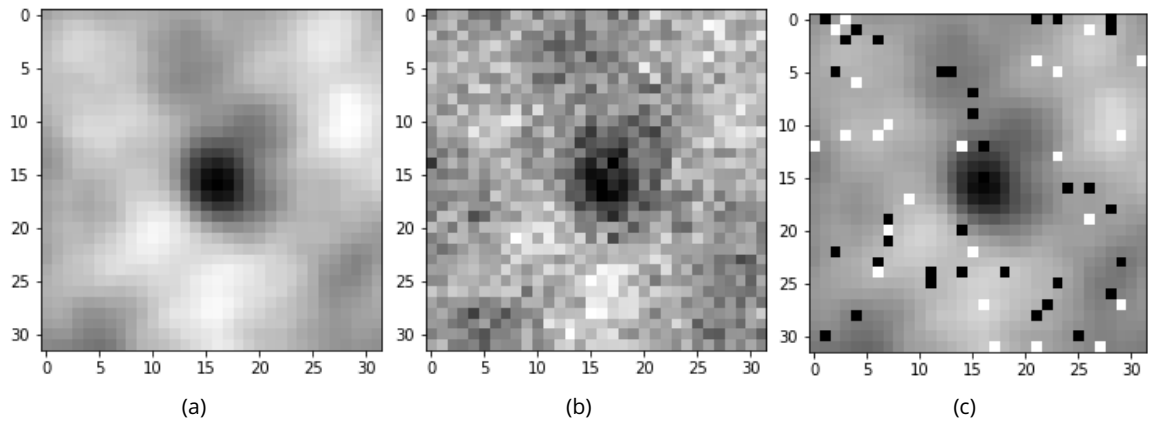


Figure 4.15: Perturbation example, (a) the original image, (b) gaussian perturbation added, (c) salt&pepper perturbation added

4.3.3 Mahalanobis distance-based OOD Detection

Mahalanobis distance-based OOD detection proposed by [al18a] is a very effective and simple method that promotes the use of theoretical connection between Gaussian distribution and softmax classifier. The proposed idea is that output weights of the penultimate layer of a trained DNN can be represented as class-conditional Gaussian distribution with a tied covariance matrix [al18a]. The parameters of these class-conditional Gaussian distribion are defined as the empirical mean and covariance of training samples [al20a]. Then for a given input sample x they calculate a confidence score by calculating the Mahalanobis distance between the Gaussian distribution output and covariance.

Mahalanobis distance based OOD detection method is claimed to be working robust in harsh conditions where the dataset is noisy, lack in numbers or adversarial[al18a]. It is observed that Mahalanobis distance-based OOD detection is robust as its been told and it is so far the best method which is applicable for urine dataset OOD analysis.

Mahalanobis Distance

Let us start by understanding the concept of Mahalanobis distance. Mahalanobis distance is the distance d between a point P and a distribution D [McL99] introduced by Prof. P. C. Mahalanobis in 1936. The core difference between Mahalanobis distance and Euclidean distance is that Euclidean distance does not consider the distances or relations between other data points. However, Mahalanobis distance uses an additional term, the covariance matrix (Σ) that holds the covariance of other data points in 2D matrix form. This makes Mahalanobis distance an effective equivalent of Euclidean distance for multivariate data. If we assume a point P that can be a multidimensional point or a list of observations defined as $\overrightarrow{P} = (P_1, P_2, P_3, ..., P_N)$ where N is the number of dimensions (space) or observations. Then the distance of P to another set of points that have a mean $\overrightarrow{\mu} = (\mu_1, \mu_2, \mu_3, ..., \mu_N)$ can be defined as $D^2 = (\overrightarrow{P} - \overrightarrow{\mu})^T \Sigma^{-1} (\overrightarrow{P} - \overrightarrow{\mu})$

Mahalanobis distance can be used to measure the dissimilarity between two points $\overrightarrow{x}$ and $\overrightarrow{y}$ of the same distribution. Additionally, these two points can be assumed as two data distributions such as random variables, mean of multiple data points etc. Then the general formula can be redefined as below where μ_1 is the first data distribution and μ_2 is the second. The covariance matrix for these two distributions are computed accordingly.

$$D^2 = (\overrightarrow{\mu_1} - \overrightarrow{\mu_2})^T \Sigma^{-1} (\overrightarrow{\mu_1} - \overrightarrow{\mu_2})$$

Mahalanobis distance is a very useful statistical analysis tool that can be used in classification, pattern recognition and novelty detection tasks. Mahalanobis distance in high dimensional space such as image classification can be used by accepting penultimate layer outputs of a pre-trained neural network as feature representations[al18a].

Gaussian Discriminant Analysis

The main idea is that pre-trained features can be fitted well by a class-conditional Guassian distribution [al18a]. A pretrained classifier neural network basically defines a posterior distribution $P(y|x)$ where y is the label and x is the input as follows:

$$P(y=c|x) = \frac{\exp\left(w_c^T x + b_c\right)}{\sum_{c'} \exp\left(w_{c'}^T x + b_{c'}\right)}$$

where b_c and w_c are the bias and weight of the softmax classifier for class c. We can define the class conditional distribution as $P(x|y)$, and the prior as $P(y)$. Gaussian discriminant analysis (GDA) suggests that $P(x|y)$ follows a multivariate Gaussian distribution and $P(y)$ follows a Bernoulli distribution [al18a] as follows:

$$P(x|y=c) = \mathcal{N}(x|\mu_c, \Sigma_c)$$

$$P(y=c) = \frac{\beta_c}{\Sigma_{c'}\beta_{c'}}$$

where μ_c, Σ_c and β_c are the empirical mean, empirical covariance and unnormalized empirical prior of the multivariate Gaussian distribution respectively. In addition to the GDA assumption, linear discriminant analysis (LDA) assumes that all classes share the same covariance matrix [al18a]; thus, $\Sigma_c = \Sigma$ for all c. This LDA assumption has been tested during experiments with urine dataset and the assumption is true, having one covariance matrix for all inlier classes give the same result. The posterior can be reformulated as follows:

$$P(y=c|x) = \frac{P(y=c)P(x|y=c)}{\sum_{c'} P(y=c')P(x|y=c')} = \frac{exp(\mu_c^T\Sigma^{-1}x - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + log\beta_c)}{\sum_{c'} exp(\mu_{c'}^T\Sigma^{-1}x - \frac{1}{2}\mu_{c'}^T\Sigma^{-1}\mu_{c'} + log\beta_{c'})}$$

Penultimate Layer Outputs as Weights

Pre-trained softmax classifier can be rewritten if the softmax layer outputs are replaced with previous sequential layer outputs:

$$P(y=c|x) = \frac{\exp\left(w_c^T f(x) + b_c\right)}{\sum_{c'} \exp\left(w_{c'}^T f(x) + b_{c'}\right)}$$

where b_c and w_c are the bias and weight of the softmax classifier for class c and $f(.)$ denotes the output of the penultimate layer l (also referred as $f_l(.)$) of the pretrained neural network. Then they define C class -conditional Gaussian distributions: $P(f(x)|y=c) = \mathcal{N}(f(x)|\mu_c, \Sigma)$ where Σ is the tied covariance matrix and μ_c is the mean of multivariate Guassian distribution of class $c \in \{1, ..., C\}$ [al18a]. The empirical class mean and covariance of training samples $\{(x_1, y_1), ..., (x_N, y_N)\}$ can be computed as below where N is the number of total training images:

$$\hat{\Sigma}_l = \frac{1}{N}\sum_c \sum_{i:y_i=c} (f_l(x_i) - \hat{\mu}_c)(f_l(x_i) - \hat{\mu}_c)^T, \hat{\mu}_c = \frac{1}{N_c}\sum_{i:y_i=c} f_l(x_i)$$

The proposed Mahalanobis confidence score for a given image x from the l-th layer can be calculated based on the above variables and the baseline Mahalanobis formula as:

$$M_l(x) = (f_l(x) - \hat{\mu}_{l,c})^T \hat{\Sigma}_l^{-1} (f_l(x) - \hat{\mu}_{l,c})$$

Calibration Methods

The formula above gives the Mahalanobis distance with respect to the closest class-conditional distribution from the l-th layer of the pre-trained neural network. The weight of each layer (α_l) is learned through and overall Mahalanobis confidence score is formulated as a *logistic regression* model that predicts 0 for out-of-distribution and 1 for in-distribution images [al20a]. Combining the confidence scores per layer is also referred as *Feature ensemble* method. It has been shown in the paper that low-dimensional features often provide higher accuracy during detection and it is the case with the previous layer outputs such as fifth and fourth sequential layers of the pretrained classifier. However, it has been observed during experiments that feature ensemble technique is not necessary. Pretrained classifier has 6 sequential layers and the last sequential layer before the softmax layer output provides consistent confidence scores. Therefore, feature ensemble technique can be seen as an optimization as it increases the accuracy of the OOD analysis. However, the computational burden increases since it requires another logistic regression model to train, storing 6 times more data (covariance matrices) in the device and increases the duration of inference.

