University of Nevada, Reno

Robust and Cross-domain Anomaly Detection and Mitigation

A dissertation submitted in partial fulfillment
of the requirements for the degree of Doctor of
Philosophy in Computer Science and Engineering

by

Chenxing Wang

Dr. Alireza Tavakkoli/Dissertation Advisor

May, 2024

THE GRADUATE SCHOOL

We recommend that the dissertation
prepared under our supervision by

Chenxing Wang

entitled

Robust and Cross-domain Anomaly Detection and Mitigation

be accepted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

Alireza Tavakkoli, Ph.D.
Advisor

George Bebis, Ph.D.
Committee Member

Mircea Nicolescu, Ph.D.
Committee Member

Shamik Sengupta, Ph.D.
Committee Member

Nicholas Murray, Ph.D.
Graduate School Representative

Markus Kemmelmeier, Ph.D., Dean
Graduate School

May, 2024

Abstract

Anomaly detection aims to identify unusual patterns in data that significantly diverge from normal instances. When an anomaly is detected, specific steps are undertaken to address and resolve the issue. Despite significant advancements in anomaly detection techniques in recent years, challenges such as low recall rates, extreme class imbalance, and high noise levels persist. The success of mitigation closely depends on the detection phase. As a strategy to enhance anomaly detection, developing a system that creates an ideal environment for this purpose is proposed.

In the realm of supply chain security, to counteract anomaly attacks and enhance mitigation, we suggest the '3D Unclonable Optical Identity' as a solution for product verification. This tag, designed with a distinctive 3D structure, is exceedingly difficult to replicate, even with advanced fabrication methods. The security of this ID rests on the difficulty of duplication, rather than on secrecy. To address issues such as class imbalance and the impact of noise in practical applications, we utilized UE4 to generate thousands of simulated images from different angles and lighting conditions. This method assists in the development of an anomaly detection system capable of identifying counterfeit tags.

Another challenge in anomaly detection is identifying anomalies across various data types, necessitating a versatile, data-agnostic method for representing typical samples. Because of these challenges, specific models are frequently developed for different anomaly detection applications. A possible solution is to employ a single model to detect diverse types of anomalies. The generative model, especially the diffusion model, has attracted attention due to its ability to create high-quality images and potential in enhancing anomaly detection. We propose a latent diffusion-based multi-class anomaly detection model. This model learns latent representations of

non-anomalous samples and is capable of detecting anomalies in multiple classes. Our extensive evaluations on benchmark datasets such as MNIST and CIFAR-10 have shown that our approach outperforms current state-of-the-art methods in latent diffusion-based anomaly detection.

Anomaly detection in the biomedical imaging field presents unique challenges, chiefly in accurately segmenting anomaly areas and quantifying anomaly behaviors. Gould Syndrome, a rare genetic multi-system disorder, is one such case. We have developed a Gould Syndrome Detection pipeline to detect gene changes based on vascular SMC phenotype. Additionally, calcium imaging, a crucial regulatory mechanism for cerebral blood flow, is addressed in our work. We have created SEANVC (Simple Semi-automated Analytical Tool for Astrocyte Ca2+ Signals and Vascular Responses in Neurovascular Coupling) to assist researchers in identifying anomalous Ca2+ signals and their corresponding vascular responses.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep thanks to my advisor, Professor Alireza Tavakkoli. Alireza has guided me throughout my entire Ph.D. journey and walked me through every step of becoming an independent researcher. I am particularly appreciative of his consistent encouragement and support. He has always supported my pursuit of innovative ideas and encouraged me to follow my passions.

Next, it is my pleasure to have Professor Alireza Tavakkoli, Professor George Bebis, Professor Mircea Nicolescu, Professor Shamik Sengupta and Professor Nicholas Murray serve on my thesis committee. I would like to thank them for their kind suggestions, feedback, and support of this dissertation.

I would also like to thank all my brilliant collaborators, including Haoting Shen, Yifei Jin, and Cam Ha Thai Tran. The discussions with my collaborators about ideas and experiments are one of the key parts of my research. This thesis could not be complete without the kind assistance they all provided to me.

A very special thanks to my father Qinghua Wang, my mother Xiurong Liang and my wife Yunhui Long, for their continuous unconditional love and support. Their love gives me the confidence and courage to chase my dreams.

Finally, I am thankful to the University of Nevada, Reno, the National Science Foundation, and the National Institutes of Health for providing funding and resources in support of my PhD study. This dissertation was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health under grant number P30 GM145646 and by the National Science Foundation under grant number 2201599.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

Chapter I

Introduction

1.1 Overview

The field of anomaly detection has garnered increasing interest across various domains due to the application of advanced techniques. Recent advancements in deep learning have enhanced the representation of complex data, significantly benefiting anomaly detection in handling high-dimensional, graph, or spatial data. These developments have led to partial or complete solutions to numerous application challenges in anomaly detection across sectors like medical data analysis, risk management, and AI safety.

Unlike many deep learning tasks, anomaly detection often operates without data labels, complicating the identification of anomalies as labeling information, when available, is generally insufficient to encompass all anomalous scenarios. Traditional methods such as PCA and nearest neighbor algorithms often fall short in these applications. Recently, the focus has shifted towards deep learning-based semi-supervised and unsupervised algorithms.

This thesis introduces a novel anomaly detection and mitigation pipeline that learns the latent features of cross-domain knowledge, enhancing the robustness of

anomaly detection across different domains. We have implemented this pipeline in three specific applications 1)3D Unclonable Optical Identity for Universal Product Verification [4] 2) Latent Diffusion based Multi-class Anomaly Detection [5] 3)Anomaly detection in the biomedical imaging field.

1.2 3D Unclonable Optical Identity for Universal Product Verification

Reliable identification (ID) is essential for improving the global supply chain by aiding stakeholders in detecting issues like IP theft, counterfeiting, and mishandling. For daily commercial use, IDs must be securely attached to the product or its original packaging, cost-effective for less expensive goods, and user-friendly for verification purposes.

We propose a novel type of ID that is irreproducible, reliable, and applicable to a wide range of products, including electronics and high-value items. This ID, in the form of a sheet containing randomly distributed micro-bubbles, utilizes the 3D spatial locations of these particles as its unique feature. These irreproducible features, introduced unintentionally during fabrication, help keep manufacturing costs low. Moreover, reproducing a specific ID focuses on replicating the characteristic 3D features rather than confidential aspects, eliminating the need for a secretive product database.

1.3 Latent Diffusion based Multi-class Anomaly Detection

The anomaly detection [6, 7] landscape has evolved significantly with the rise of deep learning, improving feature representations for various data types. Traditionally, anomaly detection algorithms have focused on one-class scenarios where the model learns a probability density function for a single class, treating any deviation as an anomaly. Our approach extends this to multi-class anomaly detection, where the model learns to identify boundaries across various normal object classes without access to category labels during training or inference.

In this work, we developed a multi-class anomaly detection framework using the Latent Diffusion Model (LDM) within the Denoising Diffusion Probabilistic Model (DDPM) framework. This model uses the generative capabilities of the diffusion model to ascertain the congruence between input and reconstructed images, determining their category alignment.

1.4 Anomaly detection in the biomedical imaging field

Pixel-wise segmentation in medical imaging is labor-intensive and requires precise localization by clinical experts for accurate diagnosis. Despite achieving human-level performance in general and medical image segmentation, convolution and transformer-based architectures struggle with overlapping objects, often resulting in low confidence in pixel-wise accuracy.

To address this, we propose an architecture that integrates and extracts manifold features from both low-resolution images for contour-like object segmentation and high-resolution images for differentiating boundaries of overlapping objects. This ar-

chitecture utilizes a multi-objective function with a distance-based loss to enhance overall model confidence in pixel-wise segmentation. We applied this technique to segment various anatomical structures in different imaging modalities with high confidence.

1.4.1 Gould Syndrome Detection

Gould Syndrome, a rare multi-system genetic disorder, is characterized by a range of abnormalities, including those affecting the brain, eyes, muscles, and kidneys. Emerging evidence suggests a broader spectrum of associated abnormalities. Our study focused on detecting the proportion of smooth muscle cells with irregular textures using image segmentation, which assigns one of three labels to each pixel: background, normal cell, or abnormal cell. This segmentation allows us to calculate the percentage of anomalous cells, distinguishing our approach from traditional segmentation that typically segments based on different object types rather than textural characteristics.

1.4.2 Simple Semi-automated Analytical Tool for Astrocyte Ca2+ Signals and Vascular Responses in Neurovascular Coupling

Understanding neurovascular coupling (NVC) is crucial as it underpins diagnostic techniques like fMRI and PET scans, providing insights into the relationships among neurons, astrocytes, and vascular cells. The rapid advancements in imaging and analytical technologies have significantly improved our capability to analyze astrocyte Ca2+ dynamics and vascular responses in vivo. However, existing analytical tools lag behind these technological advancements, often failing to analyze both aspects simultaneously in an efficient and accurate manner. Our project developed a

semi-automated tool that streamlines the analysis of Ca2+ dynamics and vasomotor responses, enhancing throughput while retaining accuracy.

Chapter II

Literature Review

2.1 Problem Definition

2.1.1 Natural and Anomalous Properties

The nature of anomaly detection is to identify anomalous samples in data based on a predefined concept of normality. A straightforward approach to this problem is to define normality based on the given data and then classify any data that does not conform to this normality as anomalous. However, unlike many other tasks that focus on majority events, anomaly detection targets minority events, leading to numerous unique and challenging issues.

- **Boundary Challenges.** Defining normality from a given dataset is challenging because it is impossible to showcase all possible normal cases within a finite dataset. Consequently, the boundary between normal and abnormal data often remains obscure. Additionally, manually labeled data can be incorrect or noisy, further complicating the clarity of this boundary.
- **Unknown Anomalies.** Typically, we have very little knowledge about the

characteristics of anomalies. They may differ from normal data in various aspects, such as the structure of the data, the data distribution, or even attack behaviors on the data. Some anomalies may remain unknown until they are discovered, such as novel network attacks or rare data types.

- **Class Imbalance.** Anomalies occur with extremely low frequency compared to normal data instances, resulting in highly imbalanced datasets. In most cases, it is impossible to obtain a large amount of labeled anomaly data.

2.1.2 Real-World Challenges

The subsection already highlights three problems stemming from the nature of anomalies. However, in real-world applications, one encounters additional challenges. Some of these challenges are intrinsic to the properties of anomaly detection, while others relate to the specific tasks that anomaly detection addresses.

- **High Dimensional Anomaly Detection.** On one hand, high-dimensional data suffers from the curse of dimensionality, which can lead to the problematic concentration of distances between normal and anomalous data. On the other hand, abnormal characteristics often manifest in low-dimensional latent spaces, while remaining hidden in the original high-dimensional space. A straightforward solution is to perform anomaly detection in a low-dimensional space extracted from the original high-dimensional space. The challenge then becomes how to identify representation features capable of capturing meaningful information from both the data and the task. This is extremely important and poses a significant challenge due to the unknown nature of anomalies.

- **Lack of Anomaly Labels.** The primary purpose of anomaly detection is to

identify changes in behaviors or object appearances that are extremely subtle compared to their normal counterparts. In most real-world applications, it is very difficult or even impossible to collect a sufficiently large number of labeled anomaly data samples. For example, in network security applications, while normal traffic is abundantly available, there are nearly infinite patterns of traffic that are anomalous, and it is impossible to label or even observe all anomalous traffic patterns. Given the limitations in the availability of anomalous data, fully supervised models are impractical for real-world applications. Consequently, much recent research has focused on unsupervised approaches to anomaly detection. The main difficulty with unsupervised methods is that models rely solely on the assumptions of what constitutes an anomaly. As discussed in the previous section, these anomalies may remain unknown, meaning that in many cases, assumptions may not align perfectly with the data. Semi-supervised anomaly detection, which trains models using only a small amount of anomaly data, has become another focus of recent research. The main challenges now relate to how to use small amounts of data to learn features representative of other anomalies, especially those with different structures.

- **Malicious Actions.** One major application area for anomaly detection is identifying malicious actions, ranging from traditional cyber intrusion detection and fraud detection to the recent challenge of deep fake detection. Anomaly detection algorithms must continually evolve to keep pace with attackers. As a result of this ongoing competition, anomalous behaviors keep evolving, particularly when attackers are aware that anomaly detection algorithms are being used. In such cases, attackers can quickly adapt to the algorithms, modifying their malicious behavior to make it appear normal, thus presenting a significant challenge to maintaining effective security measures.

- **Low Recall Rate.** Due to their rarity and diverse nature, anomalies are challenging to identify in real-world applications. To address the extreme imbalance between normal and anomalous instances, many current algorithms attempt to balance labels during preprocessing procedures. However, this approach can lead to the incorrect classification of many normal instances as anomalous. Despite numerous improvements to anomaly detection methods in recent years, a low recall rate remains a major challenge in this field. This issue is particularly critical in unsupervised methods, where establishing a clear boundary between normal and abnormal instances is quite difficult.

- **Noisy Labels** Dealing with incorrect labels presents another significant challenge in anomaly detection. The boundary between normal instances and anomalies is often blurry in many applications, leading to a strong possibility that even expertly labeled data could contain incorrect and erroneous labels. For instance, in medical image analysis, many anomalous cells are difficult to detect by human eyes, including those of medical experts. The most intuitive approach to addressing these issues is to employ unsupervised models, which do not rely on labeled data and therefore circumvent the problems associated with inaccurate labeling.

2.2 Density Estimation

All density-based methods for anomaly detection operate on the assumption that normal data follows a specific distribution. Given a known data distribution and a training dataset composed of normal instances, these methods calculate the likelihood of new test data fitting this distribution. Under this assumption, anomalous data should exhibit a lower likelihood compared to normal data, as they deviate from the

expected distribution patterns. This foundational assumption allows these methods to effectively identify discrepancies that indicate anomalous behavior.

2.2.1 Parametric density estimation

The most classic method of density estimation is parametric density estimation, where the density of the data can be expressed as a function of certain parameters. The density function can be represented as $f(\theta, x)$, where x represents an observation and θ denotes the parameters estimated from the given data. This approach assumes that the data adheres to a known distribution type, such as normal, exponential, or Poisson, and the task involves estimating the parameters of this distribution based on the observed data.

2.2.1.1 Gaussian Distribution

The most basic distribution assumption is multivariate Gaussian distribution. Gaussian distribution uses two parameters to describe the distribution, μ to represent the distribution mean and σ to imply the standard deviation. Different methods uses various ways to calculate distance between test data instance and mean value and set up threshold detecting anomalies.

The most intuitive way is based on [8], it implies distance between normal data and distribution mean should be inside 3σ. Because under Gaussian distribution, $\mu \pm 3\sigma$ contains 99.7% of all data instances. Following this idea, people begin to use the box plot technique to detect the anomalies. The classic version of box plot in [9] shows the usage of lower quartile(Q_1) and higher quartile(Q_3). Based on the definition, all normal data should located between $Q_1 - k(Q_3 - Q_1)$ and $Q_3 + k(Q_3 - Q_1)$, where

k is a customized parameter always choose as 1.5. Fig2.1 from [10] clearly shows the structure of a box plot.

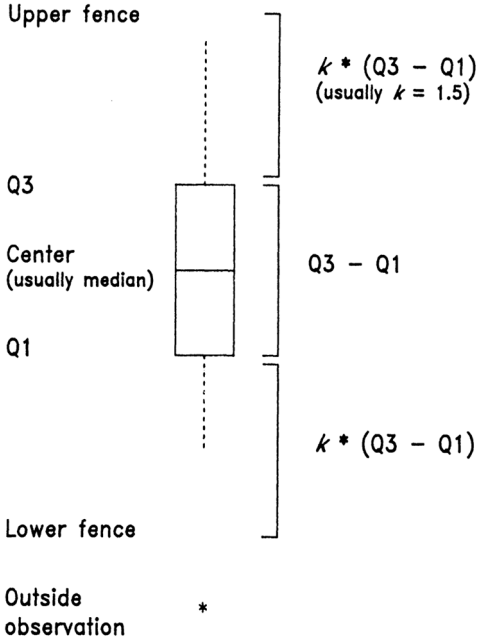


Figure 2.1: Box Plot Structure

Another approach introduced by [11] is based on the Mahalanobis distance between the test sample and the expectation of training samples. The definition of Mahalanobis distance shows as follows:

$$D_M(x) = \sqrt{(x-\mu)^\top S^{-1}(x-\mu)} \tag{2.1}$$

where x is test data, μ indicates the mean of training samples and S is the covariance matrix. We can notice that calculating the Mahalanobis distance is equal to estimate the parameters of multivariate Gaussian distribution based on training data and evaluating the log-likelihood of a test point according to the estimated distribution. Compared to modeling each dimension of the data independently, fitting a multivariate Gaussian captures linear interactions between pairs of dimensions.

2.2.1.2 Mixture Distribution

The assumption of Gaussian distribution may not always show the correct situation in the real world application. Thus, many people start to focus on the mixture distribution problem. The mixture distribution problem has two different cases. In the first case, we assume the normal data and abnormal data are following different distributions. The second case, however, assumes normal data itself is a mixture of two different distributions.

In the first case, we usually use distribution M to representation the distribution of normal data and A to indicates the distribution of anomalous data. Under the assumption, each data instance fall in distribution A with distribution λ and fall in distribution M with distribution $1 - \lambda$. Thus, the generative distribution of all the generative data D, can be written as

$$D = \lambda A + (1 - \lambda)M \tag{2.2}$$

Then the problem can be described as given a dataset generated by the distribution D, figure out the data generated form distribution A.

In [12], the author proposed an algorithm based on the measurement of log likelihood change with or without each element. In the initial status, it assumes all the data belongs to distribution M. At each time, we remove one element from distribution M and assume it belongs to distribution A and calculate the log likelihood. The log likelihood defines at time t is

$$\begin{aligned} LL_t(\mathbf{D}) = |M_t| \log(1 - \lambda) + \textstyle\sum_{x_i \in M_t} \log\left(P_{M_t}\left(x_i\right)\right) \\ + |A_t| \log \lambda + \textstyle\sum_{x_j \in A_t} \log\left(P_{A_t}\left(x_j\right)\right) \end{aligned} \tag{2.3}$$

The change from time $t-1$ to time t is to remove an element x_t from group M and add it into group A and calculate the log likelihood again. Where we can show as

$$M_t = M_{t-1} - \{x_t\} \tag{2.4}$$

$$A_t = A_{t-1} \cup \{x_t\} \tag{2.5}$$

Then we can calculate the difference between $LL_{t-1}(D)$ and $LL_t(D)$, if the difference is larger than some threshold, then we can permanently set x_t into group A. Otherwise, we can put x_t into group M and test for another data instance. When we go though all the elements in the dataset, we can finally get two groups of data A and data M.

In the second case, normal data belongs to the mixture of different distributions. Gaussian mixture models are frequently used under this situation [13]. EM algorithm is used to estimate the parameters of parameters of distributions. Once any test data found not belong to any estimated models, then it can be considered as anomalies.

2.2.2 Nonparametric Density Estimation

In real world application, real distribution is always hard to model by any predefined model. In that case, nonparametric statistical models are applied. Nonparmatric statistical models have less assumptions compared to parametric models and they don't define any prior.

2.2.2.1 Histogram Based Model

The most intuitive way to model data nonparametrically is through the use of histograms, a method widely recognized in anomaly detection literature [12, 14]. Histogram based anomaly detection typically involves two steps. In the training step, a histogram is constructed using the feature values from the training data. During the testing step, the algorithm determines whether any test instance belongs to one of the histogram bins. If a data instance lies within any of the bins, it is considered normal. Otherwise, the test data is classified as anomalous.

However, a significant challenge with histogram-based models is determining the appropriate bin size. If the bins are too small, test instances are more likely to fall into empty or rarely occupied bins, potentially leading to a high false positive rate. Conversely, if the bins are too large, both normal and anomalous test instances may fall into frequently occupied bins, resulting in missed detections of anomalous data. In many real-world applications, finding an optimal bin size that balances the false positive and false negative rates is a challenging problem.

Histogram usage can be extended to multivariate data [15, 16]. The basic idea involves creating attribute-wise histograms. After training histograms based on different attributes, the testing process involves obtaining an anomaly score for each attribute based on the height of the bin containing the attribute value. Various algorithms can then aggregate these individual scores to compute an overall anomaly score, enhancing the model's ability to detect anomalies in complex datasets.

2.2.2.2 Kernel density estimation

Another non-parametric technique is kernel density estimation, which uses a kernel function to replace the discrete histogram in a continuous way. This idea of non-parametric density estimation comes from parzen windows estimation [17]. If we assume $p(x)$ is the density function to be estimated, then with the dataset $\{x_1, x_2, ..., x_n\}$ generated by $p(x)$, the density function estimated by this n data can be expressed as

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \delta_n \left(x - x_i\right) \tag{2.6}$$

where δ_n is a kernel function. The standard kernel density estimation, along with a more recent adaptation that can deal with modest levels of outliers in the training data.

2.2.3 Energy Based Models

Energy based models are generative models which use energy function to express the probability density of variables. The energy function $E_\theta(x)$ can be expressed as

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp\left(-E_\theta(x)\right) \tag{2.7}$$

where $Z(\theta) = \int \exp\left(-E_\theta(x)\right) \mathrm{d}$ is the partition function which ensures the sum of p_θ equals to 1 [18]. The training process of original energy based models are computationally expensive, since the gradient descent in optimize process is based on Markov chain Monte Carlo(MCMC) [19]. In order to solve this problem, score matching method [20] and stochastic gradient Langevin dynamics [21] are introduced. In energy based models, E_θ is always used as an anomaly score for the reason that it is

monotonically decreasing as the density p_θ increases.

Deep belief networks [22] and deep Boltzmann machines [23] are two deep energy based models have been introduced. These two algorithms model the training data based on both input x and latent state z which can be written as $E_\theta(x, z)$. Compared to the traditional energy based models which only based input x, latent state can catch latent probabilistic dependencies in data distributions. Research in [18] using deterministic latent layers to replace the probabilistic latent layers which could evaluate $E_\theta(x)$ and use it as anomaly score in anomaly detection tasks.

2.3 Feature Extraction

2.3.1 Deep Learning Based

Deep learning based anomaly detection models aim to use different deep learning structure to extract useful low dimensional features from high dimensional data. Under this category, deep learning models are only used as a tool for feature extraction. Anomaly scores are decided after the feature extraction. In other words, feature extraction and anomaly scoring are two independent steps. We can represent the feature extraction step as

$$z = \phi(x; \Theta) \tag{2.8}$$

where $\phi : X \mapsto Z$ is a deep neural network based feature extraction function, with $X \in R^D, Z \in R^K$ and normally $D \gg K$. Then another function f works on Z is designed to assign anomaly score from feature space. Because f and θ are not trained at the same time, there is connection between these two functions.

Traditional dimension reduction methods like principal component analysis [24]

and random projection [25] are also been used as the feature extraction step in anomaly detection. However, deep neural networks have been showing a much better capability in extracting features and non-linear feature relations [26].

One intuitive way of using neural networks in feature extraction is to directly use popular pre-trained deep learning models to extract low dimensional features. With the development of deep learning, models we can choose like VGG [27], ResNet [28] can have a pretty good feature extraction results. In addition, feature representations pre-trained on one dataset can usually be transferred to a anomaly detectors on another dataset. As we can see in [29], One-class support vector machines(SVM) can be initialized with VGG models pre-trained on ILSVRC [30] then fine-tune on MNIST data [31] in order to improve the anomaly detection rate. Also, [32] shows ResNet models pre-trained on MNIST can improve the anomaly detection rate in video surveillance datasets.

Instead of using a pre-trained deep neural network, train a unique feature extraction model is another way. For example in [33], three autoencoder networks are built in order to get feature of appearance, motion, and joint information of appearance and motion in a anomaly detection task on video data.

Using a deep neural network as feature extraction in anomaly detection has many advantages. First of all, we can choose a feature extraction model from different state of art deep neural network models. Also, the implementation process is not hard due to the public availability of all deep neural networks. In addition, feature extraction works pretty well in some applications. However, the input data type is limited and decided by the extraction model. More importantly, an optimal anomaly score is hard to get because of the total separation of feature extraction and anomaly scoring.

2.3.2 Reconstruction Models

The basic idea of reconstruction methods is to learn a model which optimized to reconstruct all normal data instance and detection the anomalous data instance by high reconstruction error.

The objective function of reconstruction models can be shown as $\phi(\theta) : X \to X$ which is a feature mapping from data to itself. It includes two steps, the encoding step

$$z = \phi_e\left(x; \theta_e\right) \tag{2.9}$$

and the decoding step

$$\hat{x} = \phi_d\left(z; \theta_d\right) \tag{2.10}$$

where we can see that θ is the union of θ_e and θ_d and z is the latent representation of input data x. The propose of the reconstruction model is to train the model and force the input and output to be the same which means

$$x = \hat{x} = \phi_d\left(\phi_e\left(x; \theta_e\right); \theta_d\right) \tag{2.11}$$

Thus we can get the θ based on

$$\{\theta_e^*, \theta_d^*\} = \arg\min_{\theta_e, \theta_d} \sum_{x \in X} \left\| x - \phi_d\left(\phi_e\left(x; \theta_e\right); \theta_d\right) \right\|^2 \tag{2.12}$$

then the reconstruction error can be defined as

$$s_x = \left\| x - \phi_d\left(\phi_e\left(x; \theta_e^*\right); \theta_d^*\right) \right\|^2 \tag{2.13}$$

If no restrictions added in the model, the optimal function the model learn would

be $\phi =$ identity. This model is absolutely not we want because nothing is learned in the training process. Thus some restrictions of the model is required in the training process.

The first assumption is based on the latent space Z. It assumes that data on some lower dimensional space is embedded within the data space X, with dimensional of latent space smaller than data space. For example, if the input data is images in pixels space, the latent space should capture information such as structure of scenes, shape, size, texture and so on. This assumption makes sure that a low dimensional latent space Z exists, which we can get $x = \phi_d(\phi_e(x))$.

Another assumption is based on the prototype. It assumes there exists a finite number of elements in input data space X that correctly describe the data. This prototype assumption is also common in clustering and classification when we assume a collection of prototypical instances represent clusters or classes well.

Autoencoder networks is the most frequent used algorithm in reconstruction models. It uses various type of neural networks to encode the input data and than decode to recover it. Autoencoder is originally used for dimension reduction [34, 35]. While nowadays, it becomes the most popular algorithm used in anomalous detection [36–38]. reconstruction loss function is used to learn the parameter of both networks. In order to representation the low dimensional feature space, a bottleneck network is always used and can be seen in Fig2.2

In order to minimize the reconstruction error and detect for anomolies data, feature extracted in latent space should be highly relevant to normal data instances. Only in this way, the reconstruction error of anomalous data instance will be much higher than the reconstruction error of normal data. Then the reconstruction error can be used as anomaly score.

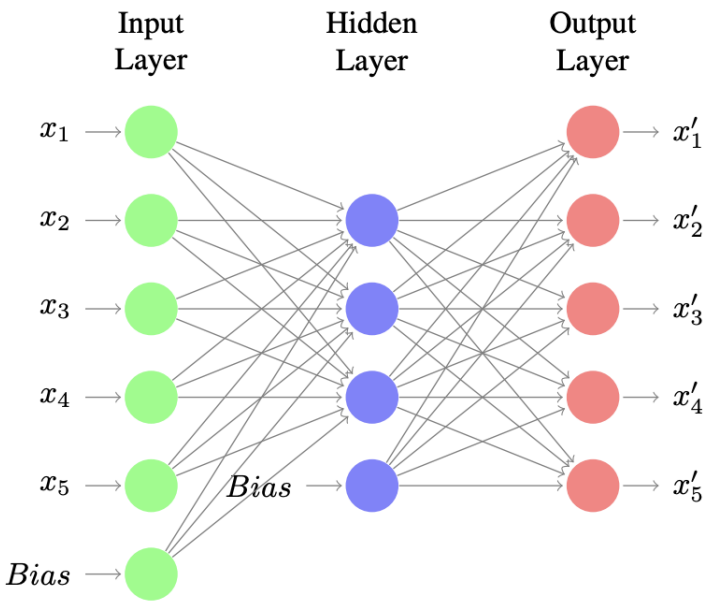


Figure 2.2: Autoencoder Architecture from [1]

Some innovative types of autoencoders have been developed to improve the feature representations. Denoising autoencoder [39] is designed to train data on corrupted data instance which required to be robust against some small variations. Sparse autoencoder [40] aims to increase the sparsity in the hidden layer where only top K most active units is kept. Contractive autoencoder [41] proposed to robust against small variations around neighbours.

The biggest advantage of reconstruction models is the straightforward detection idea. Also, autoencoder can be used in different types of data instances. Also with the development of autoencoders, different types of strong autoencoders have been introduced in recent years. Though autoencoder is a popular algorithm in anomalous detection for so many years, it still has some drawbacks. The most intuitive drawback is the requirement of a clean training dataset. If anomalies accidentally show up in training data, the feature space will be affected. In addition, the autoencoder is originally designed for dimensional reduction, the output representation is actually a

summary of regularities instead of finding the abnormal feature.

2.4 Feature Learning

2.4.1 Generative Adversarial Networks

Generative adversarial network(GAN)-based anomaly detection growth quickly after it first show up in [2]. The general idea is to learn a latent feature space of a generative network G so that the latent space well captures the normality underlying the given data. Then the anomaly score can be defined as the residual between real instance and generated instance.

AnoGAN [2] is an example of the first usage of GAN in anomaly detection. Similar with the autoencoder, the training process is only focus on normal data instance. The main idea is that given input data instances x, try to find out z in the latent feature space of the generative network G in order to make $G(z)$ and x as similar as possible. The training of GAN in only normal data will let the generator learn the underlying distribution of normal data. Once an anomalous image is encoded, the reconstruction result will be a normal image generated by G. The difference between input image and reconstruction image will show the anomalous area. The structure of AnoGAN can be seen from Fig2.3

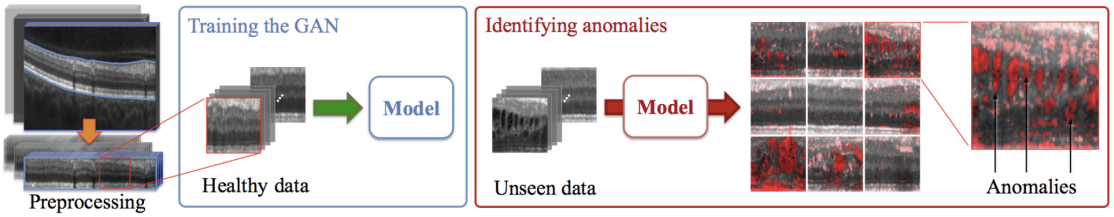


Figure 2.3: AnoGAN [2] training and anomalous detection structure

A traditional GAN object function can be written as

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_X}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_Z}[\log(1 - D(G(\mathbf{z})))] \tag{2.14}$$

where G represent generator and D indicates the discriminator. The parameters of generator and discriminator are defined as θ_G and θ_D respectively. V represent the value of object function above.

According to the algorithm, the mapping function from input samples to the latent space is trained in a iterative way. The propose of this process is for each query data sample x, find out z in latent space which makes $G(z)$ similar to x.

Two loss functions are used in order to find out the best latent value z for each x: residual loss and discrimination loss. The residual loss is used to measure the difference between generated samples and query samples in input domain.It can be written as

$$\ell_R\left(x, z_\gamma\right) = \left\|x - G\left(z_\gamma\right)\right\|_1 \tag{2.15}$$

while the discrimination loss is used to measure the discriminator response and it is defined as

$$\ell_D\left(x, z_\gamma\right) = \left\|h(x) - h\left(G\left(z_\gamma\right)\right)\right\|_1 \tag{2.16}$$

where γ is the iterative search index in latent space and h is a feature mapping. The overall loss function based on two loss values is defined as

$$\ell(z_\gamma) = (1 - \alpha)\ell_R + \alpha\ell_D \tag{2.17}$$

AnnoGAN is the first paper shows that GAN can be used in anomaly detection and at the same it introduce a new method mapping latent space to input data.

However, the biggest issue of AnoGAN is the computational difficulty due to the iterative search of z for every new input x.

In order to solve the problem, one intuitive way is to learn another mapping from data space to latent space instead of only training the one way mapping. EBGAN [42] first introduced BiGAN architecture in anomaly detection based on the idea from [43]. The basic idea is to build up an encoder E which map the input data x into latent feature z. In the training process G, D and E are training in a iterative way. BiGAN uses data instance $(x, G(x))$ and $(G(z), z)$ to replace the x and $G(z)$ in AnoGAN, the objective function can be written as

$$\min_{G,E}\max_{D} V(D,G) = E_{x\sim p_X}[E_{x\sim p_E(\cdot|x)}\log D(x,z)]+E_{z\sim p_Z}[E_{x\sim p_G(\cdot|x)}\log(1-D(G(x,z)))] \tag{2.18}$$

The anomaly score defined in EBGAN is similar with the definition of AnnoGAN:

$$\ell_G(x) = \|x - G(E(x))\|_1 \tag{2.19}$$

$$\ell_D(x) = \|h(x, E(x)) - h(G(E(x)), E(x))\|_1 \tag{2.20}$$

$$\ell(x) = (1-\alpha)\ell_G + \alpha\ell_D \tag{2.21}$$

Many other GAN based anomaly detection method have been introduced based on different GAN architectures. For example, f-AnoGAN [2] which uses Wasserstein GAN [44] to replace the standard GAN in anomaly detection.

GAN-based anomaly detection algorithms becomes popular in recent years with the advantage of strong capability in generating realistic data instances. Thus the

abnormal data instance which hard to reconstruct from the latent space will be easier to detect. In addition, GAN-based anomaly detection easily benefits from the fast development of GAN-based models. However, it still has many drawbacks based on the natural property of GAN. The most serious one is the difficulty of convergence in training the GAN-based model. Also, a generator may learn manifold different from the normal data instance especially when anomalous data unexpectedly appear in the training dataset.

2.4.2 Diffusion models based anomaly detection

The basic design of diffusion models are based on two Markov chains. Given any data $x_0 \sim q(x_0)$, the first Markov chain is called the forward chain, which transfer the data into noise. Standard Gaussian noise is typical choice when using the diffusion model because of its unique properties. The forward Markov chain uses T steps, with Gaussian noise added into the data for each step.

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right) \tag{2.22}$$

where $t = 1, 2, ..., T$ and $\beta \in [0, 1]$ denotes the noise variance schedule. From the equation above, given data x_0 and step t, we can get the distribution of a noise image

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t) I\right) \tag{2.23}$$

where here we use $\bar{\alpha}_t$ represent $\prod_{s=1}^{t}(1-\beta_s)$

The other Markov chain represents the reverse process, which begins from the standard Gaussian noise image and keeps adding small amount of noise in order to

recover the input data. This process begins at the point

$$p\left(x_T\right) = \mathcal{N}\left(x_T; 0, I\right) \tag{2.24}$$

And small amount of Gaussian noise will be added onto the image step by step.

$$p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right) \tag{2.25}$$

where μ_θ and Σ_θ are the mean value and standard variation of the Gaussian noise added in each step. In order to reverse the forward process, we set $\Sigma_\theta\left(x_t, t\right) = \beta_t I$ and μ_θ should estimate $\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$, thus we can set

$$\mu_\theta\left(x_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right) \tag{2.26}$$

In order to estimate $\epsilon_\theta\left(x_t, t\right)$, a U-net is built to minimize the objective function

$$L = E_{t\sim[1-T], x_0\sim q(x_0), \epsilon\sim N(0,I)}\left[\left\|\epsilon - \epsilon_\theta\left(x_t, t\right)\right\|^2\right] \tag{2.27}$$

where $\epsilon \sim \mathcal{N}(0, I)$. From equation above, the U-net model is trained so that, given any input x_t, the output of the U-net model should be equal to $\mathcal{N}(0, I)$ In the inference process we can get

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right) + \beta_t z \tag{2.28}$$

2.4.3 One Class Classification

One-class classification [45, 46] refers to the scenario where an algorithm learns to describe a set of training data and then determines whether incoming test data belongs

to this trained set. Unlike traditional methods that estimate the density of normal data, one-class classification-based anomaly detection algorithms directly learn the decision boundary. This approach focuses on defining a region that encapsulates what is considered 'normal', and any data point that falls outside this region is flagged as an anomaly.

One-class classification is a specialized type of classification problem where only one class is present during the training process. The primary objective in one-class classification is to minimize false detections on normal data instances and miss detections on anomalous instances.

To reduce false detections on normal data, one might consider drawing a larger boundary that encompasses all the data. However, to minimize miss detections on anomalous data, the boundary must be tight enough. Typically, to address this trade-off, a prior indicating the false alarm rate $\alpha \in [0, 1]$, is specified. Under this constraint, the goal is to minimize the miss rate of anomalous data instances. The challenge then shifts to estimating the boundary under a specific α-density level.

The concept of one-class classification originated with the support vector machine (SVM) approach. A popular kernel-based one-class classification method is the Support Vector Data Description (SVDD) [47]. Assuming a kernel function k with an associated feature space $\mathcal{F}$ and a feature mapping ϕ we can define:

$$k(x, x') =< \phi(x), \phi(x') > \tag{2.29}$$

The goal of SVDD is to find a hyperplane that encompasses the data with minimal volume.

The integration of deep learning into one-class classification, such as in deep neural

one-class classification [48], aims to maximize the distance between the training data and the origin. Instead of using the high-dimensional input space, the hyperplane is learned from the low-dimensional feature space extracted by deep neural networks.

The benefits of this approach include that deep neural networks can capture more useful information from the feature space while simultaneously reducing computational complexity, which might otherwise be high in kernel function calculations.

One advantage of one-class classification in anomaly detection is its well-established methodology and the availability of various kernel functions. Additionally, one-class classification models can be integrated with deep representations to learn better data representations. However, the learning process may be inefficient if the normal data distribution is complex, posing challenges in effectively training the model.

2.4.4 Positive-Unlabeled Learning

Positive-unlabeled learning aims to distinguish between positive and negative data under circumstances where each piece of unlabeled training data could belong to either category. The use of positive-unlabeled learning in anomaly detection primarily focuses on a semi-supervised setting where both normal data and unlabeled data are available [49, 50].

Two main methods are utilized in positive-unlabeled learning. The first method involves selecting reliable negative samples from the unlabeled data, thereby transforming the problem into a traditional supervised anomaly detection problem. The second approach operates under the assumption that the entire unlabeled dataset consists of negatives, albeit noisy.

The most challenging part of the first method is identifying reliable negatives

from the unlabeled data. This challenge can be addressed by calculating the distance between the unlabeled data and the positive data. In [51], the distance between each data instance x_i and the positive dataset P is defined as

$$d(x_i, P) = \min_{x_j \in P} \|x_i - x_j\| \tag{2.30}$$

The negative dataset N is then selected from unlabeled dataset U based on maximisation of distance between N and P where

$$\max_{N \subset U} d(N, P), d(N, P) = \sum_{x \in N} d(x, P) \tag{2.31}$$

Additionally, clustering methods [52] and density-based methods [53] are also used to more effectively filter out the reliable negatives. In the second approach, the challenge lies in dealing with noisy negative data. Both label cleaning methods [54] and sample re-weighting [55] have shown good results. A particularly interesting idea is based on reconstruction methods [56].

2.5 Measure based methods

Measure based methods refer to algorithms which learn feature representation based on one specific anomaly measurement. The objective function can be written as

$$\{\Theta^*, W^*\} = \arg\min_{\Theta, w} \sum_{x \in X} \ell(f(\phi(x; \Theta); W)) \tag{2.32}$$

$$s(x) = f\left(\phi\left(x; \Theta^*\right); W^*\right) \tag{2.33}$$

where f is a given anomaly score function applying to the latent space. Based on the anomaly scoring function, f may include parameter of W. Different from other algorithms we discussed above, f is a given fixed function, algorithms under this category only focus on finding a feature representation function specific to anomaly scoring function f.

2.5.1 Distance-based Measure

Distance-based anomaly detection algorithms based on optimized feature space based on distance-based anomaly scoring functions. Distance-based anomaly scoring function are the most intuitive methods with the benefits of easy implementation. A large number distance-based anomaly scoring functions have been introduced, such as k-nearest neighbor distance [57], Distance-based outliers [58] and so on. The biggest challenge of directly using distance-based methods is that when facing high dimensional data, distance will no longer be a good indicator. This challenge can be well solved by reduce the dimensional of data space before applying the distance measurement.