In addition to feature ensemble, input pre-processing is also current by adding perturbations to the test images as it was the case in ODIN detector. Perturbed image $\hat{x}_l$ can be calculated from the original image x and tuned on the validation data equivalently as in ODIN $\hat{x}_l = x + \eta.sign(\nabla_x M_l(x)))$. The final Mahalanobis distance score per image x where b is the bias of logistic regression model and detector G_{Mahal} are respectively [al20a]

$$M(x) = \frac{1}{1 + e^{-(\sum_l \alpha_l M_l(\hat{x}_l) + b)}}$$

$$G_{Mahal}(x; \gamma, \alpha_l, \eta, b, f) = \begin{cases} 0 & \text{if } M(x) \leq \gamma \\ 1 & \text{if } M(x) > \gamma \end{cases}$$

4.3.4 Pre-trained Classifier

The pre-trained classifier is a replica model used in fluidlab device to classify among in-distribution images. The network is trained in a supervised manner where the in-distribution images with their correct labels were available. Even though the network is trained for supervised classification purposes, its purpose of usage in the thesis is unsupervised OOD detection where the OOD data is unseen by the network. The network is used by classifier based OOD detection methods such as Maximum Softmax Probability (MSP), ODIN, Mahalanobis and Kolmogorov-Smirnov Test (described in the next section) for OOD detection.

The model accepts (32x32x1) input image array and is composed of 6 sequential blocks, each sequential block is composed of double convolution layers, following by batch normalization

and ReLU activation function. These blocks are flattened to 2048 neurons in the penultimate layer and the final linear layer is composed of 9 output weights. These 9 output neurons represent the 9 inlier classes. Total network is composed of 54249 trainable params with size of 0.21 Mb. Total size of the network with params size and forward/backward pass size is 1.03 Mb. The number of correct and incorrect predictions are summarized with count values and broken down by each class in the confusion matrix below. It can be be said that the network sometimes confuses between classes and the accuracy per class is barely above %95. It might be okay to predict WBC as RBC as they are both blood cells but 51 among 535 samples of uCry are classified as epithelial cells which is a considerable false positive rate.

	uCry	sCry	cCry	hCast	nhCast	sEC	nsEC	WBC	RBC
uCry	392	67	16	2	8	19	8	0	0
sCry	70	423	12	0	3	2	17	0	2
cCry	5	3	96	0	0	0	0	0	1
hCast	0	0	0	83	7	1	0	0	0
nhCast	1	0	0	4	54	3	0	0	0
sEC	15	7	0	9	15	670	100	0	3
nsEC	51	25	7	1	2	85	809	23	6
WBC	0	2	21	0	0	5	57	2657	99
RBC	1	32	4	0	0	1	11	90	1903

Figure 4.16: Confusion Matrix of the pretrained classifier.

4.4 Urine OOD Test

So far we have discussed reconstruction based detection methods with AEs, VAEs and PixelCNN++ model and then classifier based OOD detection methods like MSP, ODIN and Mahalanobis that can be applied to a pre-trained discriminative network. In addition to these methods, we are going to introduce Rabanser's [al19b] shift detection methods to analyze the drift between two distributions. He defined such distributions as feature maps which are the outputs of multiple dimensionality reduction models such as PCA, SPR, label classifiers, trained or random autoencoders. In order to detect the drift, statistical drift detection methods that compare the difference between two distrubutions such as Maximum Mean Discrepancy (MMD) and Kolmogorov-Smirnov (KS) test have been applied to these feature maps. We have used KS test as a simple baseline alternative to MMD. MMD is a kernel based for multivariate two-sample testing whereas KS test is a univariate testing. By multiple univariate KS test, we can simply obtain a multivariate test without a reproducing kernel. This idea is introduced and proven to be feasible[al19b]. The drift detection test can be easily adapted to the urine dataset, let us define the in-distribution samples of urine dataset as reference distribution z_{ref} and out-distribution samples of urine dataset as target distribution z.

By applying suitable DR methods we can obtain low dimensional feature maps of z_{ref} and z. In *Urine OOD Test*, PCA, trained AE, trained VAE, untrained(random) AE and pre-trained classifer is used for DR. z_{ref} and z is obtained by taking 2 dimensional PCA, encoded feature maps with size 256 from AE, VAE and random AE, and finally the penultimate layer output of the classifier with 2048 number of weights. Technically, penultimate layer output of the classifier

is no longer a DR method as images are already 32x32 size; however, the penultimate layer before the softmax layer provides meaningful weight distribution to detect OOD inputs.

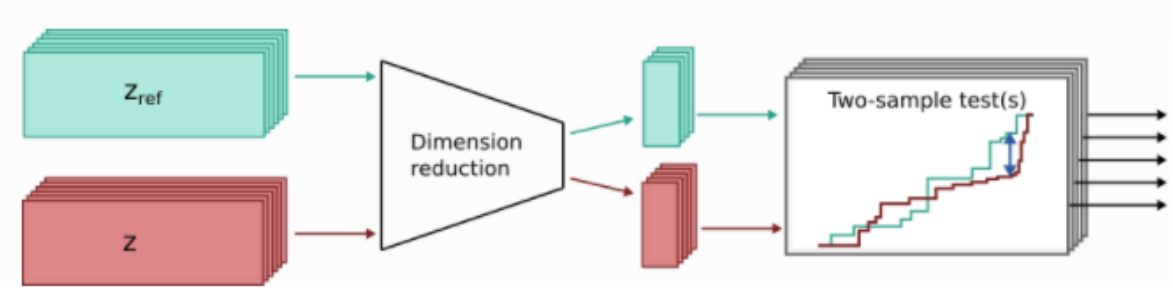


Figure 4.17: Detecting the drift, image from[Ltda], inspired from [al19b]

4.4.1 Kolmogorov-Smirnov Test

Kolmogorov-Smirnov Test (KST) is nonparametric equality test between two one-dimensional probability distribution. A distribution function $F_n(z)$ for n independent and identically distributed [Ano] ordered observations Z_i can be defined as:

$$F_n(z) = \frac{1}{n}\sum_{i=1}^{n} 1_{[-\infty,z]}(Z_i)$$

where $1_{[-\infty,z]}(Z_i)$ equals to 1 if $Z_i \leq z$ and equal to 0 otherwise. The DR techniques we have previously analyzed, PCA, AE, VAE and classifier, they all yield a continuous univariate vectors, such as the latent vector output, penultimate layer of the classifier or two dimensional pricipal component analysis. The idea is to apply univariate testing between representations of inliers and outliers. $F_n(z)$ and $F_q(z)$ will determine the inlier and outlier representations to apply the KST. The Kolmogorov-Smirnov statistic for a given $F_q(z)$ is

$$D_n = sup_z|F_n(z) - F_q(z)|$$

where sup_z is the supremum, the largest absolute difference between two distribution functions across all z values. Obtained D_n values represent statistical distance of each features to the reference distribution. Average of D_n values, mean of distance, will determine the distance of particular outlier sample to the reference in-distribution set. For instance, if the pretrained classifier's penultimate layer outputs is used for DR technique, we will have 2048 dimensional vector for each inlier sample. The same procedure will be applied to outlier samples, 2048 dimensional vector for each outlier image will be obtained. Then KST between inlier representations and representations of test dataset will be done (one by one for each sample of test dataset). The returned average D_n values will determine the distance of the test sample to the inlier distribution. So, Kolmogorov-Smirnov detector G_{KST} can be defined as:

$$G_{KST}(x;\gamma,z_{ref}) = \begin{cases} 1 & \text{if } m(D_n) \leq \gamma \\ 0 & \text{if } m(D_n) > \gamma \end{cases}$$

where $m(D_n)$ is the average value of distance distribution and γ is the threshold value which will be determined in the end manually from KST score plot for each test instance.

5 Results

5.1 Evaluation Metrics

OOD methods are tested based on different evaluation metrics described below
AUROC: Area Under the Receiver Operating Characteristic (AUROC) is a threshold independent metric that measures the area under the ROC curve. A ROC curve shows the trade-off between true positive rate (TPR) and false positive rate (FPR) across different decision thresholds. AUROC is more convenient to use with imbalanced test case scenarios than simple accuracy metric ($\frac{TruePositive+TrueNegative}{AllSamples}$). However, AUROC can give unnecessarily optimistic results when it is applied to test sets where number of negative examples than positive examples are larger in numbers. Nevertheless, urine test dataset is already composed of equal amount of inlier and outlier samples.
AUPR: Area Under the Precision Recall (AUPR) curve is simply the area under the curve of the Precion (y-axis) and Recall (x-axis) plot. Precision is simply $\frac{TruePositive}{TruePositive+FalsePositive}$ and Recall is $\frac{TruePositive}{TruePositive+FalseNegative}$. We have two different AUPR, simply AUPR-IN for in-distribution dataset and AUPR-OUT for out-distribution dataset.
FPR@95TPR: False Positive Rate at %95 True Positive Rate (FPR@95TPR) basically returns the FPR when TPR is at least %95.
Detection Error: Detection Error (DE) returns the misclassification probabilitiy when TPR is %95. Low DE value means the detector will less likely fail at detecting the next sample.

5.2 Urine OOD Test Results

Implementation of OOD detection algorithms are provided publicly at

"https://github.com/erdemunal35/ood_detection_methods" .

5.2.1 Test Models

Urine OOD test is examined in three cases: *Reconstruction Error Based OOD*, *Classifier Based OOD* and *DR Based OOD* and test cases are abbreviated in $X - Y - Z$ format. For the first reconstruction group, X determines the used trained generative model, Y determines the reconstruction error metric and sometimes Z is also defined to show whether perturbations are added or not to the test dataset. For the classifier group, X determines the OOD detector algorithm via its name, Y is classifier to determine its detector type and Z is only available for Mahalanobis-classifier test cases which determines whether perturbations are added to the test dataset or not. For the last DR group, X determines the detection algorithm whether it is Mahalanobis or Kolmogorov-Smirnov Test, Y determines the dimensionality reduction method (as ae, vae and rand-ae where the encoded feature maps of trained Autoencoder, Variational Autoencoder and random/untrained Autoencoder is used).