[59] first talked about this approach. A random neighbor distanced-based anomaly scoring method is used in this work. Instead of directly applying the anomaly scoring function to the high dimensional data, they created a low dimensional feature space to let the function apply. The main idea is to create the low dimensional feature which makes the nearest neighbor distance of anomalous data significantly larger than the distance of normal data.

Assume we have dataset X and a subset S randomly sampled from X. A is used to represent the anomaly dataset and N is used to represent the normal dataset. The

loss function can be represented by

$$L = \frac{1}{|X|} \sum_{x \in A, x' \in N} \max\{0, m + f(x', S; \theta) - f(x, S; \theta)\} \tag{2.34}$$

where

$$f(x, S; \theta) = \min_{x' \in S} \|\phi(x; \theta), \phi(x'; \theta)\|_2 \tag{2.35}$$

and m is predefined constant between two distance. The random distance is directly used as anomaly score in the evaluation stage. Following the same mechanism, we can deal with other given anomaly scoring function by just replace the scoring function f given in this project.

[60] introduced a simpler idea compared to [59]. The representation learning is based on the distance between randomly projected representation and optimized representation. The objective function is written as

$$\theta^* = \arg\min_{\theta} \sum_{x \in X} f\left(\phi(x; \theta), \phi'(x)\right) \tag{2.36}$$

where ϕ is a neural network which the objective function aim to optimize and ϕ' is a random mapping function which has the same structure with ϕ but has fixed random weights. f is the distance measurement function. In the process of optimizing the objective function, model learns the underlying pattern in the data from a random neural network. However, this method ignore the relationship between data which leads to the sensitivity when facing the anomalous data.

2.5.2 Clustering-based Measure

The object of cluster-based anomaly detection is to find a feature representation which make normal data stay together and make abnormal data far away from the normal data. Because of the intrinsic similarity between cluster and anomaly detection, in many cases the result of cluster can be directly used to determine the anomaly detection result. Various type of cluster result can be used, such as distance to cluster centers [61], cluster size [62] and so on.

The challenge of using cluster-based anomaly detection is to find a suitable feature representation. The reason that we don't want to directly use cluster algorithm is the unstable cluster result based on the data. A good feature representation can make the cluster far more robust than use it on the original data. And we also need to notice that representations optimized for one specific cluster couldn't be transferred to another cluster because of the difference cluster assumption.

Two modules are often used in cluster-based anomaly detection algorithms. One is applying clustering in forward pass of the network, the other is optimize parameters based on the clustering results in backward pass. The object function can e summarized as

$$\alpha \ell_c(f(\phi(x;\theta);W), y_x) + \beta \ell_o(X) \tag{2.37}$$

where ℓ_c represent the loss of clustering function, y_x is the cluster label of data x, l_o is another loss function which enhance other constrains on learning representations. f is a cluster function with parameter W.

In the testing phase, cluster function f can be directly used to compute anomaly

score. Because of the clustering property, detecting result is very sensitive to the input training data. If unexpected anomalous data instance is included in the training process, the cluster result may be biased. Thus some constrains may added into l_c or l_o to increase the robustness against anomalous data in the training procedure.

Cluster based anomaly detection is good at find the representation feature which easily detect the anomalies and also new development cluster method may always improve the detection result. However, detection result heavily depends on the clustering result and it is sensitive to the training input data.

Chapter III

Anomaly Detection Formulation

3.1 Motivation

As discussed in the literature review of this work, there are generally two traditional viewpoints to anomaly detection: probabilistic view and algebraic view. In the probabilistic approach, the goal is to estimate the probability density function that governs the normal class of data either parametrically (MoG) or non-parametrically (KDE). In the algebraic approach, on the other hand, the objective is to find a transformation between the feature space into a latent space on which the normal class of data could be separated from the anomalous data. Both these approaches have been utilized with various degrees of success.

Shortcomings of the traditional approaches have been elaborated in the literature review section. However, two main issues need to be reiterated. First, if samples of all possible anomalous data are not available, there will be little that can be done to learn the transformation between the feature space into the latent space that can predict the actual boundaries of the normal class. Second, estimating the probability density of the normal class could be intractable, especially in the case of very high-dimensional data (e.g., images, videos or point-clouds).

Therefore, the main motivation for this dissertation is to establish the theoretical and computational foundations for allowing an implicit estimation of the probability distribution of the normal data, predicated on the relationships between the data domain, feature space, and latent distributions that govern the normal data. The success of this approach will address two fundamental problems. First, implicit estimation of the density will not require any prior knowledge about the governing probability distribution of the normal class. Second, predicating the estimated density upon the relationships between the mappings across the feature, domain, and latent spaces enforces constraints on the learning and feature representation modules to prevent them from learning identity mappings. This will have the added benefit of establishing a tighter bounds on the estimated densities to improve robustness.

3.2 Anomaly detection Structure

Given a normal class of data, $\mathcal{D}_n \in \mathbb{R}^N$, there is a latent space $\mathcal{F}_n \in \mathbb{R}^M$ that can be reached from the data space through a deterministic mapping $\mathcal{D}_n \xrightarrow{\Phi} \mathcal{F}_n$. Within this latent space the normal data class will be governed by a latent probability distribution function $p(\mathcal{D}_n|\theta)$ and the latent space will be governed by $p(\theta|\xi)$. We have three goals in mind:

1. Find a non-identity mapping ρ between the data domain back to itself through the latent space $\mathcal{F}_h$: $\mathcal{D}_n \xrightarrow{\rho} \mathcal{D}_n$.

2. Keep adding noise to latent space θ and get a latent variable $\xi \sim N(0, 1)$. Find an implicit mapping ω between ξ and $p(\theta)$: $\xi \xrightarrow{\omega'} \theta$. The whole mapping process between latent space θ can be defined as $\mathcal{F}_\theta$: $\theta \xrightarrow{\rho_\theta} \theta$.

3. Predicate the mapping ρ on ω such that both $D_{KL}\left[p(\mathcal{D}_n|\theta)||p(\mathcal{D}_n|\rho)\right]$ and

$D_{KL}\left[p(\theta|\xi)||p(\theta|\rho_\theta)\right]$ are minimized.

Accomplishing the above three goals allows us to implicitly sample the probability density of the normal class data without the need to explicitly model it, while providing a non-identity mapping between the normal class to itself, predicated on its implicit density model. Through this non-identity mapping provides a metric (goal 3 above) that can be utilized to determine whether a data sample is normal or an anomaly. Our proposal structure can be seen from Fig3.1

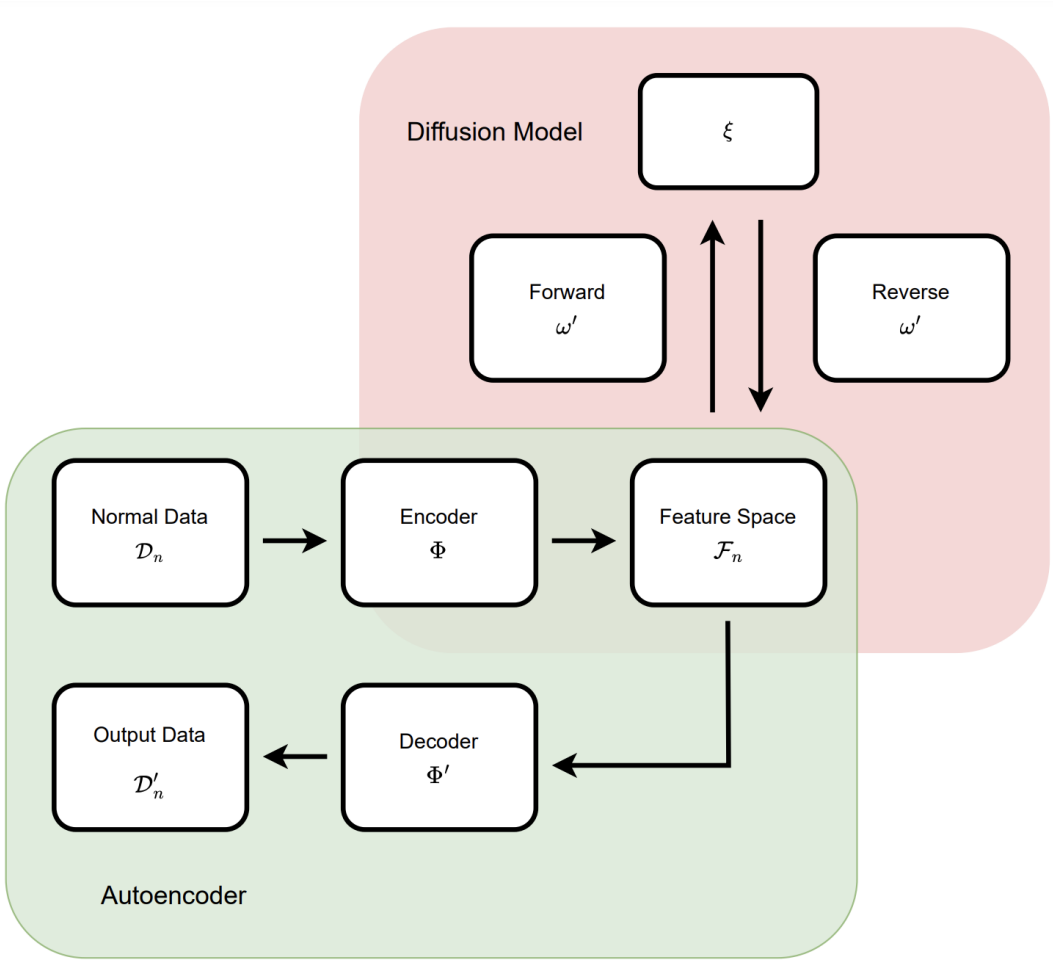


Figure 3.1: Anomaly Detection Pipeline Structure

The significant advantage of the proposed model is that it can be utilized upon any traditional regression, classification, localization, or segmentation framework to

be employed as a unified anomaly detector for various applications.

3.2.1 Data Domain to Feature Space

Given the data from the normal class $\mathcal{D}_n \in \mathbb{R}^N$, I proposed a deterministic encoder $\mathcal{E}_f$ to be designed to establish a mapping between data domain $\mathcal{D}_n$ and the feature domain $\mathcal{F}_n \in \mathbb{R}^M$.

$$\mathcal{E}_f : \mathcal{D}_n \xrightarrow[\theta]{\Phi_f} \mathcal{F}_n \tag{3.1}$$

where θ is the normal class model, and Φ_f is the deterministic encoder mapping from $\mathbb{R}^N$ to $\mathbb{R}^M$.

However, since we can't assume we know the number of classes included in the training data. This multi-cluster latent space cannot be trained via traditional auto-encoder. In order to appropriately train the classifier for anomaly detection, I plan to add a diffusion process to learn a mapping from noise image to normal class latent space. I plan to use a diffusion model that takes as input a latent random variable ξ and produces samples from the normal class domain. The diffusion is in the form of a model $\mathcal{G}_\omega$ with the loss attributed as follows:

$$\mathcal{G}_\omega : \mathcal{L} = \arg\max D_{KL} \left[p_\omega(\mathcal{X}|\xi) || p_\theta(\mathcal{X}|\rho_\theta) \right] \tag{3.2}$$

where $\mathcal{X}$ is the produced sample in the latent space θ, n is normal class distribution, and ξ is the latent random variable distribution.

To ensure that the generator is sampling from the correct implicit distribution, I plan to design an encoding discriminator $\mathcal{E}_\omega$ with a discriminative loss of:

$$\mathcal{E} : \mathcal{L} = \arg\min D_{KL} \left[p(\mathcal{F}_g|\rho) || p(\mathcal{F}_n|\theta) \right] \tag{3.3}$$

where $\mathcal{F}_g$ and $\mathcal{F}_n$ are the samples drawn from the generated data and the real data belonging to the normal class.

Chapter IV

Case Studies

4.1 Anomaly detection via diffusion model

4.1.1 Introduction

The field of anomaly detection [6, 7] has gained substantial popularity in recent years, as techniques in this domain are increasingly applied across various sectors. The advent of deep learning has significantly enhanced our capacity to represent complex data. This advancement facilitates improved feature representation for high-dimensional, graph, or spatial data in anomaly detection. Currently, we observe the utilization of anomaly detection in areas such as medical data analysis, risk management, and AI safety, to name a few. In most real-world applications, access to anomalous data is not feasible, and normal data often comprises various types of objects. For instance, in invasive species detection, access to anomaly data is limited, and the normal dataset includes different local animal species.

Many contemporary anomaly detection algorithms are designed for one-class anomaly detection. In this approach, the model is trained on samples from a particular class.

The model learns a probability density function that captures behavior for that specific class. Samples from other classes are considered anomalies, regardless of whether they belong to normal data or not. For multi-class anomaly detection, a model should learn a probability density function for all classes to delineate the boundaries of all normal data.

In this study, we aim to construct an unsupervised anomaly detection model capable of identifying anomalies across various normal object classes. Specifically, the training data consists of normal samples from several different object categories. During both training and inference processes, we do not have access to the category labels of any samples in the training data.

A commonly adopted methodology in anomaly detection involves the use of image or feature reconstruction. This approach assumes that a well-tuned model can consistently generate normal samples, even in the presence of potential anomalies in the input data. However, many widely used reconstruction networks often fail to meet the stringent requirements of this task. This failure is evidenced by an observed "identity shortcut" pattern. This shortcut leads to the direct replication of the input, potentially allowing for the accurate replication of even anomalous samples and, consequently, hampering their detection.

This challenge becomes more pronounced in contexts where the normal data distribution is inherently complex. When attempting to construct a unified model capable of reconstructing a broad range of objects, the model must endeavor to understand the joint distribution. Resorting to an "identity shortcut" might be a simpler path, but it compromises the model's effectiveness in anomaly detection.

The ever-increasing volume of digital data necessitates the development of sophisticated probabilistic models to handle inherent noise and distortion. Diffusion Denoising Probabilistic Models (DDPM), a unique category within these models, provide an innovative approach to the information bottleneck problem. Trained to systematically de-noise corrupted inputs, these models reshape the strategy for noise management. Unlike traditional models where the bottleneck is an intrinsic property, DDPM views the bottleneck as an externally adjustable feature during model inference. While previous studies have explored DDPMs as autoencoders with externally adjustable bottlenecks, none have harnessed this property for reconstruction-based anomaly detection. This paper aims to fill this void, delving into novel insights and methodologies to leverage DDPMs for enhanced anomaly detection.

In this work, we proposed a multi-class anomaly detection structure based on the LDM model. We examined the use of latent space within the DDPM framework and developed a classification model that utilizes the generative capabilities of the diffusion model. This is to determine whether the input and reconstructed image belong to the same category.

4.1.2 Method

4.1.2.1 Diffusion models

The basic design of diffusion models are based on two Markov chains. Given any data $x_0 \sim q(x_0)$, the first Markov chain is called the forward chain, which transfer the data into noise. Standard Gaussian noise is typical choice when using the diffusion model because of its unique properties. The forward Markov chain uses T steps, with

Gaussian noise added into the data for each step.

$$q\left(x_t \mid x_{t-1}\right) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right) \tag{4.1}$$

where $t = 1, 2, ..., T$ and $\beta \in [0, 1]$ denotes the noise variance schedule. From the equation above, given data x_0 and step t, we can get the distribution of a noise image

$$q\left(x_t \mid x_0\right) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, \left(1-\bar{\alpha}_t\right) I\right) \tag{4.2}$$

where here we use $\bar{\alpha}_t$ represent $\prod_{s=1}^{t}\left(1-\beta_s\right)$

The other Markov chain represents the reverse process, which begins from the standard Gaussian noise image and keeps adding small amount of noise in order to recover the input data. This process begins at the point

$$p\left(x_T\right) = \mathcal{N}\left(x_T; 0, I\right) \tag{4.3}$$

And small amount of Gaussian noise will be added onto the image step by step.

$$p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right) \tag{4.4}$$

where μ_θ and Σ_θ are the mean value and standard variation of the Gaussian noise added in each step. In order to reverse the forward process, we set $\Sigma_\theta\left(x_t, t\right) = \beta_t I$ and μ_θ should estimate $\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$, thus we can set

$$\mu_\theta\left(x_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right) \tag{4.5}$$

In order to estimate $\epsilon_\theta(x_t, t)$, a U-net is built to minimize the objective function

$$L = E_{t\sim[1-T], x_0\sim q(x_0), \epsilon\sim N(0,I)}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] \tag{4.6}$$

where $\epsilon \sim \mathcal{N}(0, I)$. From equation above, the U-net model is trained so that, given any input x_t, the output of the U-net model should be equal to $\mathcal{N}(0, I)$ In the inference process we can get

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \beta_t z \tag{4.7}$$

4.1.2.2 Architecture with Latent Diffusion models

As depicted in Fig 4.1 our multi-class anomaly detection model comprises three components: a compression model, a diffusion network, and a classification network. The compression model compresses the image into a lower-dimensional space. The diffusion network reconstructs the latent space of normal data, while the classification network determines whether the input and output of the compression model belong to the same class. An anomaly class is detected if the input and output are classified into different classes.

The input of compression model is the original image x, the compression procedure can be expressed as $z = E(x)$, and the decode procedure can be denoted as $x' = D(z)$. The architecture of our compression model, based on work [63], trains an autoencoder considering both perceptual loss and adversarial objectives. Therefore, during the image compression, it accounts for not only pixel-wise information but also the composition of image parts from a codebook constructed by the image.

Utilizing a compression model before the diffusion model in anomaly detection

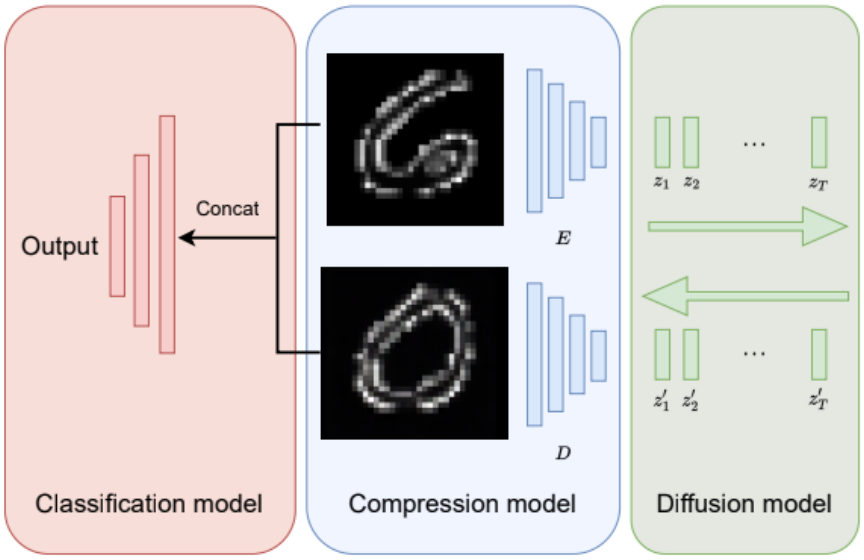


Figure 4.1: An overview of our framework, which comprises a compression model, a diffusion model, and a classification model. The compression model constructs an encoder and decoder to create a latent space. The diffusion model continuously adds noise during the forward process and estimates the input latent data in the reverse process. The classification model determines whether the input image and the reversed input image belong to the same class.

provides several benefits:

i The computational complexity during the training of the diffusion model is reduced since this model operates in the latent space.

ii The latent space prevents the DDPM reconstruction structure from encountering the "identity shortcut" issue, which arises when the network consistently produces a copy of the input data.

iii A compression model allows for flexibility in choosing an appropriate latent space for the DDPM process. Typically, both the input and output of the autoencoder should depict the original image. In this study, however, we've also experimented with transforming the autoencoder's output into an edge

label, a change that can reduce the propensity for the structure to fall into the "identity shortcut" issue.

The input of the diffusion model is the latent space vector z from compression model. Then following the forward process of the DDPM from equation 4.2. Given any time $t \in [0, T]$, the latent space z_t can be calculate by

$$z_t = z_0\sqrt{\bar{\alpha}_t} + \epsilon_t\sqrt{1 - \bar{\alpha}_t} \tag{4.8}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$.

With the t becomes larger and larger, more and more Gaussian noise is added into the image and the latent vector z_t loose its original spatial structure and looks near the Gaussian noise. In the reverse process, we can follow the equation 4.7. We need to train the U-net model $\epsilon(x, t)$ in order to let it predict the noise ϵ.