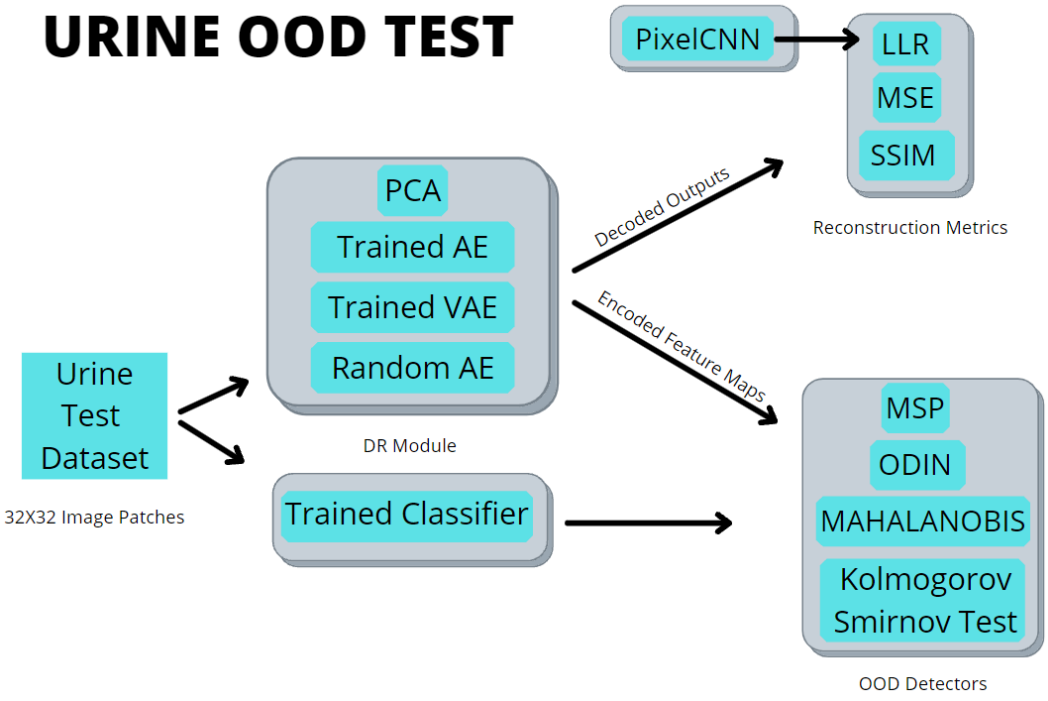
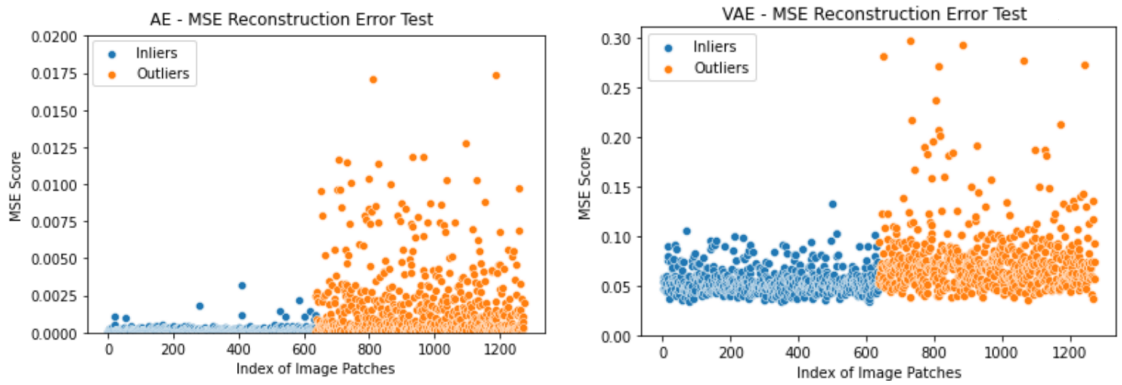


Figure 5.1: Overview of Urine OOD Test

Urine OOD Test					
Method	AUROC↑	AUPR IN↑	AUPR OUT↑	FPR@95↓	Detection Error↓
Reconstruction Error Based OOD Detection					
AE-ssim	0.899	0.887	0.912	0.378	0.287
AE-mse	0.897	0.872	0.915	0.552	0.300
AE-mse-gaussian	0.789	0.723	0.837	0.850	0.449
AE-mse-sp	0.734	0.701	0.776	0.839	0.444
VAE-ssim	0.784	0.713	0.784	0.678	0.353
VAE-mse	0.778	0.687	0.754	0.851	0.382
PixelCNN-MSE	0.902	0.839	0.876	0.307	0.198
Classifier Based OOD Detection					
MSP	0.505	0.502	0.501	0.948	0.496
ODIN-gaussian	0.562	0.573	0.504	0.903	0.475
ODIN-sp	0.334	0.403	0.389	0.984	0.499
Mahal-classifier	0.969	0.970	0.966	0.138	0.093
Mahal-classifier-gaussian	0.788	0.727	0.807	0.839	0.444
Mahal-classifier-sp	0.645	0.584	0.718	0.923	0.481
KSD-classifier	0.723	0.735	0.720	0.753	0.399
DR Based OOD Detection					
Mahal-PCA	0.665	0.695	0.745	0.897	0.425
Mahal-ae	0.881	0.830	0.903	0.743	0.395
Mahal-vae	0.492	0.478	0.516	0.979	0.5
Mahal-rand-ae	0.625	0.624	0.649	0.842	0.497
KSD-withoutDR	0.892	0.892	0.875	0.287	0.187
KSD-PCA	0.875	0.847	0.845	0.387	0.284
KSD-ae	0.894	0.899	0.879	0.365	0.206
KSD-vae	0.531	0.542	0.551	0.918	0.491
KSD-rand-ae	0.477	0.444	0.584	1	0.5

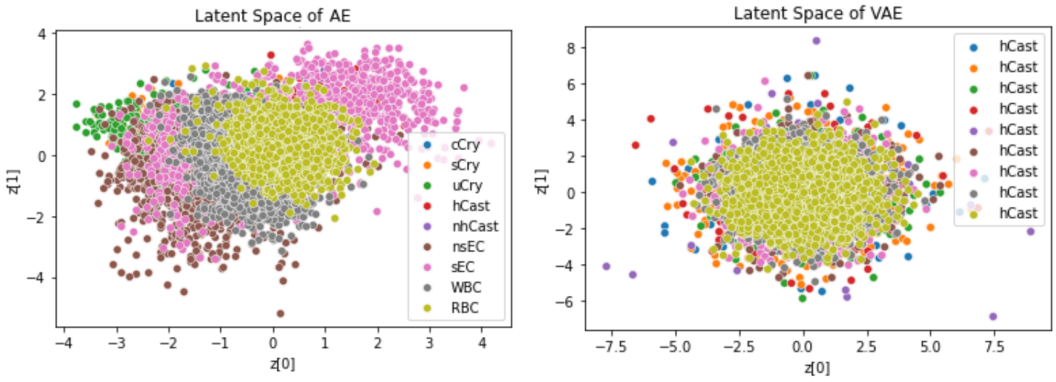
5.3 Reconstruction Error Based OOD

We have experienced disappointing results with the Variational Autoencoder. Compared to autoencoder, latent space was expected to be regularized and generalized; however, latent space was overly generalized. This generalization was expected for inlier samples only as the VAE is trained with in-distribution samples, encouraged to generalize the latent features due to the KL divergence term and it is forced to reconstruct images as an aggregation of these features. However, outlier samples were also included in this generalization. It is not possible to discuss about anomalies or normalities in the feature space as the latent features of VAE are overlapped. Moreover, reconstruction error test as you can see in figure 5.2, showed that autoencoder achieved better at assigning higher errors for outlier samples than inliers; however, this is only true for some extent, meaning that there is no possible MSE Score value we can assign as a threshold to separate between inliers and outliers.



(a) (b)

Figure 5.2: Trained Autoencoder (a) and Variational Autoencoder (b) OOD detectors based on reconstruction outputs on urine test set with MSE scores as evaluation metric



(a) (b)

Figure 5.3: 2 dimensional PCA of the latent space of trained Autoencoder (a) and Variational Autoencoder (b)

Also as you can see from the latent space generation of the VAE (figure 5.4), VAE drastically generalizes and does not learn high level features of inlier samples rather than a single dot and white background. This is because of the simplicity of the model, the size, lack of dropout and normalization in layer outputs and more importantly lack of smart architecture design. A basic VAE architecture failed the test even with optimizations. The aim was to solve the OOD detection problem with less computational burden as possible but it was not possible.

Alternative metric SSIM also obtained slightly better AUROC score than overall MSE score;

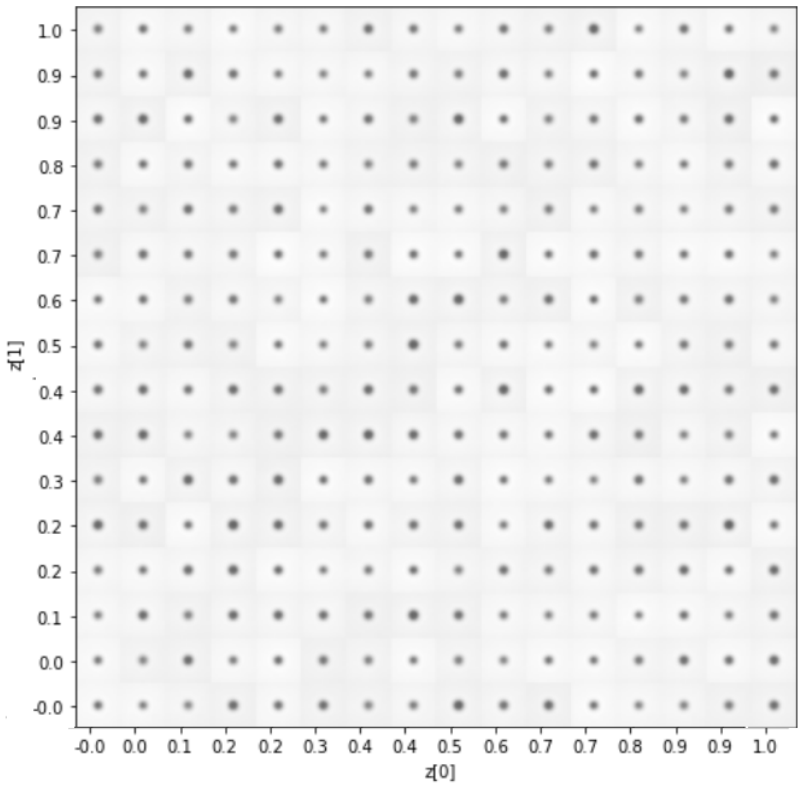


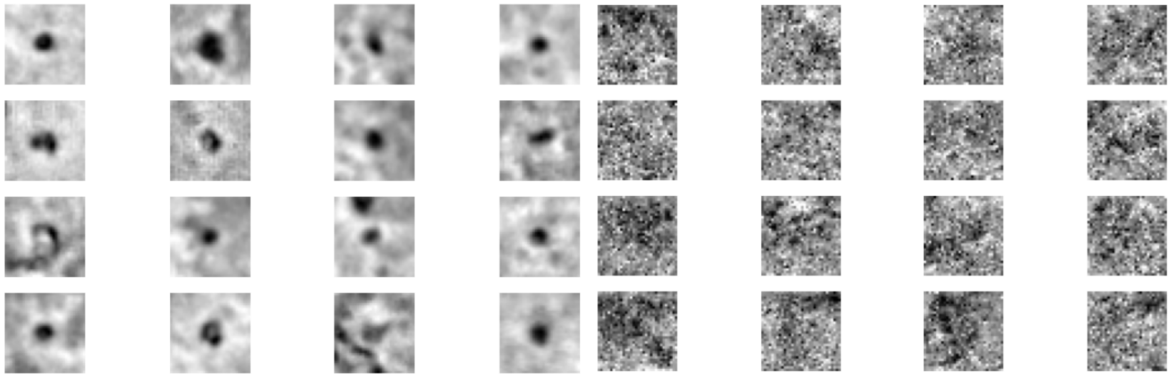
Figure 5.4: Latent Space Generation of trained VAE

however, the difference between the scores are quite low and proportional with MSE. It is also observed that SSIM score for particular test image samples are proportional with their MSE score. Therefore, MSE is still a viable metric to compare between reconstruction outputs and inputs for urine dataset. SSIM does provide significant improvement to understand the structural similarity between two samples. However, it does not provide any significant insight to understand the structural similarity between urine image patches.

5.3.1 Likelihood Ratios Metric with PixelCNN++ Model

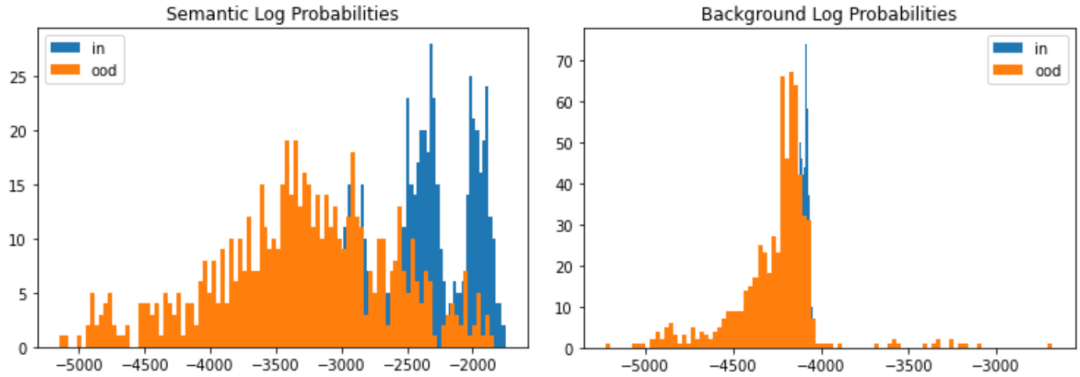
We have trained two generative PixelCNN++ models namely semantic model and background model on original and perturbed training dataset. We have observed that a state-of-the-art autoregressive model can significantly succeed at fitting the in-distribution dataset as you can see in latent space generation of the semantic model. You can observe blood cell images are generated in detailed contrast, they almost look like original red/white blood cells from our in-distribution images. However, none of these generated samples are from other classes such as crystals or casts. This is an unexpected result but a proper reason for this is urine inlier dataset is imbalance. Some classes, which are few in numbers like casts and crystal, compared to blood cell images are tried to be increased in numbers before training by adding augmentations and oversampling. However, neither of these approaches can diversify the semantic pattern of one image and the probabilistic non-linear neural network can not learn or rather generalize these semantically very different inlier classes to its latent space. As a result, we see nothing but blood cell images in the latent space.

We obtained a very similar result with the baseline VAE; however, the network was almost properly reconstructing given outlier inputs even though its latent space was sampling simple blood cell images. In the latent space generation of background model, we do not see any semantic structure like a cell, but we rather observe the same background statistics plus Bernoulli perturbation. Therefore, we can approve that semantic and background models are generating reasonably expected samples from their latent space. When we examine the



(a) (b)

Figure 5.5: Latent space generation of Semantic Generative Model(a) and Background Generative Model(b)



(a) (b)

Figure 5.6: y-axis:Histogram, x-axis:Log-likelihood, (a) Semantic model Likelihood, (b) Background model Likelihood

log likelihood-histogram plots of semantic and background models under urine test dataset, we observe that inlier samples are assigned higher likelihood in both models. This result is again different than literature since it is claimed that OOD inputs obtain higher likelihood in literature datasets and to overcome that LLR method is recommended. However, our model simply returns higher likelihood to inlier samples on just the semantic model. The background model likelihoods are not separable between inliers and outliers but likelihood is arguably higher for inliers. The likelihood ratio (LLR) between these plots are taken by simply subtracting the log-likelihood of semantic and background models. LLR improves the standard likelihood but it still does not solve our OOD detection problem. In the same figure 5.7, reconstruction error test done with the semanctic model on test dataset is illustrated. When we compare the reconstruction errors we can see that there is a clear improvement in both AUROC and MSE score plot. However, a robust threshold still cannot be drawn to detect OOD samples with low False Positive rate or Detection Error.

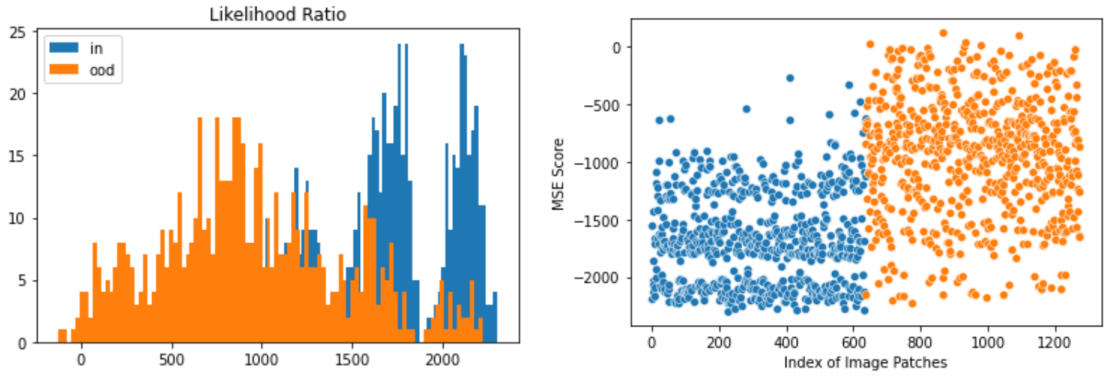


Figure 5.7: (a) Likelihood Ratios = (Semantic Log Probabilities) - (Background Log Probabilities), (b) MSE score on PixelCNN model reconstructions

5.4 Classifier Model Based OOD

[Dan16] claimed that out-distribution datasets have higher softmax scores than in-distribution datasets. A simple test has been applied to the pre-trained classifier and urine test dataset. It is obtained that the baseline detection algorithm completely fails at detecting the outliers. Maximum softmax probabilities of inliers and outliers are not distinguishable; hence, the MSP score plot of urine test dataset resembles a random detector. The MSP scores were plotted below where the blue and orange points are inliers and outliers of the urine test dataset respectively. It is also observed that the algorithm works accordingly with UTKFace dataset [al17b] which can be an example out-distribution dataset for the urine dataset.