In our anomaly task, the result of DDPM reconstruction should be the same with input latent vector z if it is a latent representation from normal data. In practice, the reconstruction could keep the similarity of input if the reverse process begins from time t. As the choice of t becomes larger, the output would become more random and lose the ability to keep the input similarity even it comes from a normal data instance.

As illustrated in Fig 4.2 the classification network determines whether the input image and the reconstructed image belong to the same category. The classification network takes in a channel-wise concatenation of the input image x_0 and the reconstruction estimation $\hat{x}_0$. A CNN-based architecture is employed for this classification network. One challenge in this classification is that we only have access to normal

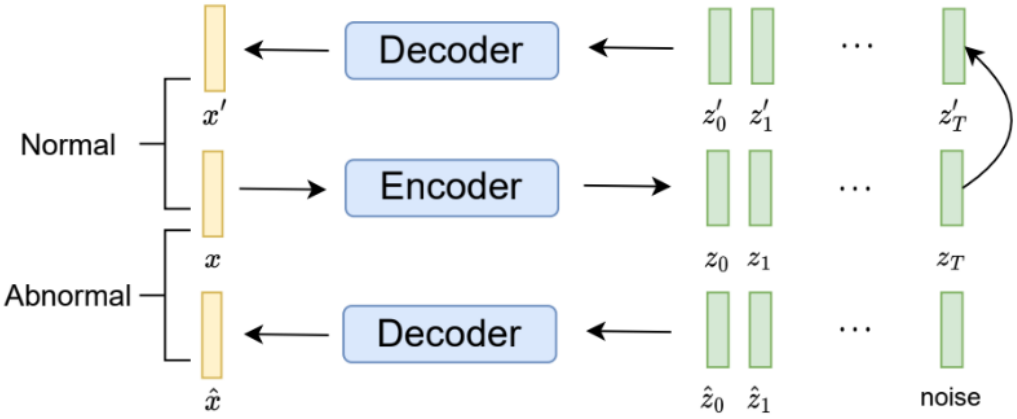


Figure 4.2: Illustration of the training process within the classification model. If the reversed input equals the forward diffusion process, the input image and reversed image are considered to belong to the same class. If the reversed input equals random noise, the input image and reversed image are considered to belong to different classes.

data instances, providing us with only positive labels.To obtain negative labels, for each training data input x_0 we reconstruct x'_0 by reversing the DDPM process from a random noise latent space.

4.1.3 Experiment

4.1.3.1 Datasets and metrics

This research primarily employs two datasets: MNIST [31] and CIFAR-10 [64]. MNIST is an extensive database of handwritten digits, ranging from 0 to 9, with images sized at 28×28 pixels. CIFAR-10 is a widely-recognized image classification dataset containing ten distinct object categories, each image being $32 \times 32 \times 3$ in size.

For anomaly detection studies associated with both datasets, the prevalent approach is the one-versus-rest scenario. In this, one object category is treated as normal data,

while the others are deemed anomalies. Notably, prior literature hasn't explored the MNIST dataset in a many-versus-one scenario. In this setting, models are trained on nine categories as normal data, with the remaining category considered anomalous. For the CIFAR-10 dataset, Semantic AD [65] has tackled the many-versus-one scenario using transfer learning. Meanwhile, UniAD [66] employed an embedding method in a many-versus-many context.

In our study, we explore the many-versus-one setting for the MNIST dataset and delve into the many-versus-many scenario for the CIFAR-10 dataset, employing a fundamentally distinct approach.

In this paper, all the experiments are using the Area Under the Receiver Operating Curve(AUROC) as the evaluation metric. AUROC scored is defined based on False Positive Rate(FPR) and True Positive Rate(TPR).

$$FPR = \frac{FP}{FP + TN} \tag{4.9}$$

$$TPR = \frac{TP}{TP + FN} \tag{4.10}$$

where FP represents false positive, TN represents true negative, TP represents true positive and FN represents false negative.

4.1.3.2 Reconstruction selection

Reconstruction-based anomaly detection algorithms are one of the most researched topics in anomaly detection. Numerous studies [2,39,67] have been developed in recent years. The primary assumption behind using a reconstruction model is that the reconstruction distribution should closely match the normal distribution. This assump-

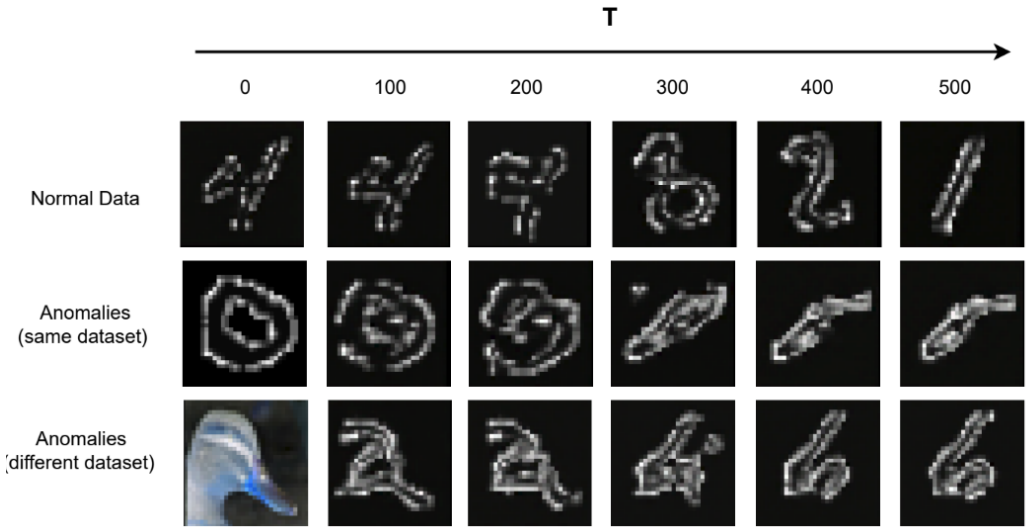


Figure 4.3: Reconstructions using our model trained on the MNIST dataset, excluding all instances of the digit '0'. The figure depicts reconstruction results for normal data, anomalies from the same dataset, and anomalies from a different dataset.

tion rarely fails under the one-versus-rest setting because learning the distribution of one category is typically straightforward. However, in a many-versus-one setting or many-versus-many setting, normal data includes different object categories, making the distribution challenging to describe. Often, the reconstruction-based model falls victim to the "identity shortcut" issue, where the output always attempts to replicate the input, regardless of the context.

Diffusion models show immense potential in image generation. Because the forward process of diffusion involves adding noise to the image, the reverse process becomes unstable. This instability can be beneficial, as it can prevent the model from taking the "identity shortcut" when evaluating an anomalous instance. However, it can also cause the reconstructed version of normal data to differ from the input. As seen in Fig.4.3 the reconstruction results change from timestamps 0 to 500. The stability of the reconstruction of normal data starts deteriorating after the diffusion timestamp

Table 4.1: AUROC score of anomaly detection on MNIST dataset

Anomaly digit	0	1	2	3	4	5	6	7	8	9
Autoencoder	53.1	60.2	62.2	57.8	55.2	56.9	56.3	50.3	63.1	51.2
AnnoDDPM	57.0	54.6	57.3	51.0	54.8	57.3	60.9	53.1	58.9	52.1
DDPM [70]	65.0	61.4	67.5	65.8	59.9	65.5	61.5	51.2	61.5	52.1
Our Method	64.9	73.2	72.6	69.7	69.7	68.7	68.0	72.6	71.5	56.3

200. Yet, the reconstruction results for anomalies begin to deviate from the input even before the diffusion timestamp 200. Therefore, we have chosen diffusion timestamp 200 in this study to effectively detect anomalous data.

4.1.3.3 Anomaly detection on MNIST

For our MNIST experiments, we adopted a many-versus-one setting. In each iteration, one digit was designated as anomalous data while our model was trained using images of the remaining nine digits. The architecture of the compression and diffusion models is grounded on the Latent Diffusion Model [68]. For the compression model, we employed a 3-layer autoencoder with channel sizes of [64, 128, 256]. This model compresses the image from a size of 32×32 down to a $8 \times 8 \times 3$ latent space, and it also incorporates a VQ-regularization [69] term. Subsequently, the diffusion training is facilitated by a 3-layer U-net model with channel sizes [224, 448, 672]. For classification, we deployed the ResNet-18 model. The input to this classifier is a concatenation of the original and the reconstructed image. As observed from Table 4.1 when our model is compared to three other reconstruction-based anomaly detection methodologies, our method consistently outperforms the others. Specifically, across all ten experiments, our model ranked as the most effective in nine out of the ten anomaly detection tests.

Table 4.2: AUROC score of anomaly detection on CIFAR-10 dataset

Anomaly classes	{01234}	{23456}	{45678}	{67890}
Autoencoder	50.4	51.4	60.8	51.2
AnnoDDPM	52.3	56.4	54.7	56.2
DDPM	57.6	51.8	54.6	53.3
Our Method	64.5	60.1	54.0	57.4

4.1.3.4 Anomaly detection on CIFAR-10

For the CIFAR-10 dataset, our experimental approach was grounded in the many-versus-many setting. In each iteration, we designated five distinct classes as the 'normal' dataset and the remaining five as 'anomalous' datasets. To clarify, in Table 4.2, the numerals 0 through 9 respectively symbolize the classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

The architectural foundation of our model for the CIFAR-10 dataset remains consistent with that employed for the MNIST dataset. However, our results on the CIFAR-10 were not as promising as those on the MNIST. Even though our model still surpassed other existing reconstruction-based algorithms, the performance decrement can primarily be attributed to the less stable reconstruction results on the CIFAR-10 dataset.

This instability might arise due to CIFAR-10 images being more complex and diverse in content than MNIST's handwritten digits. Thus, while our model demonstrates superiority over other reconstruction-based approaches, there remains a potential for refining and optimizing it further, especially when tackling complex datasets like CIFAR-10.

4.1.4 Summary

Tackling multi-class anomaly detection is a formidable challenge, given the intricate distribution characterizing normal data. Our approach, anchored in the latent diffusion model, underscores the promise and efficacy of this method for addressing such anomaly detection challenges. Notably, our model presents a remedy to the identity-shortcut predicament that frequently plagues conventional reconstruction-based anomaly detection mechanisms. A promising frontier for ensuing research in this domain is delving deeper into methodologies that can further stabilize the reverse process in diffusion during anomaly detection tasks.

4.2 3D Unclonable Optical Identity

4.2.1 Introduction

Table 4.3: Ideal secure ID requirements and the features of popular ID techniques in use today.

	ID cost	Reproduction Cost	Verification Cost	Application Scenarios
Ideal Secure ID	Low	High	Low	Wide
Traditional Optical ID	Low	Low	Low	Wide
Hologram	Low	Medium	High	Wide
Nanostructure, DNA	Low	High	High	Wide
PUF	Medium	High	Low	Limited to electronic device

Reliable identity (ID) is the cornerstone for the improvement of global supply chain, as it helps owners and participants in the supply chain to detect the IP theft, counterfeiting, mishandling and other potential risks. For the convenience of daily commercial activities, product ID need to meet several conditions. For example, the ID needs to be safely attacked to the product or on the original package. The manufacturing cost of the ID should be low enough to be used for less expensive goods. The verification process should be convenient to the users. People have developed

numerous techniques to ensure and protect the production IDs, however techniques meets all these requirements are still in a lack. For example, bar-code [71, 72], quick-response (QR) code [73] and radio-frequency ID (RFID) [74] can easily be copied and used on counterfeited goods. Holograms, DNA marks [75] and nanomaterials based IDs [76], which are based on unique and random micro/nano-structures, typically require equipment in labs for verification. The silicon based physical unclonable function (PUF) [77] is only realized in integrated circuit chips and thus not suitable for non-electronic productions. The features of these popular commercial ID techniques are listed in Table 4.3. The requirements on ideal secure ID techniques are also listed for comparison.

In this project, we propose a novel type of IDs that is irreproducible, reliable, and applicable on most productions, including but not limited to electronics and high value goods. The ID is in sheet form, including randomly distributed micro-bubbles. The unique feature of each ID is based on 3D-spatial locations of the particles. The irreproducible features are introduced during the fabrication in an unintentional way, which helps keep the fabrication cost low. While to reproduce a specific ID, the characteristic 3D feature reproducing rather than secret features, the product database for verification does not have to be kept secret. The novelties of our technique include:

- 3D structure information of each ID is exploited, which makes the duplication unlikely even with the state-of-art fabrication techniques. The ID security relies on the duplication difficulties instead of any secrets.
- The IDs fabrication is adaptable to 3D printing, which makes the technique applicable on most physical items with firm or flexible solid surfaces, such as electronics, clothes, food packages, and pharmaceuticals. The IDs can also be applied on as-fabricated items.

- The ID features are recognized by machine-learning enhanced affordable optical systems, which makes the verification user-friendly and feasible in daily commercial activities.

- Environmentally sensitive materials can be used, which allows the IDs to irreversibly record the environmental experience such as an exposure to radiation or high temperature.

4.2.2 Attack and Defense Model

Along the life cycle of a physical product item, it may go through different stages, such as design, manufacturing, distribution, and being deployed. Here we define the attack and defense model through several assumptions. *Assumption 1: ID register is trusted. The original item owners are responsible for the item registration.* They can only register their items with their own names/brands. *Assumption 2: product database provides write access to authorized registers and read access to the public.* Original owners can add the item information and identity into a product database, which provides public read access while the write access is only open to the original owners. *Assumption 3: no secrets. For the item that is available on market, we also assume the adversaries have complete knowledge about the items they try to compromise, and the adversaries have accesses to all known technology.* It is worth noting that this is an assumption of the most powerful attacking, indicating that there is no secret (e.g., private key) for the adversaries. *Assumption 4: a verifier representing the receiver is required every time when the item possession is transferred, and the verifier is trusted.* When the item is transferred to a new party or enrolled into a new system, honest verification should be performed. Since the verifier is responsible for the verification results, we assume the verifier is trusted. The verifier shall have

read access to the product database. Although the Assumption 4 typically requires human involving, it is NOT an extra demand as in most scenarios people present on item receiving. The important thing is to make the verification user-friendly without requiring professional training. The target of the defense strategy is to make the ID reproducing and reuse difficult enough. Thus, we can ensure that with reasonable cost, the adversaries can ONLY make an insignificant number of compromised items to pass the verification successfully. The definition of an insignificant number can be 0 or a small percentage, depending on the specific application scenarios.

4.2.3 Experimental

4.2.3.1 Bubble Tag Preparation

In this work, epoxy resin (EcoPoxy, Morris, MB) was selected to create bubble tags, which was composed of two parts: liquid resin (Part A) and liquid hardener (Part B) [78]. Bubble tags were prepared by mixing Part A with Part B at a volume ratio of 1:1 in a plastic cup and then manually stirring with a glass rod for 5 minutes. Thus, numerous air bubbles were generated in the mixture. After that, the mixture was poured into a plastic petri dish as a sample with trapped bubbles. After 48 hours, the sample was completely cured at room temperature and was able to be used as a bubble tag. Figure 4.5 shows the size of our prototype tag compared with an American quarter coin. Each intersection point can be seen as the center of one tag, the size of each tag is about 2 cm$\times$ 2 cm $\times$ 0.5 cm and it is much larger than what we need.

It is noted that torching was a promising method to control the bubble size and concentration. Herein, a butane torch (Corkas) was utilized at 30 cm away from

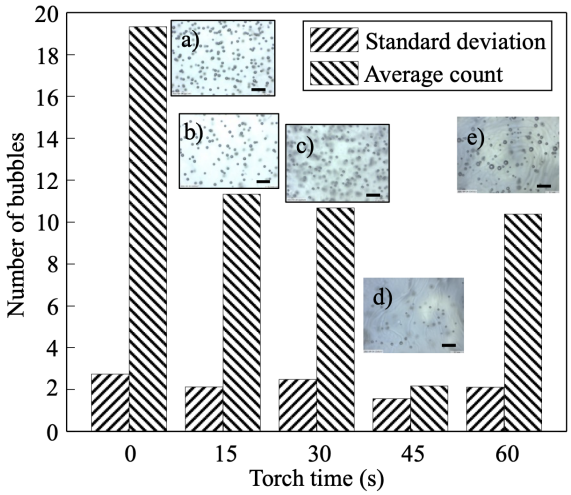


Figure 4.4: Standard deviation of the number of bubbles and average bubble count for the samples at different torch times. Insets: sample torched for a) 0, b) 15, c) 30, d) 45, and e) 60 seconds. (Scale bars: 0.5 mm.)

the surface of the sample to remove unnecessary bubbles from epoxy resin. A series of experiments were designed to investigate the effects of torch time on the deviation/average count, in which different torch times (0, 15, 30, 45, and 60 seconds) were applied to the samples. Then, a high precision measurement system (Micro-Vu Vertex System, Windsor, CA) was used to image and measure the bubble size and number in the samples with different torch times as shown in Figure 4.4. It is found that the increase of torch time results in the decrease of bubble quantity and generation of rippled surfaces (as shown in Figure 4.4d) and e)), which may bring background noise in the following bubble tag detection. Only when the torch time was 15 seconds, the samples had the smooth surface, lower standard deviation in the number of bubbles, suitable bubble density (10-12 bubbles/1.5 mm2), and average bubble size of 14 µm. As a result, the torch time was determined as 15 seconds.

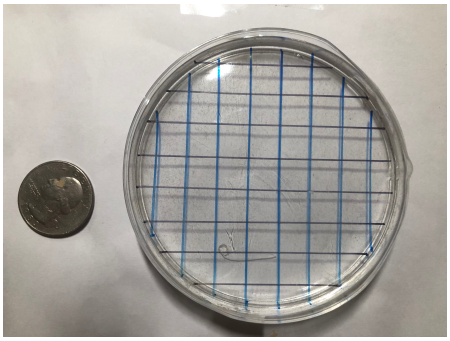
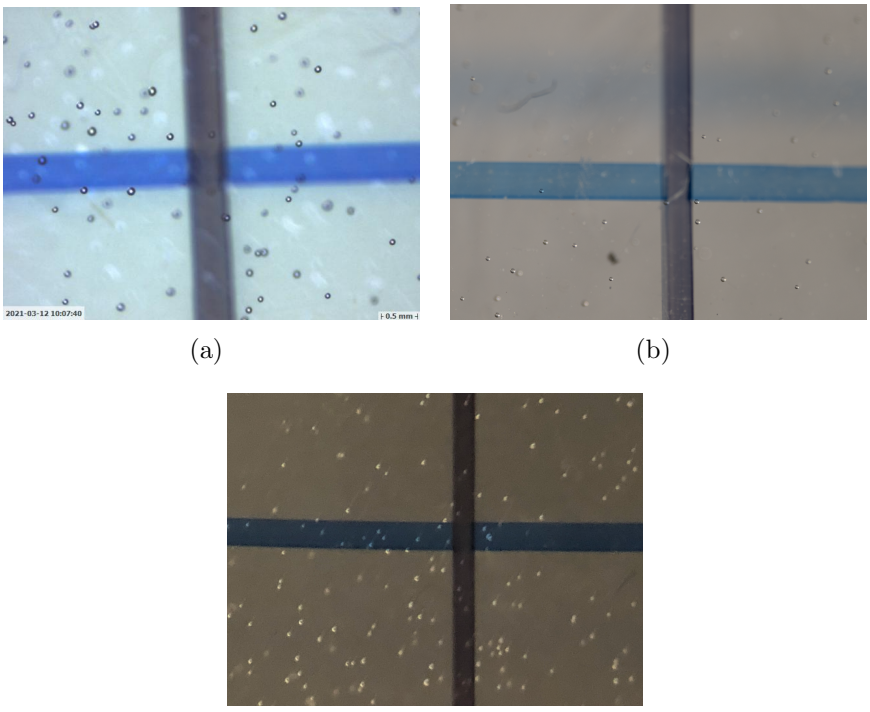


Figure 4.5: Size of our tag compared with a quarter coin (about 40 tags in the disk).