The second OOD detection method, ODIN [Shi17], introduced temperature scaling and input preprocessing calibration techniques to the MSP approach to calibrate the maximum softmax scores. As the MSP algorithm behaved as a random classifier, calibration of the MSP scores resulted shifting of MSP scores. Using input perturbations on test dataset, decreased the score of both inliers and outliers proportionally. However, when the outlier dataset is composed of UTKFace dataset, we can confirm that perturbations do decrease the softmax score of inliers more than outliers, resulting a clear plot (figure 5.9 (c)) to draw a threshold with an AUROC value of 1.

5.4.1 Why do MSP and ODIN fail?

MSP and calibrated MSP (ODIN) methods act like a random classifier and fail at detecting outlier samples of urine dataset but can detect outlier samples from a completely different out-distribution dataset like UTKFace with %100 accuracy when a suitable threshold γ is chosen. This is the first hint and first realization of the importance of out-distribution dataset. The outlier samples of urine dataset is sometimes impossible to detect by human eyes. Therefore, it is important to state that the research aim of this paper is 'out-ouf-distribution detection' but the out-distribution dataset is not semantically or visually separable as the ones in literature. The name 'out-distribution' is still technically correct to define outlier samples of urine dataset but it is evident that the research task is the same as literature according to

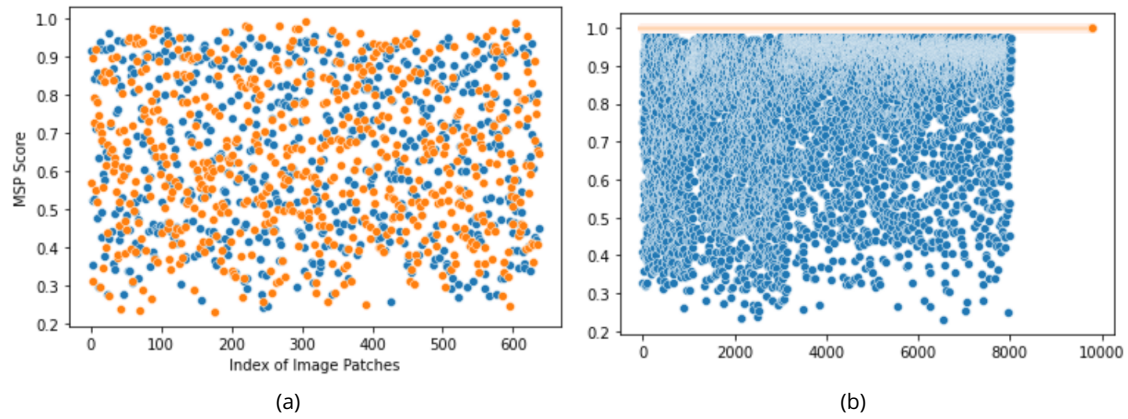


Figure 5.8: MSP classifier results, (a) urine test dataset where blue: inliers, orange: outliers, (b) blue: urine in-distribution dataset, orange: samples of UTKFace [al17b] dataset

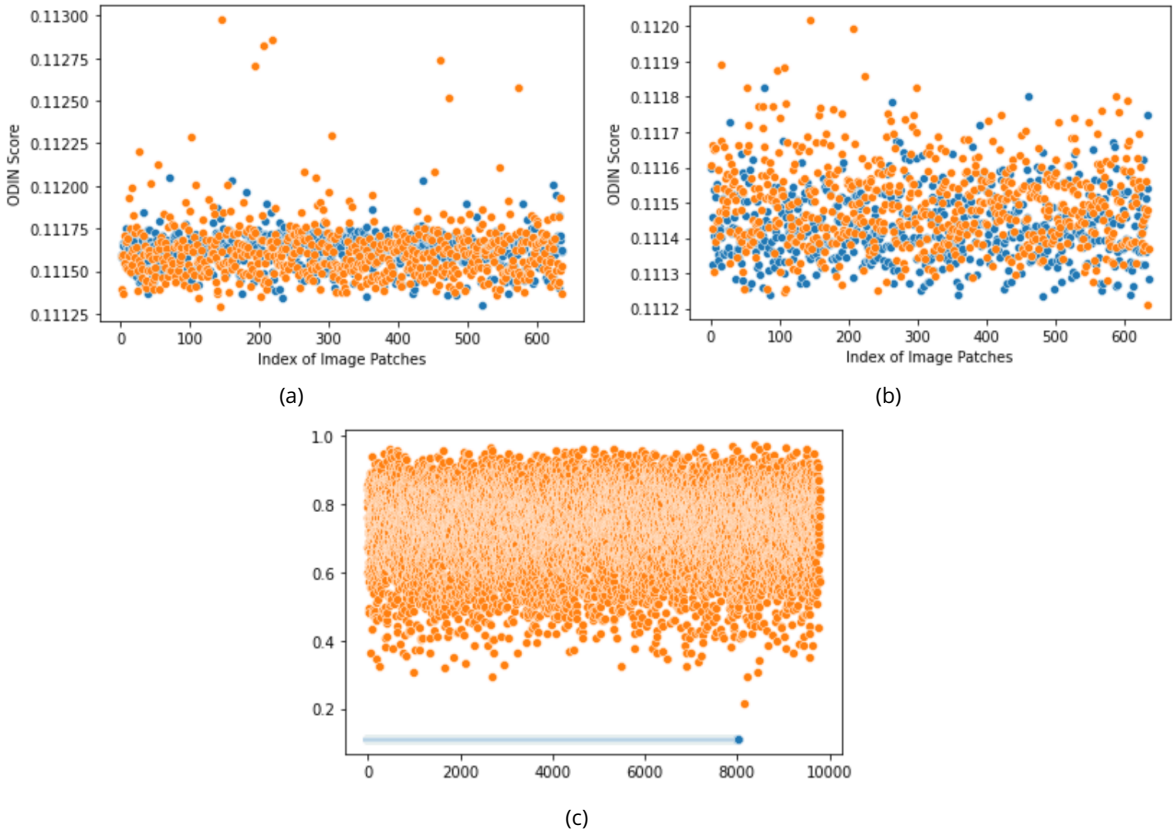


Figure 5.9: ODIN classifier results, blue: inliers, orange: outliers, (a) and (b) urine test dataset, (c) blue: urine in-distribution dataset, orange: samples of UTKFace [al17b] dataset

the name but it is very different in practice. Therefore, baseline OOD approaches are not useful for urine dataset. Robust OOD detection methods which are adversarial friendly have to be investigated.

5.5 Mahalanobis Detector

As you can see in the table, Mahalanobis distance based OOD detector using penultimate layers of the classifier, briefly Mahal-classifier, outperforms other OOD detectors by a

big margin. First of all, proposed[al18a] input pre-processing calibration method, adding

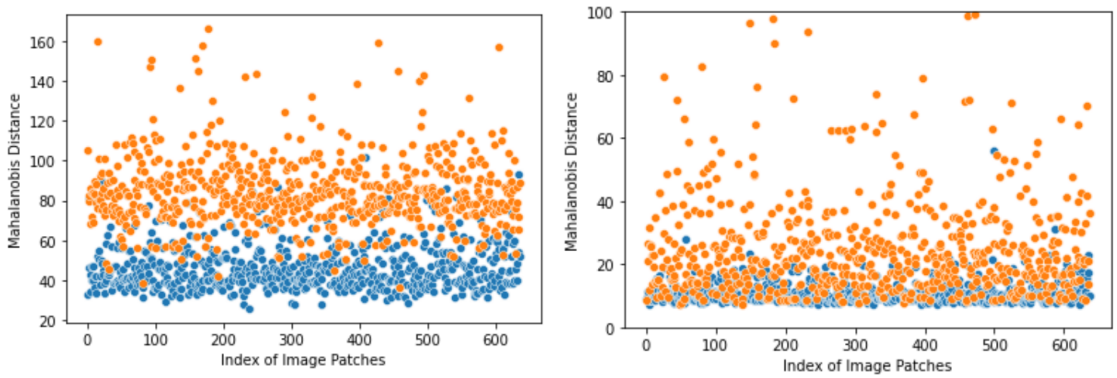


Figure 5.10: (a) Mahal-classifier vs (b) Mahal-ae models distance scores blue: inliers, orange: outliers

perturbation to test inputs, does not increase the Mahalanobis distance of outliers more than inliers. Adding gaussian or salt&pepper perturbation to test images increase the Mahalanobis distance of inliers more than outliers, making them inseparable in the score plot. This is another contradiction with literature[al18a] where input preprocessing increased the AUROC and overall accuracy of the Mahalanobis model. We conclude that perturbations should not be added to urine dataset during test for any kind of OOD detection algorithm. Because perturbations make both inlier and outlier input samples more noisy than before. This makes both inliers and outliers undetected by the classifier, resulting low probability in softmax layer. Hence, it makes them inseparable by the classifier and not suitable for detection purposes.