(a) (b)

(c)

Figure 4.6: Tag imaging sample from different imaging devices. (a)microscope (b)digital camera (c)cellphone camera with macro lens

4.2.3.2 Bubble Tag Imaging

To easily locate the bubbles in different depths inside a sample, the sample was made via the aforementioned protocol and after curing, horizontal and vertical lines were drawn on the surface of this layer with a permanent marker. The distance between the adjacent horizontal/vertical lines was 1 cm. The high precision measurement system (Micro-Vu Vertex System, Windsor, CA) was used to image the bubbles at each crosshair at 4 different depths, which has the imaging resolution of 0.1 µm along X, Y, and Z-directions. Images can be taken in color or monochrome using a highly sensitive camera that has a 36:1 zoom range [79]. The light-emitting diode (LED) lights have a lighting angle range from 25 degrees to 90 degrees and have about a 10,000-hour lifespan [79]. The brightness and pattern of the LED lights can also be customized to take images.

In the verification process, we also tried different look out devices. Figure 4.6 shows imaging result from microscope, digital camera and cell phones. Cell phone camera may not good enough to catch the bubbles in our tag, however a $10 worth macro lens as a cellphone accessory may clearly get the results.

Table 4.4: Database Structure

number	ID	center position	layer depth	identifier	Product Information
i	p_i	$(x_c, y_c)_i$	0	$(x_0, y_0)_{i,0}, (x_1, y_1)_{i,0}, ..., (x_{m_0}, y_{m_0})_{i,0}$	1. Manufacturer 2. Ingredients 3. Product description 4. Expiration date
			z	$(x_0, y_0)_{i,z}, (x_1, y_1)_{i,z}, ..., (x_{m_1}, y_{m_1})_{i,z}$	
			...	...	
			nz	$(x_0, y_0)_{i,nz}, (x_1, y_1)_{i,nz}, ..., (x_{m_2}, y_{m_2})_{i,nz}$	

4.2.4 System

4.2.4.1 Initial Strategy

From the figure of our 3D tag, we can notice that the location of particles can be seen as the identifier of our tag, thus we need to build up a 3D coordinate system and save the (x, y, z) information for each particle in our sample. In order to collect the depth information z, we took images on different focal planes for each sample. By adjusting the distance between each focal plane, we can make sure each particle only appears in one or two images. The depth of the focal plane can be used to indicate the depth information of the particle. For the horizontal information x, y we can get it from the images we take. Also, we define the cross point on the top level as the origin of our coordinate system.

The position of the cross point in sample images can be easily found by using the Sobel Operator. Sobel Operator can detect the edge of cross lines, thus we can get the cross point by calculating the intersection points of four edges. The Sobel Operator can be shown as follows:

$$L_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} L$$

$$L_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} L$$

As we can see from the sample figure, the shape of our particles is a circle. We

have two ways to find out the position of all circle patterns in the sample image. The naive way is using the Hough Transform method [80]. This method can find out all the circle shape patterns in the images. However, under real-world circumstances the sample image may be noisy, using Hough Transform we will get a large number of noisy data. Furthermore, we may change the materials of particles in the real-world application. When the shape of particles changed, Hough Transform will not correctly detect the particles. Another method we could use to detect particles is machine learning based object detect algorithms [81]. The first advantage of machine learning based algorithms is that they can easily adapt to the change of particle shapes. Furthermore, it can ignore the noisy information if the model can be well trained. However, training a generalized machine learning model requires a large number of data. The creation of training data will cost months in our laboratory.

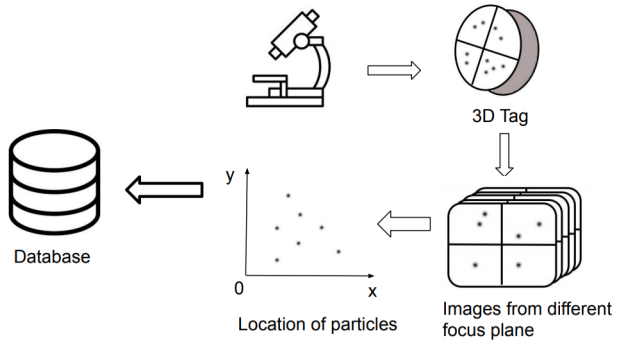


Figure 4.7: System working principle

Our strategy of getting enough data is creating simulated data. We created thousands of simulated images with a clean background and round-shape particles. Then we used the Hough Transform method to detect the position of particles and feed the result into the machine learning model. Then we can use a small number of real data to make the model detect the real particles.

4.2.4.2 Architecture and Pipeline

Our 3D tag consists of two parts: (i) a certain number of small particles(e.g. bubbles) and (ii) cross lines on the surface of our tag. The position of particles is in charge of generating a unique, unclonable identifier for tracking and tracing commodities in the supply chain. The cross line on the surface is in charge of finding the initial focal plane and locate the center of the tag.

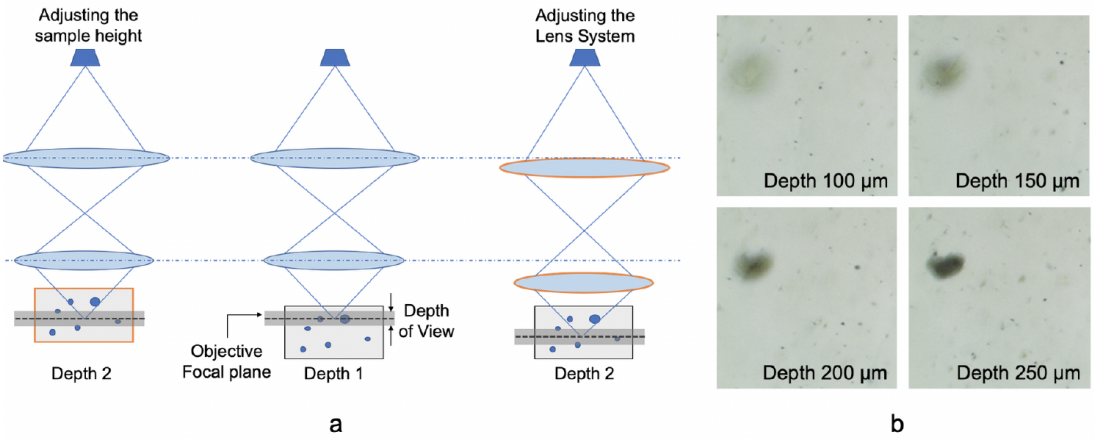


Figure 4.8: Depth information from imaging: (a) schematics showing depth information extraction by adjusting the sample height or adjusting the lenses position and (b) images obtained from one sample by adjusting the sample height

We obtained the depth information by controlling the objective focal plane. For microscope system, the depth of field (DOF) is typically very shallow. As shown in the fig4.8, only particles near the objective focal plane (i.e. within the DoF) can be clearly recorded on the image. The depth information accuracy is decided by DoF and calculated by

$$DoF = \frac{2NCD^2f^2}{f^4 - N^2C^2D^2} \tag{4.11}$$

Where N is aperture, C is circle of confusion, D is focusing distance, and f is focal length. For the microscope system to be used in the proposed project, the DoF is from 5 to 30 μm. Given the system information, the depth corresponding to particular

particles can be extracted. In order to record the particles at different levels, we first set the focus plane to the top of the tags where we can clearly see the cross lines. Then we move the position of the lenses and take an image for each millimeter change. Figure 4.8 shows the images taken from our samples by microscope.

As show in Figure 4.7 the procedure of implementing a machine learning based tag enrollment system for each sample is as follows:

1. Use a series number p to denote the smaple
2. Adjust the location of the lens in order to clearly see the cross markers on the top of the sample, also put the intersection point at the center of the lens.
3. Take one image at the top layer then move the lens vertically down, and take one image for each z millimeter.
4. Use yolo algorithm on each image to find out the center of each bubble, we use (x^m, y^m, nz) to denote m^{th} bubble in layer n and save all the data into our database
5. Use edge detection algorithm find out the center point (x_p, y_p), save it into our database

The procedure of authentication process for each sample is as follows:

1. Adjust the location of the lens in order to clearly see the cross markers on the top of the sample, also put the intersection point at the center of the lens.
2. Move the lens vertically down s millimeter and take one image
3. Use yolo algorithm find location of bubble location in this image

Algorithm 1 3D tag authentication

Input:

Product series number, test identifier, center position, distance to top $\leftarrow$ p_i, $\{(x,y)_{p_i,test}\}$, $(\hat{x},\hat{y})$, s;

The database information test identifier in layer k, center position $\leftarrow \{(x,y)_{p_i,kz}\}$, $(\hat{x}',\hat{y}')$;

Output:

Initial parameters $m, n = 0$
Calculate center shift distance $(\delta x, \delta y) = (\hat{x}',\hat{y}') - (\hat{x},\hat{y})$
Find out k such that $kz \leq s \leq (k+1)z$
for all (x,y) such that $(x,y) \in \{(x,y)_{p_i,test}\}$ **do**
    n = n+1
    for all (x',y') such that $(x',y') \in \{(x,y)_{p_i,kz}\}$ **do**
        if D$((x+\delta x, y+\delta y),(x',y')) < d_{threshold}$ **then**
            m = m+1
            Break
        end if
    end for
end for
if $\frac{m}{n} > s_{threshold}$ **then**
    printf("test sample matches tag p_i")
else
    printf("test sample does not belong to the database")
end if

4. Use edge detection algorithm find out the center point of this image

5. Compare the image with its neighbour images enrolled in the database. If the similar rate is larger than a threshold r_{th}, then we determine that the test sample matches with this tag entry; otherwise, we determine that this sample does not belong to the database.

However, because we couldn't save all the 3D information into several 2D images, the test image may slightly different from the data we saved in the database. For example, we assume the depth of the test image is between layer k and layer $k+1$ in the database, which means $kz < s_{test} < (k+1)z$ With the movement of the focus plane, some bubbles show up in layer k or layer k+1 will disappear in the test image. We assume in the registration process, we record all the bubbles in this 3D model, which means each bubble at least appears in one layer we saved in the database. Thus if the test sample and the target sample are the same one, we know that

$$\{(x,y)_{i,kz}\} \cup \{(x,y)_{i,(k+1)z}\} \subseteq \{(x,y)_{i,s_{test}}\} \tag{4.12}$$

We can define the distance between test image and the sample saved in our database based on the percentage of bubble in test image which can also be found in layer k or layer k+1. The equation can be wrote as follows:

$$f(p_i, p_{test}) = \frac{\#\text{bubble in test sample and sample } i}{\#\text{bubble in test sample}} \tag{4.13}$$

Because of noise interference and small error in detection algorithm, the position detected for the same bubble may be slightly different in different detection. Thus we use euclidean distance between two points to determine whether these two points

are in the same location.

$$D((x,y),(x',y')) = \sqrt{(x-x')^2 + (y-y')^2} \tag{4.14}$$

Two points are determined to belong to the same location if their distance is not larger than a threshold, we will discuss the infection of the threshold in the evaluation subsection.

4.2.4.3 Application Scenario

Figure 4.9 illustrates the communication flow in the real application scenario. Our 3D tag is really small, it can be placed anywhere inside the package. Users can use a smart phone with our 3D tag reader software to take an image of the tag and download the tag-related information from the centralized database. The communication flow of our 3D tag system is as follows:

Step 1: 3D tag reader take an image 3D tag

Step 2: Reader find out the center position and identifier information of the tag.

Step 3: Reader send the information to the centralized database for authentication. The authentication process is as algorithm 1 shows.

Step 4: The centralized database sends the authentication result and corresponding product information to the reader.

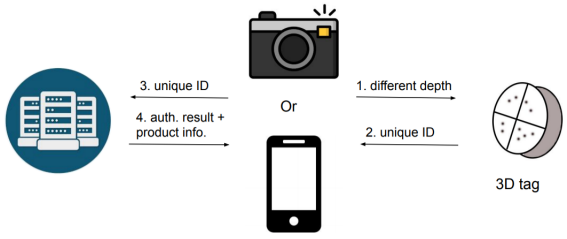


Figure 4.9: Communication flow in real application scenario.

4.2.5 Evaluation

4.2.5.1 Object detection

In the process of training our object detection model, we first manually labeled 50 sample images. For the reason that we only want the algorithm find out the bubbles in the focus plane rather than the blurry bubble out of focus. We can see the difference in Figure 4.10 We manually labeled 50 samples and train the YOLO model based on these 50 samples.Our image size for training is 1024×1024, our batch size is 5. The result of our training result is as Table 4.5 shows.

Most of undetected bubble are under motion blur situation where it looks similar with the out of focus object. Thus, if too much motion blur exists on the object, the algorithm will consider object is out of focus and will not correctly detect them. Another detection error is that some out of bubbles are incorrectly detected. The reason for this situation is because of some incorrect labeling in the manual labeling process.

Figure 4.10: Bubble in focus plane versus bubble out of focus plane

Table 4.5: Training YOLOv5 NEURAL NETWORK RESULTS

Epoch	Precision	Recall	mAP value
200	0.11	0.23	0.08
300	0.12	0.25	0.10
400	0.13	0.51	0.12
500	0.62	0.39	0.41
600	0.65	0.80	0.66
700	0.73	0.83	0.75

Table 4.6: Design Parameters for 3D Tag

Variable	Parameter	Value
z	distance between different layers	1mm
d_{th}	distance threshold between same point	1mm
s_{th}	similarity threshold in identification	0.4
n	number of layers we saved	4

4.2.5.2 Performance evaluation of 3D pipeline

In this subsubsection, we analyze the effectiveness of pipeline efficiency in the real world application. We built 100 samples, each sample included 4 layer images, which we saved into the database, and one test image which we used to evaluate our pipeline. The settings of our experiment is listed in Table 4.6 and the verification result can be seen as the Figure 4.11 shows.

From the experiment result we can see that when we verify the tag with it's own image, the similarity of all 100 test images are all above 0.3. Actually the similarity of 97 test samples are larger than 0.5. However, when we verify the tag with a random image from our dataset, all test images have similarity smaller than 0.25, 98 of them are smaller than 0.2. There is a huge gap between 0.2 and 0.5, we can set a threshold between 0.2 and 0.5, if the similarity is larger than the threshold then the test image belongs to the tag. If the similarity is smaller than the threshold, the test image does not belong to the tag. If we set the threshold equal to 0.4 in our experiment, than the accuracy of our verification process is equal to 0.995.

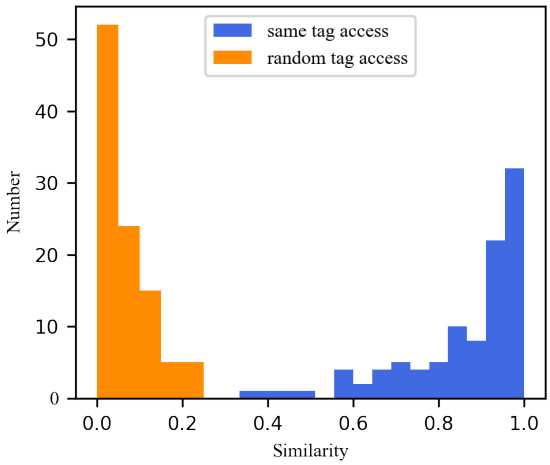


Figure 4.11: Tag verification results

To further evaluate the performance of verification process, we used Unreal Engine 4(UE4) to simulate the whole process. Unreal engine is a game engine which can provide us different light condition and camera settings. The simulation is very fast after we built the model, it takes about two minutes to build one sample and takes all the images we need. Further more, we test our samples within different light environment and camera settings. Figure 4.18 shows the difference between simulation tag samples and real tag samples. In this experiments we simulate 400 samples and trained a new YOLO model to detect the black points on the simulation images. Then we exactly follow the whole process in the real tag verification part. Figure 4.13 shows the result of the verification process of our simulated samples.

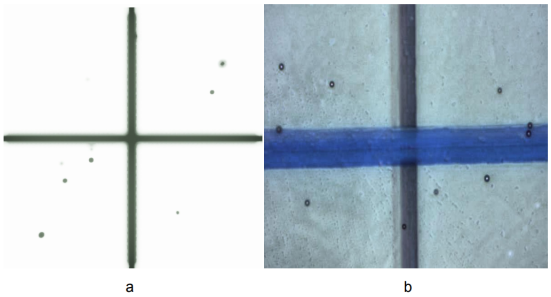


Figure 4.12: Comparison between simulation tag and real tag: (a)simulation tag built based on UE4 (b) real tag built based on resin

This experiment result is very similar with the previous experiment result. Among 400 test images from the correct tag, about 93% of them have similarity larger than

0.5. While with the 400 test images from the incorrect tag, about 93% of them have similarity smaller than 0.3. Also, if we choose 0.4 as the threshold to identify whether the test images belongs to the tag or not, the accuracy is about 96.9%.

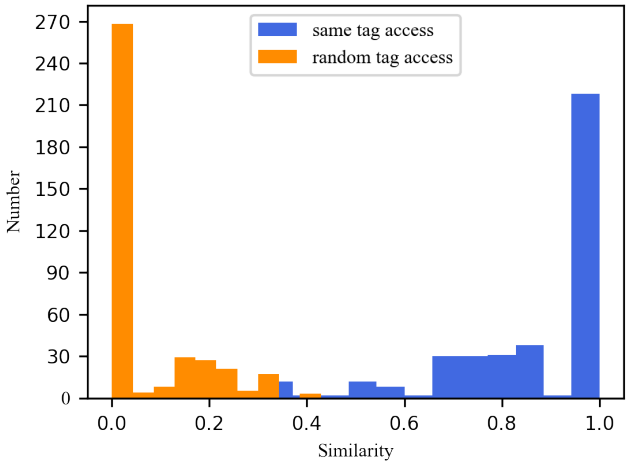


Figure 4.13: Simulated tag verification results

4.2.6 Reformulation with Anomaly Detection

The result of using YOLO could highly depend on the training data and verification data. The user would get their verification data under different circumstances, for example different verification devices, or different light conditions. Thus our multi-domain anomaly detection system will help under this situation. As we can see from Fig.4.14

The input of compression model is the background image x, the compression procedure can be expressed as $z = E(x)$, and the decode procedure can be denoted as $x' = D(z)$.

The input of the diffusion model is the latent space vector z from compression model. Then we follow the forward process of the DDPM from equation 4.2. Given

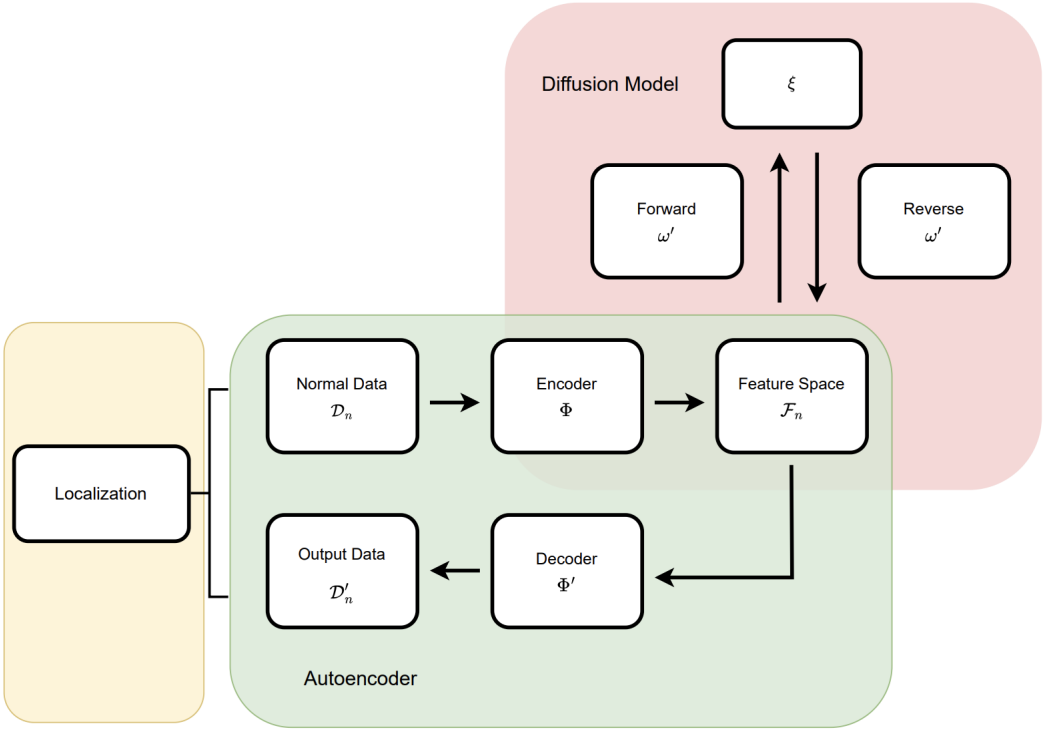


Figure 4.14: Anomaly Detection in 3D tag

any time $t \in [0, T]$, the latent space z_t can be calculate by:

$$z_t = z_0\sqrt{\bar{\alpha}_t} + \epsilon_t\sqrt{1 - \bar{\alpha}_t} \tag{4.15}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$.

As t becomes larger and larger, more and more Gaussian noise is added into the image and the latent vector z_t looses its original spatial structure and looks near the Gaussian noise. In the reverse process, we can follow the equation 4.7.

In our localization task, the result of DDPM reconstruction would then be the actual point localization based on the background subtraction.

4.2.7 Security Analysis

The ID-based verification relies on several essential factors, including 1) registered ID, 2) database storing the ID information, 3) proper ID reading, and 4) securing communication between database and ID reader. In this scenario, the compromising of database (on cloud), ID reading (by cellphone for example), and the communication are related to more general cyber-attacks and thus are not discussed in this paper. The exclusive counterfeiting attack that is specifically targeting our technique is the ID reproducing. As discussed in Section 4.2.2, the ID information is public, which means the adversary can get the 3D structure information. To reproduce the structure, popular manufacturing techniques, such as casting [82], printing [83], and lithography [84, 85], can be used.

Different techniques have different advantages and limitations for the ID tag reproducing. For example, the casting is suitable for 3D structure, the cost is economical, but the precision is only around 100 µm. Therefore it is impractical to present structures with feature size below 100 µm. In addition, it is also difficult to fabricate some complex patterns like the separate particles in our ID tags. Different from casting, modern printing techniques provide more precise and controllable ways for the manufacturing. However, the printing performance in several aspects like precision, speed, and cost, are mutually restrictive. Although the precision of 3D printing can be up to 10 µm (i.e. several thousands of dpi) with uniform materials, it is challenging to handle different ones.

Lithography is another technique that has been widely used for micro-structure manufacturing, from cost-efficient printed circuit board to expensive advanced CMOS processing. Considering from the perspective of economic rewards, the counterfeiting may only use mask-based contact lithography [84] or maskless lithography [85]

(e.g. digital light processing, DLP), both providing precision up to 10 µm. Since the lithography is layer-based operation, multiple rounds are necessary to stack several layers for the 3d structure manufacturing. The number of layers depends on the verification protocol. For example, if the verification requires images from five different depths, then the adversary has to fabricate at least 5-layer counterfeited ID tags with corresponding thickness.

It is worth noting that lithography only generate binary patterns with sharp edges (i.e., black-while, no gradient gray). Considering the particles that are out of focus but still shadowed in the depth-related image (gradient gray) are also used for verification, a counterfeited ID will need more layers (>5 in this example) to clone the shadows as well. Using the Figure 4.10 as an instance. Although only one image is used for verification, the adversary needs to generate two layers at different depth, such that first counterfeited layer can be in focus to show clone the in focus bubble while the second counterfeited layer is "properly" out of focus to clone the shadow of the bubble that is out of focus. According to the z-direction distribution, the thickness of each counterfeited layer needs to be adjusted as well. Thus, the manufacturing cost will increase rapidly.

Overall, casting and 3D printing are technically difficulty to manufacture patterned 3D structure with feature size at or blow 10 µm. Lithography, on the other side, can provide higher precision but the layer-based operation mode makes the 3D structure reproducing expensive. Potential manufacturing issues, such as the layer curing and the interface between layers, may result in further challenges to the adversary during the imaging and verification.

4.2.8 Summary

In this article, we presented an innovative unclonable 3D tag that carries a unique identifier that can be used on all kinds of materials. The effectiveness of tag recognition has been verified via experiments based on our 3D tag prototype. Our 3D unclonable can be printed directly on products or their packages, integrated onto PCBs of electronic products. Compared with existing approaches, our 3D unclonable tag has the following advantages: (1) The random 3D structure and uncontrollable process during tag fabrication make the tag hard to clone. (2) The fabrication process can be done by a 3D printer which makes our tag can be applied on all physical surfaces. (3) Our proposed tag look-up system has a user-friendly verification process. (4) Environmentally sensitive materials can be used in our tag, which makes it possible for the IDs to irreversibly record the experience such as exposure to radiation or high temperature.

Our future work mainly focuses on two directions. Finding the suitable environmentally sensitive material inside our sample is our first direction. These kinds of materials can track the environmental change of commodities in the supply chain. The other direction is improving the object detection model to verify test images taken from different tools(e.g. cell phone, camera, and microscope).

4.3 Gould Syndrome Detection

4.3.1 Introduction

Gould Syndrome is a rare, genetic, multi-system disorder. Gould Syndrome is often characterized by abnormal blood vessels in the brain, eye development de-

fects, muscle disease, and kidney abnormalities. However, many other aspects of the syndrome including abnormalities affecting the structure of the brain and lung abnormalities continue to emerge and the full spectrum is still uncharacteristic.

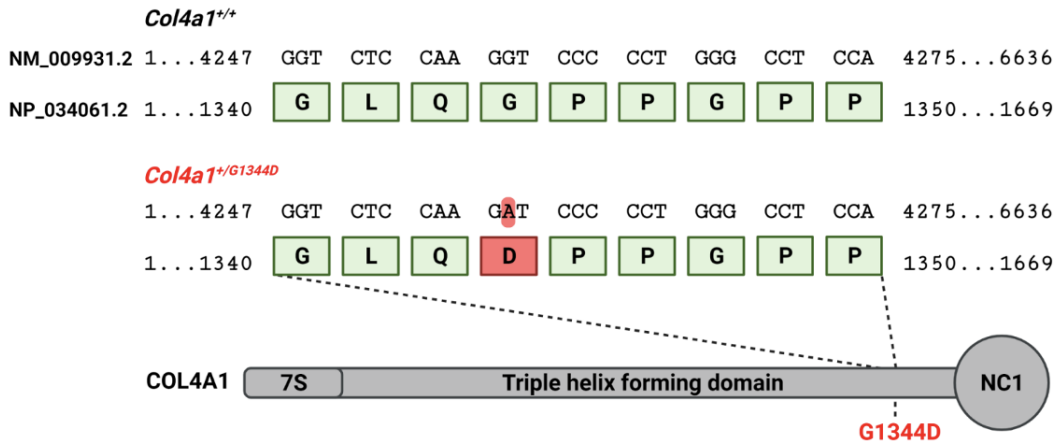


Figure 4.15: The mutation of COL4A1 gene

Recent study shows that Gould Syndrome is caused by mutations in components of type IV collagen. Type IV collagen is the main collagen component of the basement membrane which which is important for various physiological and pathological functions. Collagen IV composed by heterotrimers of COL4A1, COL4A1, COL4A2. In this research we are focusing on COL4A1 gene as the COL4A1 mutations are heterozygous dominant of mutations in type IV collagen. The example of COL4A1 mutations can be seen as Fig4.15 shows. In our experiment, we mutate COL4A1 gene in mice and exhibit age-dependent anterior segment ocular dysgenesis and intracerebral hemorrhages, it is shown in Fig4.16.

The next step is to figure the affection of mutated COL4A1 gene in vascular smooth muscle cells(SMC) phenotype. From Fig4.17 we can see different SMC phenotype between mice with normal COL4A1 gene and mice with mutated COL4A1 gene. We can see that most part of smooth muscle cells are similar, the texture structure of the cells looks like parallel lines. However, smooth muscle cells in mice with

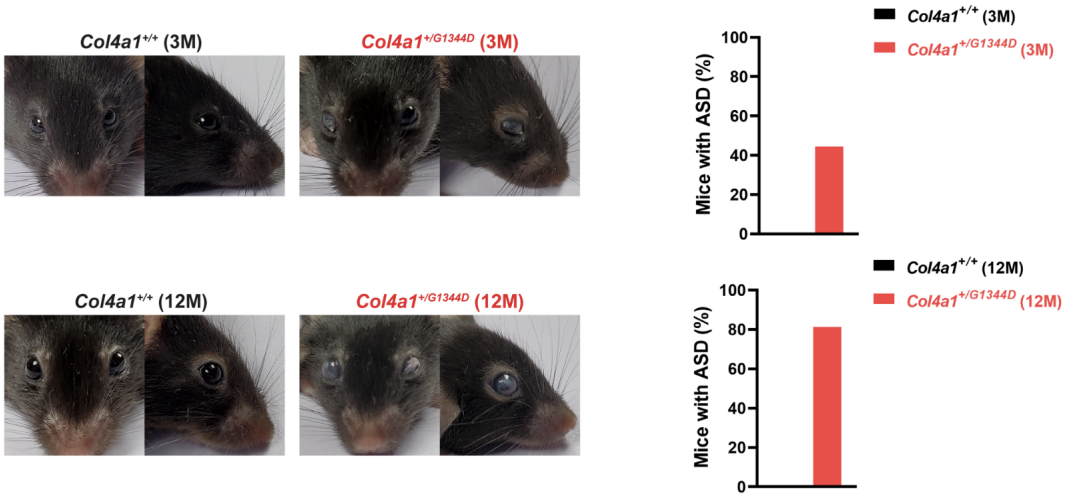


Figure 4.16: The comparison of mice with normal COL4A1 gene and mutated COL4A1 gene. The left top figure shows mice in 3 month and the left bottom figure shows mice in 12 month. The right figures indicate the percentage of mice with anterior segment ocular dysgenesis.

mutated COL4A1 gene forms irregular cell texture in some particular places.

The critical task is to find out the percentage of anomalous smooth muscle cells with irregular cell texture. In order to complete this task We used image segmentation techniques to find out the normal smooth muscle cells and anomalous smooth muscle cells. Image segmentation aims to give a label to each pixel in the image, in this task we created three labels, background, normal cell and abnormal cell. Once we get the segmentation result, we can calculate the percentage of anomalous cells among all cells. The difference between our task and traditional image segmentation job is that traditional image segmentation aims to segment based on different objects but in our task, we aims to segment based on image textures.

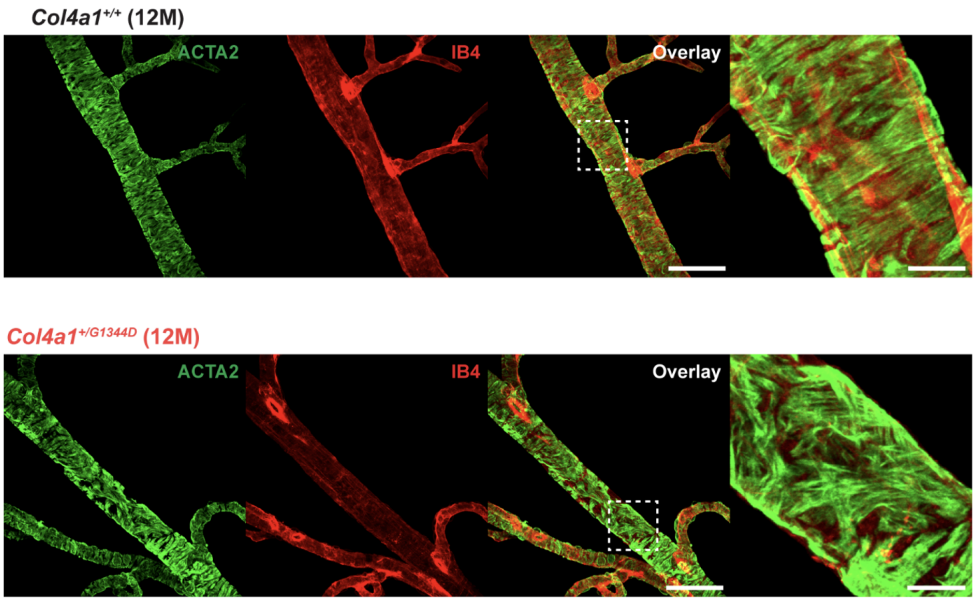


Figure 4.17: Mice exhibit changes in vascular SMC phenotype

4.3.2 Challenges

In this task, we are facing many challenges similar with typical anomalous detection. Some main challenges are list below:

- **Blury boundary** The boundary between normal and anomalous cells are not clearly defined. Because all the data are labeled manually, some texture pattern between line texture and messy texture are hard to labelled. Actually, in some figures, the first manual label result and the second manual label result are different.

- **Limited Labeled Data** With the usage of deep neural network to finish the segmentation task, it requires a large amount of training data. However, in our experiment, The number of mice we could use are very limited. In addition to that we need to manually annotated all the labels which is impossible to get a large amount of data.

- **Class Imbalance.** Anomalous cell type are hard to find. Most of smooth muscle cells are structured with normal texture. Messy texture only show up in small area in the body of gene mutated mice.

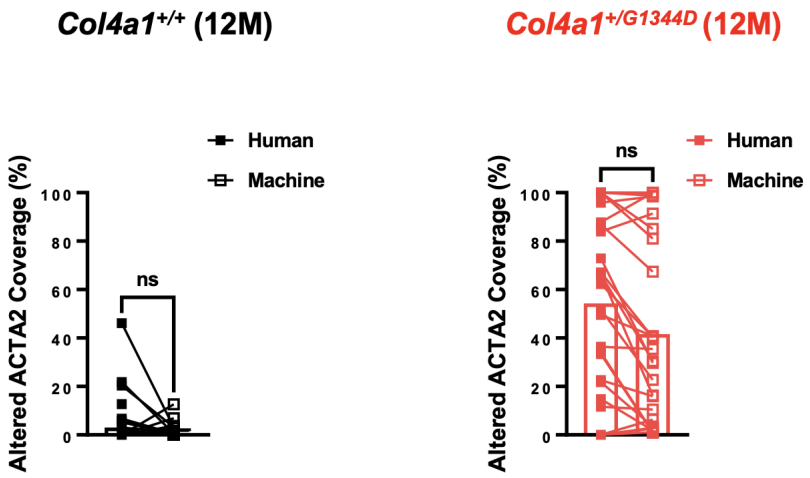


Figure 4.18: Unet labeling result vs Human labeling result

4.3.3 Initial Method

We tried to use existing algorithm to solve this problem. From the beginning, as limited annotated data is available, we used data argumentation to increase the size of the training data. We used a set of affine transformation like flip, rotate and mirror. Then we tried to use UNet model to finish the image segmentation task. We compared the UNet segmentation result between manual labeled result and our automated process, we can see the result from Figure4.18.

Manually labelled result and segmentation result from UNet shares the same trend. Notice that there were no statistically significant difference between our method and human. However, when looking into the details in Fig.4.19 we can find that UNet

couldn't catch some of the anomalous texture area. More importantly, UNet segmentation could cut one cell into different part. It is impossible half of the cell is normal and another half is anomalous.

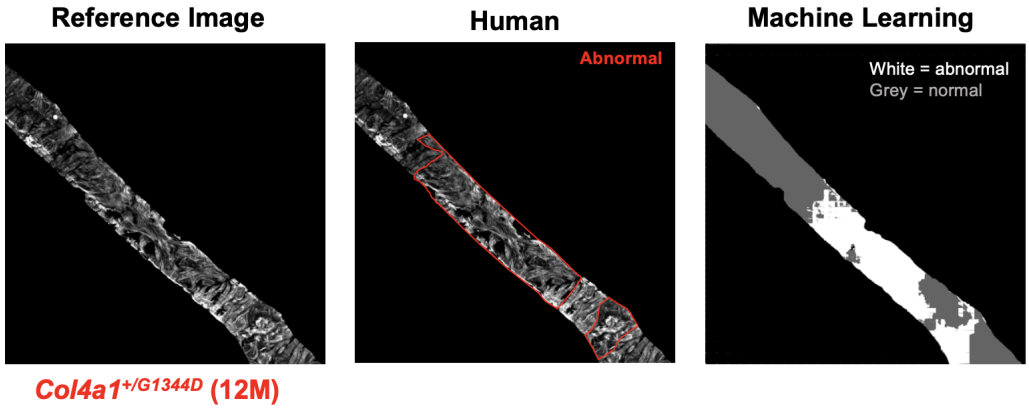


Figure 4.19: Segmentation result between human vs Unet

I have two future plans for this project. The first one is to build up a unsupervised model to let the model detect anomalous cells. The second one is build up a texture based segmentation model which directly fit into the smooth muscle cells segmentation.

4.3.4 Reformulation with Anomaly Detection

The main challenge in Gould Syndrome detection is due to the label quality and quantity. Anomaly cases are very rare, and the boundary between normal and abnormal areas is ambiguous. Thus, we can use our multi-domain anomaly detection pipeline to address these challenges with only normal data needed in our training process. The pipeline for using anomaly detection to solve this problem can be seen in Fig.4.20.

The normal image x serves as the input for the compression model. The compression step is represented as $z = E(x)$, while the decompression step is denoted as

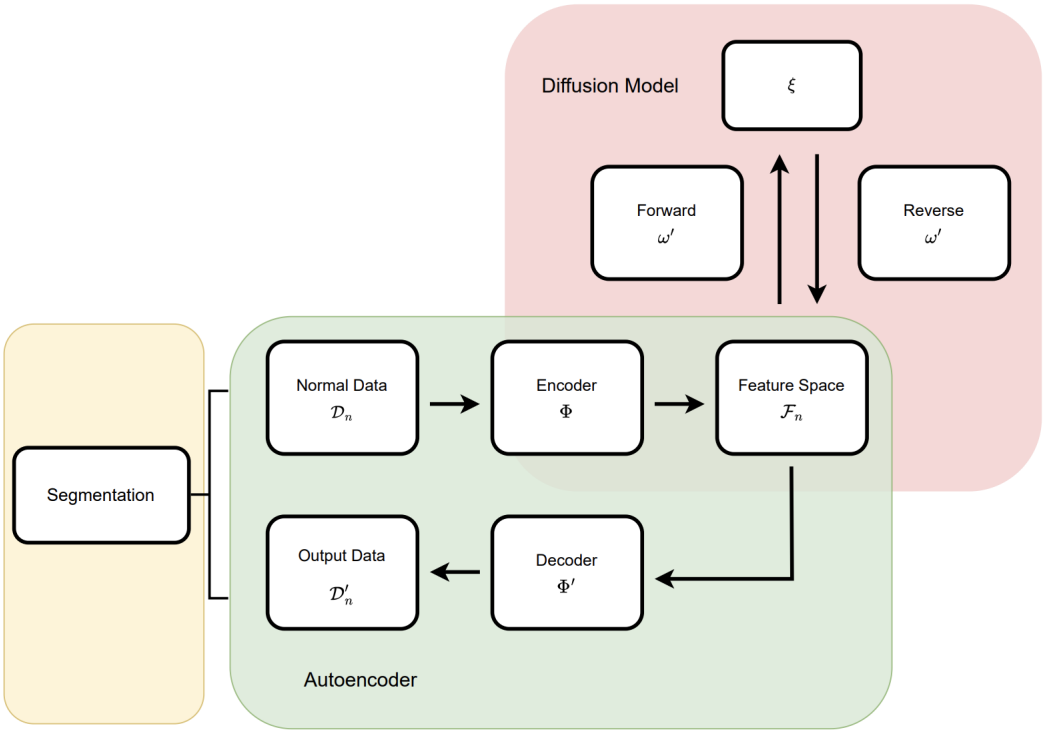


Figure 4.20: Gould Syndrome Detection Pipeline

$x' = D(z)$.

The latent space vector z from the compression model acts as the input for the diffusion model. According to the forward process of the DDPM described in equation 4.2, for any given time $t \in [0, T]$, the latent space z_t can be determined by

$$z_t = z_0\sqrt{\bar{\alpha}_t} + \epsilon_t\sqrt{1 - \bar{\alpha}_t} \tag{4.16}$$

where ϵ_t follows a normal distribution $\mathcal{N}(0, I)$.

As t increases, more Gaussian noise is added to the image, causing the latent vector z_t to lose its initial spatial structure and become similar to Gaussian noise. The reverse process follows equation 4.7.

In our segmentation task, the outcome of the DDPM reconstruction should highlight the anomaly detection area based on the mean square error.