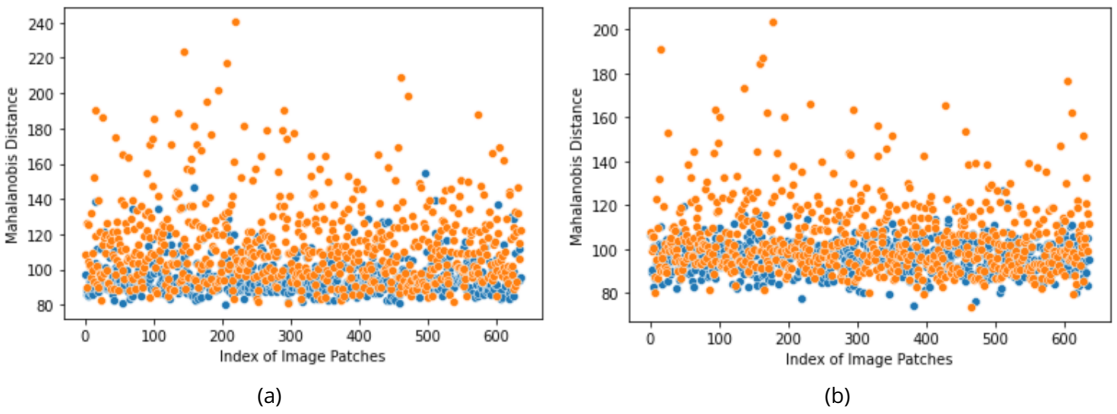


Figure 5.11: (a) Mahal-classifier-gaussian vs (b) Mahal-classifier-sp distance scores, blue: inliers, orange: outliers

Secondly, Mahalanobis distance based detector using feature embeddings of AE, VAE and rand AE models, fail at giving distinct distances for outliers and inliers. For Mahal-ae model, it is clear that some of the outliers are assigned to higher Mahalanobis distance but most of the outliers received low distance as the inliers, again resulting inseparability between test samples in the distance plot. Multiple scenarios where autoencoders have different architecture and hyperparameters are trained and it is observed that large latent space increase the detection accuracy. This can be one reason why classifier based Mahalanobis detector outperforms generative model based DR methods. However, no matter how large

the latent space, almost half of the outlier samples in the test dataset receive low Mahalanobis distance.

In order to better understand why Mahalanobis detector fails detecting feature embeddings of generative model, we have to visualize the image patches which are labelled as *outlierish* by having the highest Mahalanobis distance and *inlierish* by having the smallest distance. The figure at 5.12, urine test dataset images are sorted from the most inlierish to the most outlierish based on Mahal-classifier detector scoring. Only some extent of the images are shared from the most inlier to most outlier. As you can see, outlier samples that are empty (only background) images, multiple blood cells in one patch are detected with high Mahalanobis distance.

However, in figure 5.13 where Mahal-ae detector scoring based illustration, already fails at the most top (outlierish) and bottom (inlierish) images. There you may see that image patches that should have detected as outlier are red squared and images should have detected as inlier are green squared. It is clear that an autoencoder embeds train images based on their pixel value; therefore, empty image patches are encoded similarly as blood cell images as most of their pixel values are background. In addition, images that have dense dark semantic pattern like crystals are embedded close with other outliers. This shows that generative model embedding is misleading compared to classifier embedding. Classifier is trained based on labels, so DNN is aware of class labels and as a result the penultimate layer outputs are returned with class information. However, the generative models are trained without additional class information, they are trained with an unsupervised fashion to reconstruct better decodings. Therefore, outlier samples with high pixel instensity values are aligned together and resulted high Mahalanobis distance together with inlier samples that share similar semantic pattern.

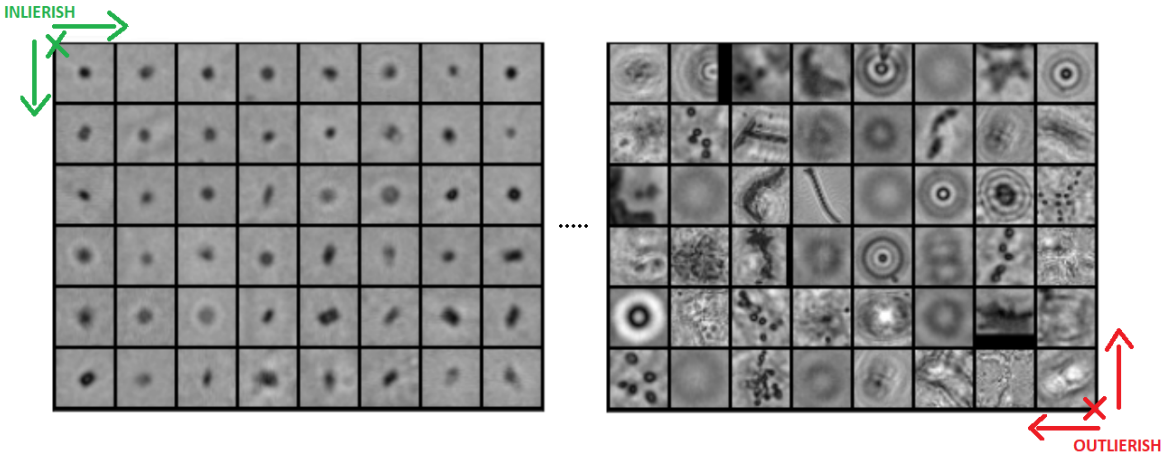


Figure 5.12: Test images are sorted from low score to high score based on Mahal-classifier detector

5.6 Kolmogorov-Smirnov Detector

Two sample KS Test is applied to multiple distributions of inlier and outlier duple. These duple distributions are obtained by classifier's penultimate layer outputs, and encoded feature mappings of AE, VAE, random AE. The highest accuracy is obtained with KSD-withoutDR tests

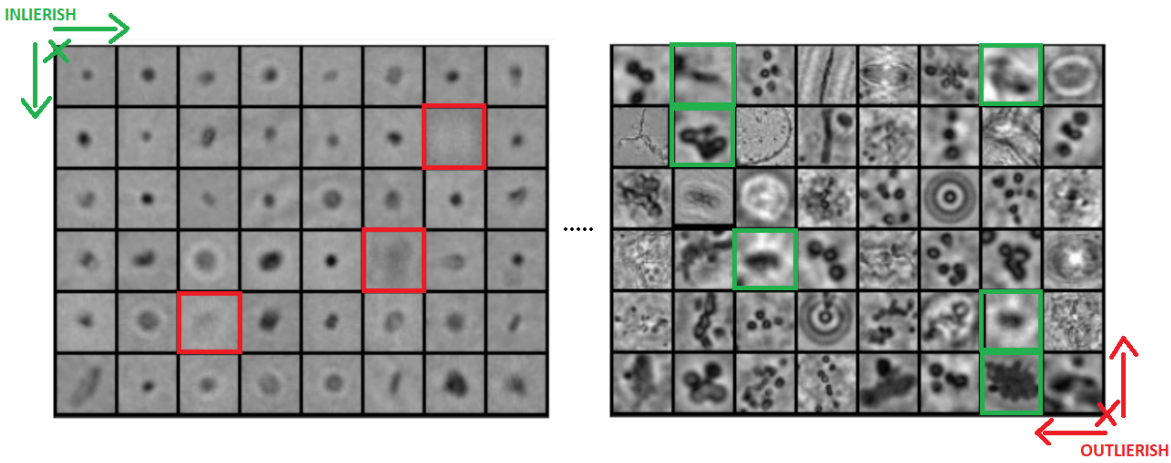


Figure 5.13: Test images are sorted from low score to high score based on Mahal-ae detector

where the distributions are directly obtained by flattening the two dimensional 32x32 image samples; hence, no dimensionality reduction. The worst results are obtained by embeddings of variational autoencoder and random autoencoder. None of the KSD detectors yield more robust detection accuracy compared to previous Mahal-classifier detector. However, KS Test uncovers important conclusions.

First of all, let us analyze why KS Test failed with classifier as its DR method. 2048 number of weights are returned from classifier for each inlier and outlier sample. Applying KS test to compare between two 2048 dimensional weight distribution returns a distance feature array with a length of 2048. Most of the weights share comparably similar values regardless of input array, inlier or outlier. In other words, they are *joint variable*. As a result, the distance returned for the representative features of these weights is approximated to zero. Since most of the elements of the distance feature array is zero, returned mean value gets closer to zero. As a result we obtain a KS Distance plot in figure 5.14 (b) where outliers' distance score is highly correlated to inliers. However, this situation does not apply for Mahal-classifier detector due to the covariance matrix. When an outlier and inlier sample is passed through the Mahal-classifier detector, their weight distributions are multiplied with the covariance matrix to calculate the Mahalanobis distance. Joint variable weights that resulted 0 in the distance array of KS Test, are disregarded as they share close values in the covariance matrix.

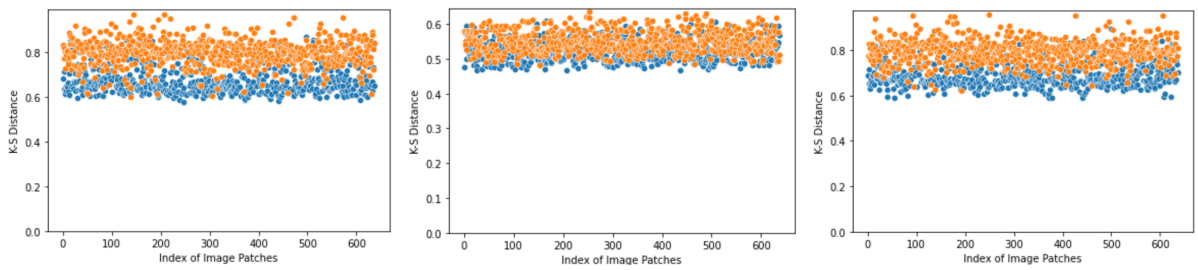


Figure 5.14: (a) KSD-pure, (b) KSD-classifier and (c) KSD-ae detectors on urine test set scores blue: inliers, orange: outliers

5.7 Proposed Solution: Double-threshold Mahalanobis Detector

All in all, a robust fully accurate solution to detect outlier samples could not be achieved yet. To reduce the number of false positives and false negatives, we propose Double-threshold Mahalanobis Detector. Samples received Mahal distance between these threshold values will be labelled as *undetermined* for further analyses by lab assistants.