4.4 Vascular Activity and Calcium Dynamics in Neurovascular Coupling

4.4.1 Introduction

Understanding the mechanistic basis of neurovascular coupling (NVC) is critical as it provides insights into the integrated relationship between neurons, astrocytes, and vascular cells during activity. Furthermore, NVC forms the basis of functional techniques such as functional magnetic resonance imaging (fMRI), position emission tomography that are important diagnostic tools in clinical settings. Calcium is a crucial second messenger for almost every physiological process, as such it has been commonly used as an index for various cellular activity. This is even more critical in non-excitable cells such as astrocytes. Studies have shown that NVC relies on Ca^{2+} dependent pathways that trigger the release of diffusible lipid and/or gaseous messengers that communicate with vascular cells [86–90]. As technical advances get evolved rapidly, current technologies provide incredible acquisition tools that allow investigators to critically examine Ca^{2+} dynamics at the cellular level [3]. The development of cutting-edge technologies such as two-photon microscopes accompanied by the advancement of genetically engineered calcium indicators (GECIs) have furthered our understanding of astrocyte Ca^{2+} dynamics *in vivo* under physiological conditions. In addition, NVC field has made significant progress over the past decades as neuroscientists and vascular biologists join forces with a common goal of uncovering the

mechanistic basis of NVC in a holistic approach rather than investigating individual cell types in isolation then inferring the ultimate consequential effects on the remaining cell types of the neurovascular unit. Accompanying the rapid growth of imaging modalities is the development of acquisition and analytical software [91–93]. Although these analytical tools have significantly enhanced our ability to analyze Ca^{2+} signals , they are not comprehensive and somewhat lagging the rapid evolvement of acquisition tools. Two-photon microscopes enable researchers to simultaneously monitor Ca^{2+} signals and vascular response (aka functional hyperemia) in response to stimuli *in vivo* under physiological conditions. However, to our knowledge there is no analytical software that permits the investigators to analyze both Ca^{2+} dynamics and vascular responses simultaneously in a simple, efficient, and accurate fashion. In this project, we were set out to develop an analytical tool using algorithm that semi-automates the process so that it can detect and quantify Ca^{2+} signals and vasomotor response with relatively high throughput. With this this software, named NVC_analysis, we hope to simplify the analytical process, provide the efficiency yet still retain the accuracy.

4.4.2 Challenges

In this task, challenges are based on how to find out anomalous cell information from video data. Some main challenges are list below:

- **Unknown Components Type** The shape and size of cell components are unknown in this task. It is difficult to tell whether cell components we detected is a potential useful components or a data noise point.
- **Limited Labeled Data** Because of the limited experiment results and time consuming labeling process. Only small amount of data are labeled in the whole

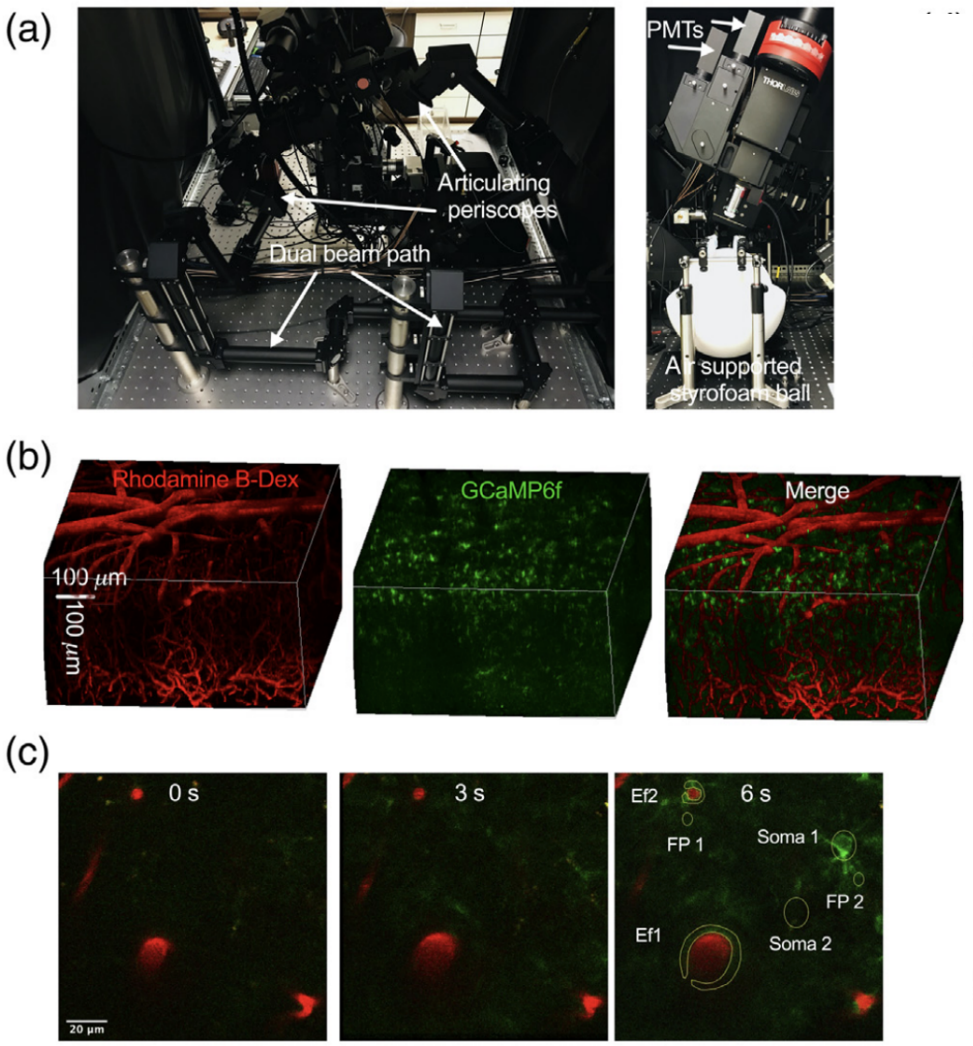


Figure 4.21: Imaging astrocytic Ca^{2+} and vascular responses to whisker stimulation using a two-photon microscope in a behaving mouse [3]. (a) Layout of a two-photon microscope for awake *in vivo* imaging with dual-beam path and articulating periscopes (left) and air-supported Styrofoam ball for a headfixed running mouse (right). (b) 3D reconstruction of the barrel cortex of a mouse showing astrocytes expressing GCaMP6f (green) and vasculature labeled with Rhodamine B-dextran (red). (c) Arteriole and astrocytic Ca^{2+} responses from different subcellular compartments to 5s whisker stimulation at different time points.

process. In this case, we are hard to completely represent the component signal based on the data we get.

- **Class Imbalance.** Strong signals only appear in very limited number of frames in the whole video process. Searching for a continuous process is hard do in the whole process.

4.4.3 Initial Method and Result

4.4.3.1 Animals

The Animal Care and Use Committee of the University of Nevada, Reno approved all the animal procedures. All studies were performed on male FVB-Tg(Aldh1l1 cre/ERT2)1Khakh/J (Jax#029655) × 129S-Gt(ROSA)26Sor$^{tm95.1(CAG-GcaMP6f)Hze}$/J (Jax#024105) between postnatal day 30 (P30) and P90. Animals were injected on 5 consecutive days with tamoxifen (75 mg/kg, Sigma), prepared as a 10 mg/mL stock in corn oil. Injections started between P21 and P35. Animals were kept on a normal 12-hour light/12-hour dark cycle and had ad libitum access to food and water.

4.4.3.2 Chronic Awake *in vivo* Preparation

All surgical procedures and isoflurane anesthesia were performed as previously described 10,11 Briefly, a head-bar was surgically installed on the animal, which was followed by a craniectomy. Bone and dura over the primary somatosensory cortex were removed and a double cover glass (i.e., 2.6 mm cover glass glued onto a 3.5 mm cover glass) was installed over the cranial window (with a smaller cover glass on top of the brain tissue). The animal was returned to its home cage to recover. Mice need to be recovered for at least 2 weeks before the first imaging session can take

place. Prior to imaging, mice were trained on a passive air-supported Styrofoam ball treadmill under head restrain for 45 minutes and habituated to whisker stimulation with an air puff on contralateral vibrissae every minute for 5s using a picospritzer III (General Valve Corp.) for two consecutive days.

4.4.3.3 Vessel Indicators

Texas Red-dextran (MW 70,000; Sigma) was injected via the tail vein (100-200μL of a 2.3% (w/v) solution in saline) to visualize the blood plasma. The animal was slightly sedated for the injection and was allowed to recover on the treadmill, with its head immobilized. The animal needs to be completely awake before imaging can take place (i.e., at least 30-minute recovery).

4.4.3.4 Two-Photon Fluorescence Imaging and Whisker Stimulations

Fluorescence images were obtained using an *in vivo* two-photon microscope illuminated with a tunable Ti:sapphire laser (Tiberius, Thorlabs), equipped with GaAsP PMTs (Hamamatsu) and controlled by ThorImage. We can see from Fig4.21. A Nikon 16X objective lens (0.8NA, 3mm WD) or an Olympus 20X objective lens (1.0NA, 2.5mm WD) was used. GCaMP6f and Texas Red dextran were excited at 920 nm. Green fluorescence signals were obtained using a 525/50nm band-pass filter, and orange/red light was obtained using a 605/70nm band-pass filter. Bidirectional xy raster scanning was used at a frame rate of 3.2Hz. Animal behaviors were captured using a near-infrared LED (780nm) and a camera at 14Hz. A 5-s air puff that deflected all whiskers on the contralateral side without impacting the face was applied using a Picospritzer while vessel surface area and astrocyte Ca^{2+} responses were monitored in the barrel cortex (layers 1-3).

4.4.3.5 Basis for the Development of the Program

The development of our software is founded on Python 3.8, incorporating critical pakages such as "numpy" for numerical computations, "opencv" for image processing, "tkinter" for the graphical user interface, "PIL" (Python Imaging Library) for image manipulation, and "scipy" for scientific and technical computing. We utilized the "PyInstaller" package to bundle our application along with all its dependencies into a single executable file, simplifying distribution and deployment.

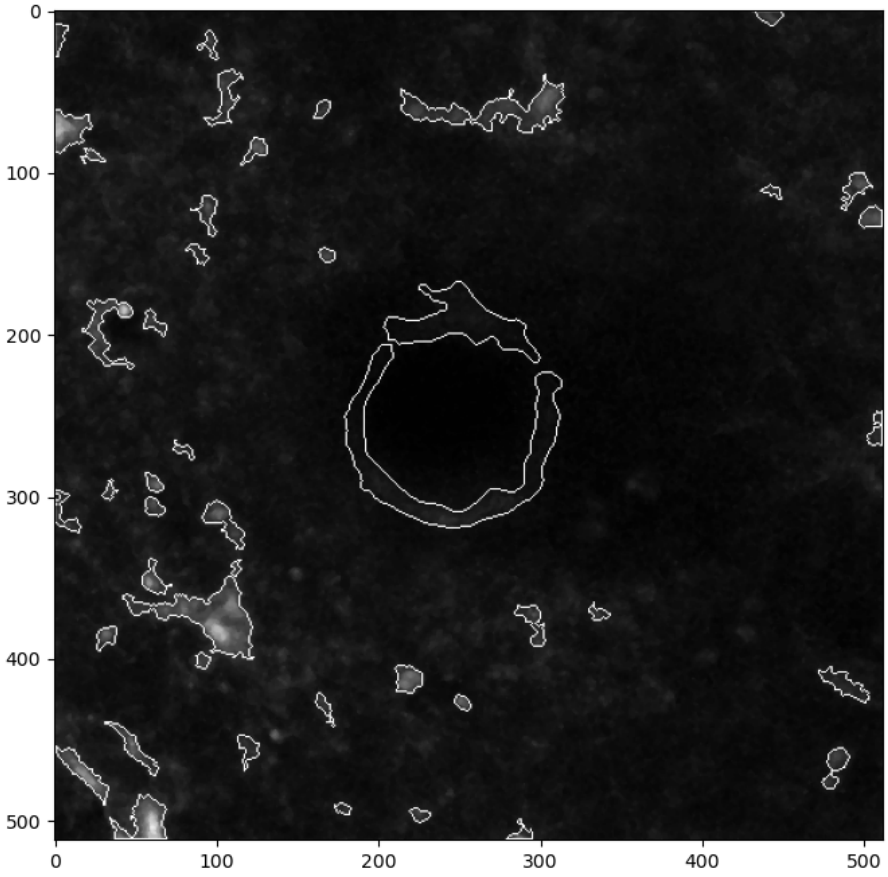


Figure 4.22: Possible cell components in the simulation

4.4.4 Software Development

Our software is developed to simultaneously analyze both astrocyte Ca^{2+} signals and vascular response. It involves 5 key procedures. Overall sturucture can be seen in Fig4.23

1. Pre-processing the image
2. Locating the vessel (e.g. penetrating arteriole)
3. Identifying the ROIs for different subcellular compartments of an astrocyte
4. Processing Ca^{2+} signals
5. Quantifying Ca^{2+} signals and vessel diameter

4.4.4.1 Pre-processing the image

Pre-processing the image can help with data visualization by enhancing the contrast and/or removing background noise and it involves 2 steps. Normalization is the initial step that is applied to both the vascular channel and Ca^{2+} signal channel. The primary propose of it is to adjust the intensity scale for better visualization of the original images. Normalization can be achieved by dividing the input image by its maximum pixel value. The next step involves histogram equalization to improve contrast by redistributing the most common intensity values. To illustrate, let's denote an image as x. We used n_k to represent the number of pixels with intensity value k. The probability of pixels at intensity value k can be calculated as

$$p_x(k) = p(x = k) = \frac{n_k}{n} \tag{4.17}$$

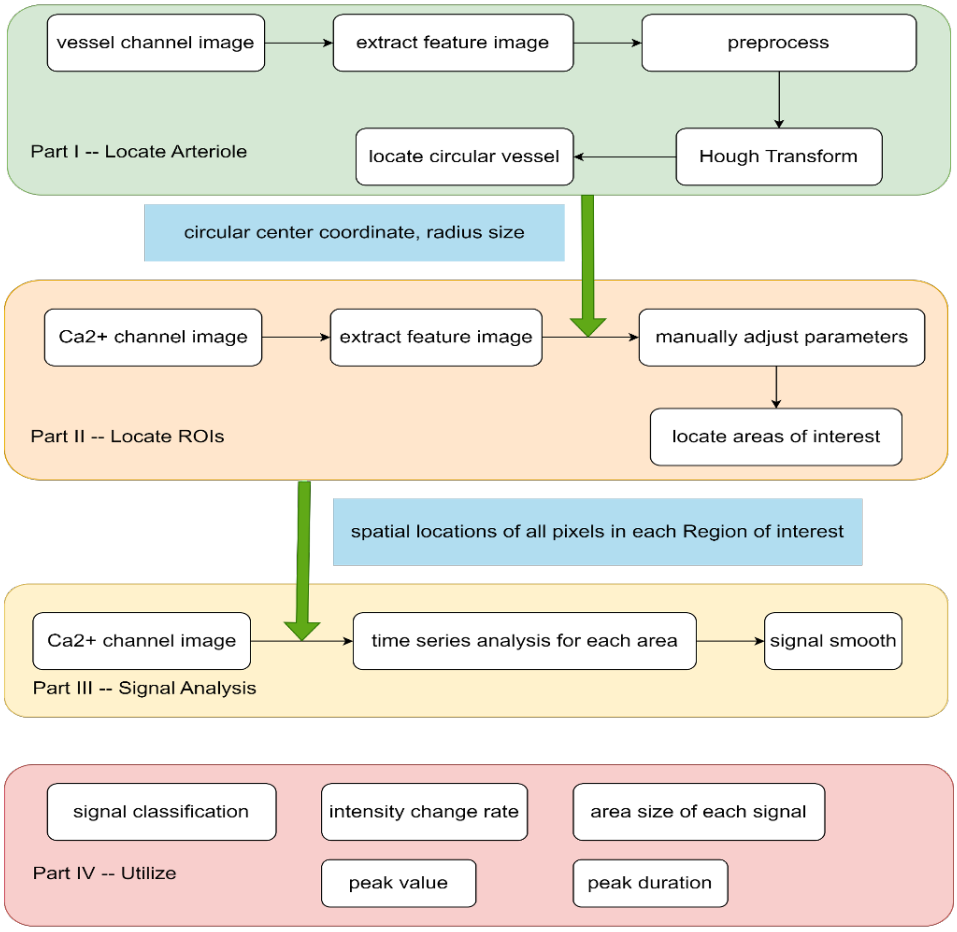


Figure 4.23: Software development structure

where n is the total number of pixels in the image.

The cumulative distribution function (CDF) for intensity value k is given by:

$$cdf_x(k) = \sum_{l=0}^{k} p_x(l) \tag{4.18}$$

We then transform image x into image y, such that $y = T(x)$ where $T(x)$ has a uniform distribution across its intensity values, ideally achieving a flat histogram. This means that each intensity value should be equally represented across the image, rather than all pixel values being identical. In other words, the transformed image should have a linear CDF across the entire range of intensity values.

$$cdf_y(k) = (k+1) \cdot \frac{1}{L} \tag{4.19}$$

Where k is an integer within a range of [0, 1, 2, ..., L-1], and L is the total number of possible intensity levels. Therefore, the transformation T maps the original image x to the new image y, ensuring that the number of pixels at each intensity level is equalized.

4.4.4.2 Locating the penetrating arteriole

Each pixel in the image can be designated as P(i,j,t), where i (row index) and j (colume index) represent the spatial coordinates of the pixel, and t denotes the time stamp. A featured image F, which retains the same row and column dimensions as the original data, is extracted by computing the maximum intensity value at each spatial location over all time stamps. This is represented as:

$$f(i,j) = \max_t P(i,j,t) \tag{4.20}$$

This featured image provides spatial information on the potential locations of the vessel. To identify the cross section of an arteriole in our recording, we employed the Hough Circle Transform [94] method to search for circular-shaped objects within this image. The Hough Circle Transform method comprises of 4 critical steps:

1. Edge Detection: Initially, the Canny edge detector [95] is applied to the featured image to delineate the boundaries of all discernible objects. This edge detection is vital for tracing the contours of vessels.

2. Circle Prediction: An algorithm iterates through every plausible circle center and radius for pixels located within the detected boundaries. For each potential circle defined by a center coordinate and a radius, a vote is cast in the corresponding cell of an accumulator matrix, which represents the spatial profile of vessel candidates.

3. Vote Accumulation: As the edge pixels are processed, votes are cumulated in the accumulator matrix for various circle configurations. This aggregation of votes assists in distinguishing between probable circle candidates.

4. Circle Detection: Subsequently, the accumulator matrix is scrutinized to pinpoint the local maxima, indicative of the detected circles. The threshold for recognizing a local maximum as a valid circle is determined by our training datasets.

By following this methodology, we can systematically locate and identify of the arterioles.

4.4.4.3 Identifying the ROIs

In this step, we ascertain the spatial locations of all Regions of Interest (ROIs) from the Ca^{2+} channel. The raw image from the Ca^{2+} channel undergoes the same pre-processing step as described previously. A featured image is extracted and subjected to histogram equalization to bolster the contrast. Furthermore, there is an option to denoise the image by either substracting background noise or using median filter [96]. The background noise subtraction technique operates under the premise that Ca^{2+} signals should not be present within the arteriole area. Having determined the location of the arteriole in the preceding step, we can calculate the average intensity within this region from the Ca^{2+} channel, designated as background noise. We then substract the calculated noise intensity from the entire channel. The Median filter, alternatively, is a non-linear technique that excels at mitigating salt-and-pepper noise. It replaces each pixel's value with the median of the intensity values from a 3×3 square kernel surrounding that pixel. It sorts the values within the kernel and adopts the median value as the new intensity for the central pixel. We utilized the unique feature of the astrocyte endfeet (i.e., ensheathing the vasculature), and the differences in signal intensity between different astrocyte subcellular compartments (e.g., endfoot vs. fine processes) to design our semi-automated detection system to identify endfoot Ca^{2+}, and fine processes and/or soma independently. This process employed two distinct parameter sets, empowering users to fine-tune the boundaries of the ROIs from different subcellular compartments independently. Each set of parameters comprises two components, t_1 and t_2. Parameter t_1 sets the minimum intensity threshold for ROI identification, ranging from 0 to 255. Parameter t_2 determines the intensity ratio relative to the surrounding area's average intensity, ranging from 0 to 1.5. A low t_1 and a high t_2 increase the software's sensitivity, and thus detect more ROIs, while the converse reduces the software's sensitivity. This dual-parameter strategy provides

precise control in pinpointing ROIs across different image regions. A separate displayed window allows users to visualize and validate the detected ROIs. Users can manually adjust the parameters to set the sensitivity for detecting ROIs.