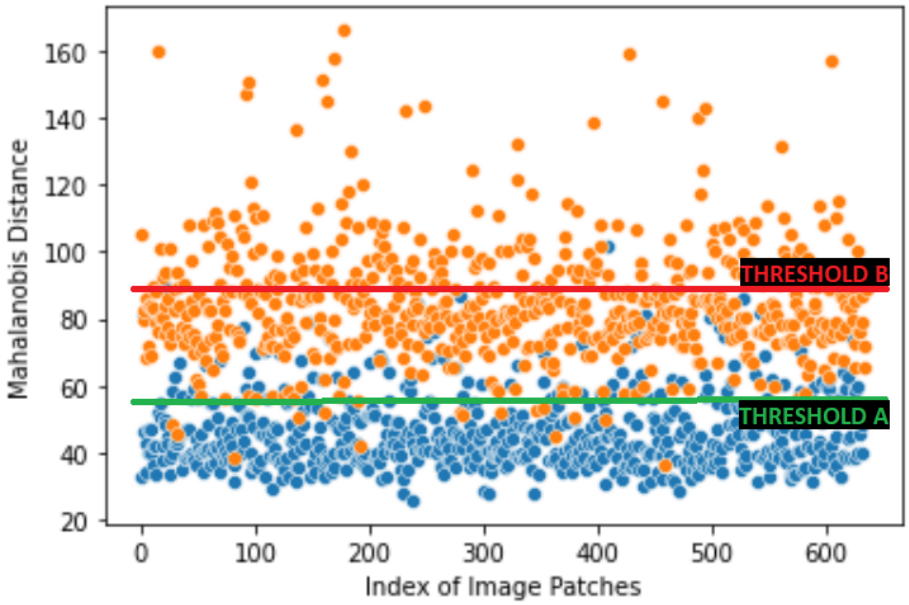


Figure 5.15: Proposed Mahalanobis detector with double thresholds

5.7.1 What makes Mahalanobis method effective and applicable?

Mahal-classifier is not just the most accurate detector among other tests but also it is the most applicable method for fluidlab. One of the reason is simplicity. Mahal-classifier detector only requires a covariance matrix additional to a pretrained network inside the device. Second reason is inference speed. Once the covariance matrix and mean vectors are available, computation of Mahalanobis distance corresponds to multiple vector multiplication with penultimate weights.

Third reason is reproducibility, as new inlier samples are introduced to urine dataset, covariance matrix and mean vectors per class can be updated accordingly. Even new inlier classes can be introduced to the system, this makes Mahalanobis detector class-incremental. However, there is one constraint which is having a large covariance matrix might cause a possible memory problem. The matrix is computed by the covariance of penultimate layer outputs so the matrix size is 2048x2048. The value in each block of matrix is float, so a rough assumption of required memory is $2048x2048x8Bytes \approx 33Mbs$. However, we know that covariance matrix is always symmetric and positive semi-definite. Knowing that other half of the matrix that is separated by diagonal values can be excluded. Plus, instead of using float64 that uses 64 bit numbers, float32 that uses 32 bits can be used. Therefore, required memory can be downsized to $\approx 8Mbs$.

6 Conclusion and Further Work

6.1 Conclusion

This thesis has discussed the applicability of baseline generative model based and classifier based OOD detection methods on microscopic images. It has been shown that common unsupervised OOD detection algorithms fail drastically depending on the used OOD dataset. Common input pre-processing calibration techniques such as adding perturbations to test samples were also found to be not applicable for urine dataset. Simple unified framework: Mahalanobis distance that uses multi-dimensional measuring of standard deviations of penultimate layer outputs of classifier have been found highly effective to resolve the OOD detection problem in microscopic images.

Furthermore, vast majority of literature research has been done in generative model based OOD methods due to their recent popularity and higher accuracy compared to baseline classifier based OOD detection approaches. We have found that encoder-decoder based schemes are impractical to detect OOD microscopic samples. The generative models are inclined to learn the semantic pattern distribution on given data. As a result, neither the learnt latent space distribution nor reconstruction error based algorithms are found to be effectual to detect OOD features. Indeed, only a baseline part of generative model based OOD detection methods have been evaluated in this research. However, statistical shift detection algorithms were impotent to detect the drift in feature distributions. These feature representations varied from penultimate layer outputs to latent space of trained autoencoder schemes. We understood that drift fails on generative models since the latent space of generative models were seriously generalized and overfitting. Moreover, its observed that drift detection on weight distributions of classifier's penultimate outputs was not feasible due to many joint variable weight distributions. Further dimensionality reduction methods are required to fix the joint variable problem by discarding them from the weight distributions. With these further adjustments and optimizations, statistical drift detection might succeed detecting OOD samples much more effectively than Mahalanobis method. In fact, even Mahalanobis detector might achieve tremendous results with such aggregations and obtain even more accurate detection.

6.2 Future Work

Even though we already concluded that generative models are generalizing on inliers and failing to detect outliers, we only elaborated autoencoders trained without class information where each inlier samples are assumed to be one class by the AE. Therefore, a label-aware generative model can specifically learn class related semantic patterns only. However, this is not feasible to implement yet as there is not enough training samples for each inlier classes as we have discussed. Furthermore, we did not elaborate it in the thesis but an ensemble autoencoder model was trained where each autoencoder trained only on a specific class of in-distribution dataset. Then each autoencoder is tested with outliers by MSE and SSIM reconstruction error metrics. We observed that the AE trained with blood cell samples obtained worse reconstruction with outliers than other trained AEs with other classes. However, the latent space formation of these ensembled AEs were not examined with statistical drift analysis or Mahalanobis distance based framework. In addition, a conditional autoencoder[Soh15] where input images are trained with an additional variable c to determine its class can be used instead of an ensemble structure to avoid potential computational burden.

This research is determined to analyze unsupervised OOD detection methods; however, semi-supervised approaches may be introduced to increase the detection performance. Lack of outlier samples was the main limiting criterion on trying supervised or semi-supervised algorithms. However, if the main constraint is resolved, one of the latest promissing classifer based OOD detection method, Outlier Exposure[al18e] can provide very useful insights. Urine in-distribution dataset is very large and complex to distinguish between classes. By training the classifiers on auxiliary samples of outliers with label 1 can directly improve the detection performance. Furthermore, Black Box predictors[al18b] can be used to reduce the dimension of weight distributions and adjust the joint layer problem in classifier penultimate layer. It is observed that estimating outlier drift in high dimensional representations is a meticulous subject. and requires optimization. Finally, kernel based multivariate two-sample statistical Maximum Mean Discrepancy (MMD) can be used as an advanced statistical test instead of multiple univarite Kolmogorov-Smirnov test. The kernel based detector might increase the computational cost; however, it might detect a possible shift that the KS test failed to detect.

Bibliography

[HW79] John A Hartigan and Manchek A Wong. *A k-means clustering algorithm. Journal of the Royal Statistical Society*. 1979.

[FV82] James D. Foley and Andries Van Dam. *Fundamentals of Interactive Computer Graphics*. USA: Addison-Wesley Longman Publishing Co., Inc., 1982. ISBN: 0201144689.

[McL99] G J McLachlan. *Mahalanobis Distance*. 1999.

[al00] Bernhard Scholkopf et al. *Support Vector Method for Novelty Detection*. 2000.

[al02] Eleazar Eskin et al. *A geometric framework for unsupervised anomaly detection*. 2002.

[al04] Zhou Wang et al. *Image Quality Assessment: From Error Visibility to Structural Similarity*. 2004.

[Mil+06] Peter A. Milder et al. "Fast and Accurate Resource Estimation of Automatically Generated Custom DFT IP Cores". In: *Proceedings of the 2006 ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays*. FPGA '06. Monterey, California, USA: Association for Computing Machinery, 2006, pp. 211–220. ISBN: 1595932925. DOI: 10.1145/1117201.1117232. URL: https://doi.org/10.1145/1117201.1117232.

[Was06] Larry Wasserman. *All of Nonparametric Statistics*. 2006.

[al09] Deng et al. *Imagenet: A large-scale hierarchical image database*. 2009.

[KH09] Alex Krizhevsky and Geoffrey. Hinton. *Learning multiple layers of features from tiny images*. 2009.

[LeC10] Cortes Corinna LeCun Yann. *Mnist handwritten digit database.* 2010.

[al11a] Emmanuel J Candes et al. *Robust principal component analysis?* 2011.

[al11b] Netzer et al. *Reading digits in natural images with unsupervised feature learning*. 2011.

[Jol11] Ian Jolliffe. *Principal component analysis.* 2011.

[al14] Diederik P. Kingma et al. *Auto-Encoding Variational Bayes*. 2014.

[Anh14] Jeff Clune Anh Nguyen Jason Yosinski. *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*. 2014.

[May14] Takehisa Yairi Mayu Sakurada. *Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction*. 2014.

[al15a] James A. Jablonski et al. *Principal Component Reconstruction Error for Hyperspectral Anomaly Detection*. 2015.

[al15b] Yu et al. *Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop.* 2015.

[Ian15] Jonathon Shlens Ian J Goodfellow. *Explaining and harnessing adversarial examples*. 2015.

[LB15] Aäron van den Oord Lucas Theis and Matthias Bethge. *A note on the evaluation of generative models*. 2015.

[Soh15] Kihyuk et al. Sohn. *Learning Structured Output Representation using Deep Conditional Generative Models.* 2015.

[Sun15] Jinwon An Sungzoon Cho. *Variational Autoencoder based Anomaly Detection using Reconstruction Probability*. 2015.

[al16a] Shaofeng Zou et al. *Nonparametric Detection of Anomalous Data Streams*. 2016.

[al16b] Irina Higgins et al. *beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*. 2016.

[al16c] Oord et al. *Pixel recurrent neural networks*. 2016.

[Dan16] Kevin Gimpel Dan Hendrycks. *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks*. 2016.

[DB16] Alexey Dosovitskiy and Thomas Brox. *Generating images with perceptual similarity metrics based on deep networks*. 2016.

[al17a] Chuan Guo et al. *On Calibration of Modern Neural Networks*. 2017.

[al17b] Zhang et al. *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[al17c] Salimans et al. *A PixelCNN implementation with discretized logistic mixture likelihood and other modifications.* 2017.

[al17d] Salimans et al. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*. 2017.

[Cho17] Randy C. Paffenroth Chong Zhou. *Anomaly Detection with Robust Deep Autoencoders*. 2017.

[Shi17] R. Srikant Shiyu Liang Yixuan Li. *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks*. 2017.

[al18a] Kimin Lee et al. *A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks*. 2018.

[al18b] Zachary C. Lipton et al. *Detecting and Correcting for Label Shift with Black Box Predictors*. 2018.

[al18c] Bo Zong et al. *Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection*. 2018.

[al18d] Choi et al. *Generative ensembles for robust anomaly detection*. 2018.

[al18e] Dan Hendrycks et al. *DEEP ANOMALY DETECTION WITH OUTLIER EXPOSURE*. 2018.

[al18f] Kingma et al. *Glow: Generative flow with invertible 1x1 convolutions*. 2018.

[al18g] Nalisnick et al. *Do deep generative models know what they don't know?* 2018.

[al18h] Shafaei et al. *Does your model know the digit 6 is not a cat? a less biased evaluation of" outlier" detectors*. 2018.

[Ber18] et al. Bergmann Paul. *Improving unsupervised defect segmentation by applying structural similarity to autoencoders.* 2018.

[Car18] Vijay S. Pande Carlos X. Hernández Mohammad M. Sultan. *Using Deep Learning for Segmentation and Counting within Microscopy Data*. 2018.

[Mat18] Alex P. Pentland Matthew A. Turk. *Face Recognition Using Eigenfaces*. 2018.

[Ter18] Graham W. Taylor Terrance DeVries. *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. 2018.

[al19a] Jie Ren et al. *Likelihood Ratios for Out-of-Distribution Detection*. 2019.

[al19b] Rabanser et al. *Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift*. 2019.

[Ber19] et al. Bergmann Paul. *MVTec AD–A Comprehensive Real-World Datasetfor Unsupervised Anomaly Detection.* 2019.

[CC19] Raghavendra Chalapathy and Sanjay Chawla. *Deep learning for anomaly detection*. 2019.

[al20a] Jiefeng Chen et al. *Robust Out-of-distribution Detection for Neural Networks*. 2020.

[al20b] Lu Wang et al. *Image Anomaly Detection Using Normal Data Only by Latent Space Resampling*. 2020.

[al20c] Tsatsral Amarbayasgalan et al. *Unsupervised Anomaly Detection Approach for Time-Series in Multi-Domains Using Deep Reconstruction Error*. 2020.

[Deh20] et al. Dehaene David. *Iterative energy-based projection on a normal data manifold for anomaly localization.* 2020.

[Lir20] Yedid Hoshen Liron Bergman. *CLASSIFICATION-BASED ANOMALY DETECTION FOR GENERAL DATA*. 2020.

[Zhi20] Yali Amitr Zhisheng Xiao Qing Yan. *Likelihood Regret: An Out-of-Distribution Detection Score For Variational Auto-encoder*. 2020.

[al] Zhou Wang et al. URL: https://www.cns.nyu.edu/~lcv/ssim/.

[Ano] Anonymous. URL: https://en.wikipedia.org/wiki/Kolmogorov%5C%E2%5C%80%5C%93Smirnov_test.

[Ltda] Seldon Technologies Ltd. URL: https://docs.seldon.io/projects/alibi-detect/en/latest/cd/background.html.

[Ltdb] Seldon Technologies Ltd. URL: https://docs.seldon.io/projects/alibi-detect/en/latest/od/methods/llr.html.

[Mun] Abhishek Mungoli. URL: https : / / towardsdatascience . com / dimensionality - reduction-pca-versus-autoencoders-338fcaf3297d.

[Pla] John Platt. *Platt scaling*. URL: https://en.wikipedia.org/wiki/Platt_scaling.

[Ste] Fraedrich Stefan. URL: https://anvajo.com/about-us.

List of Figures

2.1 Step by step preparation of urine sample analysis 7
2.2 Image Analysis 7
2.3 Segmented Image Patches where red squares represent red blood cell . . 7
2.4 Inliers (black) and some outliers (red) are exemplified, numbers under each class represent the amount of available samples 8

3.1 How SSIM score differs where MSE score continues returning the same error, image source[al] 10
3.2 Pipeline of the VQ-VAE framework, image source[al20b] 11
3.3 Rabanser's pipeline to detect shift between two dataset using DR methods. Image from [al19b] 11
3.4 An overview of Deep Autoencoding Gaussian Mixture, [al18c] 12
3.5 Likelihood of outliers is higher that in-distribution dataset (Fashion-MNIST) but likelihood-ratio is higher for in-distribution samples than OOD samples(MNIST), image from [al19a] 13
3.6 Comparison of AUROC (%) between classifier-based OOD detectors where the x-axis represents the number of training data. (a) small number of training data, (b) Random label is assigned to training data [al18a] 14

4.1 Example of a rotated and oversampled crystal image 16
4.2 Inlier and outlier image analysis with 2 principle components. The red circle represents the same region between two plots. 17
4.3 Encoded inlier and outlier images in 2 values. The red circle represents the proportional region between two plots. 18
4.4 Encoding and reconstruction of a sample inlier by two different autoencoders 19
4.5 Encoding and reconstruction of a sample outlier by two different autoencoders 19
4.6 Importance of choosing low encoding dimension for anomaly detection via error in reconstructed outputs 19
4.7 Difference between Traditional (Deterministic) Autoencoder and Variational (Probabilistic) Autoencoder 21
4.8 Example of an input image, encoded version and decoded reconstruction image 22

4.9 VAE Model Loss 22
4.10 AE Model Loss 23
4.11 Encoding and reconstruction of a sample inlier by Autoencoder and Variational Autoencoder 23
4.12 Encoding and reconstruction of a sample outlier by Autoencoder and Variational Autoencoder 24
4.13 Architecture of PixelCNN++ 24
4.14 Sample Reliability Diagram, Left uncalibrated Resnet110 on Cifar100, Right calibrated version with Temperature Scaling calibration method 26
4.15 Perturbation example, (a) the original image, (b) gaussian perturbation added, (c) salt&pepper perturbation added 27
4.16 Confusion Matrix of the pretrained classifier. 31
4.17 Detecting the drift, image from[Ltda], inspired from [al19b] 32

5.1 Overview of Urine OOD Test 34
5.2 Trained Autoencoder (a) and Variational Autoencoder (b) OOD detectors based on reconstruction outputs on urine test set with MSE scores as evaluation metric 36
5.3 2 dimensional PCA of the latent space of trained Autoencoder (a) and Variational Autoencoder (b) 36
5.4 Latent Space Generation of trained VAE 37
5.5 Latent space generation of Semantic Generative Model(a) and Background Generative Model(b) 38
5.6 y-axis:Histogram, x-axis:Log-likelihood, (a) Semantic model Likelihood, (b) Background model Likelihood 38
5.7 (a) Likelihood Ratios = (Semantic Log Probabilities) - (Background Log Probabilities), (b) MSE score on PixelCNN model reconstructions 39
5.8 MSP classifier results, (a) urine test dataset where blue: inliers, orange: outliers, (b) blue: urine in-distribution dataset, orange: samples of UTKFace [al17b] dataset 40
5.9 ODIN classifier results, blue: inliers, orange: outliers, (a) and (b) urine test dataset, (c) blue: urine in-distribution dataset, orange: samples of UTKFace [al17b] dataset 40
5.10 (a) Mahal-classifier vs (b) Mahal-ae models distance scores blue: inliers, orange: outliers 41
5.11 (a) Mahal-classifier-gaussian vs (b) Mahal-classifier-sp distance scores, blue: inliers, orange: outliers 41
5.12 Test images are sorted from low score to high score based on Mahal-classifier detector 42
5.13 Test images are sorted from low score to high score based on Mahal-ae detector 43
5.14 (a) KSD-pure, (b) KSD-classifier and (c) KSD-ae detectors on urine test set scores blue: inliers, orange: outliers 43
5.15 Proposed Mahalanobis detector with double thresholds 44