4.4.5 Reformulation with Anomaly Detection

The main challenge in Vascular Activity and Calcium Dynamics in Neurovascular Coupling is that the Vascular signal data and Ca^{2+} data are from different imaging channels. Their background, noise level even light condition are different. We can use our multi-domain anomaly detecion model to detect vascular, cell-body, endfoot and other Ca^{2+} signal area at the same time. The structure can be seen from Fig.4.20.

The feature image x serves as the input for the compression model. It includes both vascular channel Ca^{2+} channel. the compression process is described as $z = E(x)$, while the decompression step is represented as $x' = D(z)$.

The latent space vector z, generated by the compression model, serves as the input for the diffusion model. Following the forward process of the DDPM outlined in equation 4.2, at any given time $t \in [0, T]$, the latent space z_t can be computed as

$$z_t = z_0\sqrt{\bar{\alpha}_t} + \epsilon_t\sqrt{1 - \bar{\alpha}_t} \tag{4.21}$$

where ϵ_t is sampled from the normal distribution $\mathcal{N}(0, I)$.

As t increases, progressively more Gaussian noise is incorporated into the image, causing the latent vector z_t to lose its original spatial characteristics and resemble Gaussian noise. The reverse process is governed by equation 4.7.

In our segmentation task, the result of the DDPM reconstruction should reveal the anomaly area based on the mean square error.

4.4.6 Processing Ca^{2+} signals

Once all the ROIs are identified, we proceed with a time series analysis for these areas. Each ROI is denoted as R_m, representing a set of spatial locations for the m^{th} ROI. The average intensity of the Ca^{2+} signal for each time frame is calculated as:

$$I_k(t_0) = \frac{1}{|R_m|} \sum_{(i,j)\in R_m} p(i, j, t_0) \tag{4.22}$$

Where

$$\|R_m\|$$

is the number of spatial locations in R_m, and $p(i, j, t_0)$ is the pixel intensity at location (i, j) and time t_0.

To attenuate noise and transient peaks in the time series data, we employ the Savitzky-Golay filter [97]. This digital filter is adept at smoothing the data while preserving features like peaks, troughs, and width. The filter is applied in the following steps:

1. Window Size and Polynomial Degree. We select a sliding window size of 11 data points and a polynomial function of degree 3.

2. Convolution Coefficients. The coefficients for the convolution are determined based on the chosen degree of the polynomial function and the window size. These coefficients are crucial for the least-squares fitting process.

3. Polynomial Fitting. At each position of the sliding window, we fit the selected

polynomial to the data points within the window using a least-squares method. This technique computes the optimal polynomial coefficients that minimize the sum of the squares of the differences between the actual data points and the fitted polynomial curve.

4. Smoothed Value Calculation. The smoothed value for the time series is obtained by evaluating the polynomial at the central point of each sliding window. As the window advances across the data points, we compile a complete set of smoothed values for the time series.

By systematically applying these steps, we enhance the signal quality of the time series data associated with each ROI, facilitating more accurate subsequent analyses.

4.4.6.1 Quantifying Ca^{2+} signals and vessel diameter

The determination of vessel diameter utilizes the vascular channel from our input dataset. Initially, we approximate the location of the penetrating arteriole in the second step. With the assumption that the arteriole is circular, the Hough Circle Transform method is employed to identify the center and radius of potential round-shaped signals. However, since arterioles are not perfectly circular in practical datasets, we adjust the radius to 2 times the original calculated radius, ensuring the inclusion of the entire arteriole area within the selected range.

After finding the target area around the whole vessel, another challenge is addressing the motion blur problem in measuring the vessel diameter. Motion blur is a common phenomenon in data collection, caused by the relative motion between the camera and the object during exposure time. It is particularly hard to avoid when the object is a live animal. This blur significantly impacts the detection of the vessel

area, as it enlarges the detection region, making it appear larger than it actually is. To counter this, our software uses the Wiener filter [98] to deblur the vessel region. The Wiener filter provides an estimation of the original image by minimizing the mean square error between the estimated and the true images. Subsequently, we analyze the topological structure [99] within this range to detect the actual vessel area. The true size is calculated by counting the pixel number within the vessel area and multiplying it by the "pixel size" provided by the dataset. The vessel diameter is estimated using the formula $2\sqrt{\text{Area}/\pi}$.

For each ROI identified in previous steps, we compute various metrics based on the smoothed Ca^{2+} signal. These metrics include the Signal Peak value(α), Onsite Time, Durtion Time and Peak percentage response. The baseline is established using the first twenty values of the Ca^{2+} signal, where no stimulus response is observed. We calculate the average(μ) and standard deviation(σ) of the these twenty values, and posit that a signal response occurs if the Ca^{2+} value exceeds $\mu + 3\sigma$. The Signal Peak value (α) is the highest Ca^{2+} value observed during the time frame for a given ROI. Onset Time is defined as the first frame showing a Ca^{2+} signal response, while Duration Time is the count of frames showing the Ca^{2+} signal response. Real time is deduced by multiplying the "frame rate" obtained from the dataset. Peak percentage response is determined using the formula $(\alpha - \mu)/\mu * 100\%$.

4.4.7 Execution of the Software

4.4.7.1 Detecting vessel diameter, and astrocyte Ca^{2+}

We used NVC_analysis to analyze vessel diameter changes and astrocyte Ca^{2+} transients in response to 5 s whisker stimulation. We imported a time series record-

ing acquired from Aldh1l1Cre-ERT2 x GCaMP6f mouse using Bergamo II operated by ThorImage, NVC_analysis can identify the cross section of the penetrating arteriole from the vessel channel (i.e., red channel), and ROIs astrocyte Ca^{2+} transients (i.e., green channel). By implementing background substraction or Median filter, NVC_analysis can remove background noise. By changing the "endfoot parameters" and/or "other parameter", NVC_analysis can identify different ROIs across all frames independently of each other.

4.4.7.2 Detecting vessel diameter, red blood cell flux, and astrocyte Ca^{2+}

There are recordings that we acquired included both penetrating arteriole and capillaries. In this case, NCV_analysis can simultaneously detect the cross section penetrating arteriole and counting the number of red blood cells (RBCs) from a user-defined region of the capillaries.

4.4.7.3 Quantifying the penetrating arteriole diameter, RBCs flux, and astrocyte Ca^{2+} transients

NVC_analysis automatically calculates the region size, peak value, onset time, duration and peak percent response. In the result section generated by NVC_analysis, there are arteriole diameter, RBC analysis files, and folder for endfoot, and others. Each ROIs can be viewed by clicking onto the ID of the region, a separate window will display the ROI, the curve of the Ca^{2+} signal from that particular peak.

4.4.8 Summary

We have described the development and main features of our NVC_analysis software. With this single software, we are able to detect and quantify vessel diameter changes, RBCs counts, and astrocyte Ca^{2+} dynamics from different subcellular compartments in response to whisker stimulation. Our NVC_analysis is not restricted to only the Bergamo system operated by ThorImage, but it can also be used to analyze vascular responses and astrocyte Ca^{2+} changes recorded from different imaging systems.

Chapter V

Conclusion and Future work

In this dissertation, we investigate the challenges of cross-domain anomaly detection and mitigation. We propose a novel approach for anomaly detection in multi-domain situations. This approach is applied to three different applications, including anomaly tag mitigation in supply chains, multi-class anomaly detection, and bio-image segmentation and signal analysis. Our major contributions are summarized as follows:

1. For supply chain security, we proposed a novel type of ID that is irreproducible, reliable, and applicable to most productions. A novel anomaly detection system based on the YOLO algorithm is utilized within the tag system to locate feature points at different depth levels. This structure helps users easily recognize the ID features and compare them with records in our database. This tag verification system, together with the 3D features, makes the ID tag feasible for daily commercial activities.

2. To improve the robustness of the anomaly detection system, we proposed an innovative pipeline for the multi-class anomaly detection problem. First, we used Latent Diffusion Models, which learn the latent features and a mapping from a

Gaussian distribution to a latent feature distribution simultaneously. Then, we built a classification system to determine whether the input image belongs to an anomalous class. We tested our pipeline on the CIFAR-10 and MNIST datasets, both of which showed significant improvements in experimental results.

3. For the bio-image application, an anomaly detection system is built to help reduce manual labeling time and improve accuracy. We utilized a U-net network to build an anomaly detection system, which we applied to the Gould Syndrome detection application. Our labeled results are comparable to those manually generated by a domain expert, achieved with minimal training data. In applications involving Vascular Activity and Calcium Dynamics, our system automatically detects signal changes in Ca^{2+} and vascular activity.

In the future, our multi-domain anomaly detection pipeline will be used to address questions involving mixed information. For example, temperature-sensitive materials could be added to our 3D unclonable tags. In this case, our model will need to consider both tag shape feature information and the changes in temperature-sensitive materials. The next step for our multi-class anomaly detection is to utilize possible language information combined with the image. Innovative language-image pretrained models like CLIP have the potential to improve anomaly detection results by adding more language information to image anomaly detection. In bio-image analysis, various signals can be considered together to analyze single neuronal activity.

Bibliography

[1] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*. IEEE, 2018, pp. 1–5.

[2] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[3] C. H. T. Tran, "Toolbox for studying neurovascular coupling in vivo, with a focus on vascular activity and calcium dynamics in astrocytes," *Neurophotonics*, vol. 9, no. 2, p. 021909, 2022.

[4] C. Wang, L. Raymond, Y. Jin, A. Tavakkoli, and H. Shen, "3d unclonable optical identity for universal product verification," in *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2021, pp. 136–146.

[5] C. Wang and A. Tavakkoli, "Latent diffusion based multi-class anomaly detection," in *International Symposium on Visual Computing*. Springer, 2023, pp. 487–498.

[6] A. Boukerche, L. Zheng, and O. Alfandi, "Outlier detection: Methods, models, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–37, 2020.

[7] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[8] W. A. Shewhart, *Economic control of quality of manufactured product*. Macmillan And Co Ltd, London, 1931.

[9] P. F. Velleman and D. C. Hoaglin, *Applications, basics, and computing of exploratory data analysis*. Duxbury Press, 1981.

[10] M. Frigge, D. C. Hoaglin, and B. Iglewicz, "Some implementations of the boxplot," *The American Statistician*, vol. 43, no. 1, pp. 50–54, 1989.

[11] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, "Informal identification of outliers in medical data," in *Fifth international workshop on intelligent data analysis in medicine and pharmacology*, vol. 1. Citeseer, 2000, pp. 20–24.

[12] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," 2000.

[13] D. Peel and G. J. McLachlan, "Robust mixture modelling using the t distribution," *Statistics and computing*, vol. 10, no. 4, pp. 339–348, 2000.

[14] D. Dasgupta and F. Nino, "A comparison of negative and positive selection algorithms in novel pattern detection," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, vol. 1. IEEE, 2000, pp. 125–130.

[15] M. V. Mahoney and P. K. Chan, "Learning nonstationary models of normal network traffic for detecting novel attacks," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 376–385.

[16] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 275–300, 2004.

[17] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[18] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng, "Learning deep energy models," in *ICML*, 2011.

[19] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[20] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching." *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.

[21] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 681–688.

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[23] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep boltzmann machines," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 693–700.

[24] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[25] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 287–296.

[26] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] J. Andrews, T. Tanay, E. J. Morton, and L. D. Griffin, "Transfer representation-learning for anomaly detection." JMLR, 2016.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[32] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai, "Self-trained deep ordinal regression for end-to-end video anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 173–12 182.

[33] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *arXiv preprint arXiv:1510.01553*, 2015.

[34] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[35] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

[36] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 36–51.

[37] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 90–98.

[38] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 665–674.

[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[40] A. Makhzani and B. Frey, "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[41] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive autoencoders: Explicit invariance during feature extraction," in *Icml*, 2011.

[42] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient gan-based anomaly detection," *arXiv preprint arXiv:1802.06222*, 2018.

[43] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[44] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[45] D. M. Tax and R. P. Duin, "Support vector domain description," *Pattern recognition letters*, vol. 20, no. 11-13, pp. 1191–1199, 1999.

[46] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.

[47] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[48] P. Wu, J. Liu, and F. Shen, "A deep one-class neural network for anomalous event detection in complex scenes," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2609–2622, 2019.

[49] B. Zhang and W. Zuo, "Learning from positive and unlabeled examples: A survey," in *2008 International Symposiums on Information Processing*. IEEE, 2008, pp. 650–654.

[50] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *Machine Learning*, vol. 109, no. 4, pp. 719–760, 2020.

[51] C. Wang, C. Ding, R. F. Meraz, and S. R. Holbrook, "Psol: a positive sample only learning algorithm for finding non-coding rna genes," *Bioinformatics*, vol. 22, no. 21, pp. 2590–2596, 2006.

[52] S. Chaudhari and S. Shevade, "Learning from positive and unlabelled examples using maximum margin clustering," in *International Conference on Neural Information Processing*. Springer, 2012, pp. 465–473.

[53] F. He, T. Liu, G. I. Webb, and D. Tao, "Instance-dependent pu learning by bayesian optimal relabeling," *arXiv preprint arXiv:1808.02180*, 2018.

[54] C. Scott, "A rate of convergence for mixture proportion estimation, with application to learning from noisy labels," in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 838–846.

[55] A. Menon, B. Van Rooyen, C. S. Ong, and B. Williamson, "Learning from corrupted binary labels via class-probability estimation," in *International conference on machine learning*. PMLR, 2015, pp. 125–134.

[56] K. Tian, S. Zhou, J. Fan, and J. Guan, "Learning competitive and discriminative reconstructions for anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5167–5174.

[57] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[58] E. M. Knorr and R. T. Ng, "Finding intensional knowledge of distance-based outliers," in *Vldb*, vol. 99. Citeseer, 1999, pp. 211–222.

[59] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2041–2050.

[60] H. Wang, G. Pang, C. Shen, and C. Ma, "Unsupervised representation learning by predicting random distances," *arXiv preprint arXiv:1912.12186*, 2019.

[61] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern recognition letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[62] M.-F. Jiang, S.-S. Tseng, and C.-M. Su, "Two-phase clustering process for outliers detection," *Pattern recognition letters*, vol. 22, no. 6-7, pp. 691–700, 2001.

[63] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[64] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[65] L. Deecke, L. Ruff, R. A. Vandermeulen, and H. Bilen, "Transfer-based semantic anomaly detection," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2546–2558.

[66] Z. You, L. Cui, Y. Shen, K. Yang, X. Lu, Y. Zheng, and X. Le, "A unified model for multi-class anomaly detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4571–4584, 2022.

[67] W. Lu, Y. Cheng, C. Xiao, S. Chang, S. Huang, B. Liang, and T. Huang, "Unsupervised sequential outlier detection with deep architectures," *IEEE transactions on image processing*, vol. 26, no. 9, pp. 4321–4330, 2017.

[68] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[69] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[70] M. S. Graham, W. H. Pinaya, P.-D. Tudosiu, P. Nachev, S. Ourselin, and J. Cardoso, "Denoising diffusion models for out-of-distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2947–2956.

[71] J. P. Ruppert, R. C. Fish, T. A. Yap, and R. M. Ames, "Portable rf id tag and barcode reader," Jun. 17 1997, uS Patent 5,640,002.

[72] R. Hou, G. Zhang, H. Xu, Y. Zhou, and S. Colavito, "Laser barcode scanner," Jun. 9 2015, uS Patent 9,053,378.

[73] Y. Liu, J. Yang, and M. Liu, "Recognition of qr code with mobile phones," in *2008 Chinese control and decision conference*. IEEE, 2008, pp. 203–206.

[74] R. Bhattacharyya, C. Floerkemeier, and S. Sarma, "Low-cost, ubiquitous rfid-tag-antenna-based sensing," *Proceedings of the IEEE*, vol. 98, no. 9, pp. 1593–1600, 2010.

[75] J. A. Hayward and J. Meraglia, "Dna marking and authentication: A unique, secure anti-counterfeiting program for the electronics industry," in *International Symposium on Microelectronics*, vol. 2011, no. 1. International Microelectronics Assembly and Packaging Society, 2011, p. 000.

[76] W. Ren, G. Lin, C. Clarke, J. Zhou, and D. Jin, "Optical nanomaterials and enabling technologies for high-security-level anticounterfeiting," *Advanced Materials*, vol. 32, no. 18, p. 1901430, 2020.

[77] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *2007 44th ACM/IEEE Design Automation Conference*. IEEE, 2007, pp. 9–14.

[78] Ganoksin, "3 ways to use 2-part epoxy resins - ganoksin jewelry making community," ONLINE, https://www.ganoksin.com/article/3-ways-to-use-2-part-epoxy-resins/ Accessed: 30-Apr-2021.

[79] Micro, "Vu precision measurement equipment," ONLINE, https://www.microvu.com/machines/vertex Accessed: 30-Apr-2021.

[80] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of hough transform methods for circle finding," *Image and vision computing*, vol. 8, no. 1, pp. 71–77, 1990.

[81] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[82] J. Campbell, "Complete casting handbook: Metal casting processes," *Metallurgy, Techniques and Design*, vol. 2, p. 244, 2011.

[83] J.-U. Park, M. Hardy, S. J. Kang, K. Barton, K. Adair, D. kishore Mukhopadhyay, C. Y. Lee, M. S. Strano, A. G. Alleyne, and J. G. Georgiadis, "High-resolution electrohydrodynamic jet printing," *Nature materials*, vol. 6, no. 10, pp. 782–789, 2007.

[84] M. Wang, *Lithography*. InTech, 2010.

[85] R. Menon, A. Patel, D. Gil, and H. I. Smith, "Maskless lithography," *Materials Today*, vol. 8, no. 2, pp. 26–33, 2005.

[86] D. Attwell, A. M. Buchan, S. Charpak, M. Lauritzen, B. A. MacVicar, and E. A. Newman, "Glial and neuronal control of brain blood flow," *Nature*, vol. 468, no. 7321, pp. 232–243, 2010.

[87] P. Thakore, M. G. Alvarado, S. Ali, A. Mughal, P. W. Pires, E. Yamasaki, H. A. Pritchard, B. E. Isakson, C. H. T. Tran, and S. Earley, "Brain endothelial cell trpa1 channels initiate neurovascular coupling," *Elife*, vol. 10, p. e63040, 2021.

[88] J. A. Filosa, A. D. Bonev, S. V. Straub, A. L. Meredith, M. K. Wilkerson, R. W. Aldrich, and M. T. Nelson, "Local potassium signaling couples neuronal activity to vasodilation in the brain," *Nature neuroscience*, vol. 9, no. 11, pp. 1397–1403, 2006.

[89] Y. Shi, X. Liu, D. Gebremedhin, J. R. Falck, D. R. Harder, and R. C. Koehler, "Interaction of mechanisms involving epoxyeicosatrienoic acids, adenosine receptors, and metabotropic glutamate receptors in neurovascular coupling in rat whisker barrel cortex," *Journal of Cerebral Blood Flow & Metabolism*, vol. 28, no. 1, pp. 111–125, 2008.

[90] C. Lecrux and E. Hamel, "Neuronal networks and mediators of cortical neurovascular coupling responses in normal and altered brain states," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 371, no. 1705, p. 20150350, 2016.

[91] A. Agarwal, P.-H. Wu, E. G. Hughes, M. Fukaya, M. A. Tischfield, A. J. Langseth, D. Wirtz, and D. E. Bergles, "Transient opening of the mitochondrial permeability transition pore induces microdomain calcium transients in astrocyte processes," *Neuron*, vol. 93, no. 3, pp. 587–605, 2017.

[92] R. Srinivasan, B. S. Huang, S. Venugopal, A. D. Johnston, H. Chai, H. Zeng, P. Golshani, and B. S. Khakh, "Ca2+ signaling in astrocytes from ip3r2-/- mice in brain slices and during startle responses in vivo," *Nature neuroscience*, vol. 18, no. 5, pp. 708–717, 2015.

[93] Y. Wang, N. V. DelRosso, T. V. Vaidyanathan, M. K. Cahill, M. E. Reitman, S. Pittolo, X. Mi, G. Yu, and K. E. Poskanzer, "Accurate quantification of astrocyte and neurotransmitter fluorescence dynamics for single-cell and population-level physiology," *Nature neuroscience*, vol. 22, no. 11, pp. 1936–1944, 2019.

[94] P. Mukhopadhyay and B. B. Chaudhuri, "A survey of hough transform," *Pattern Recognition*, vol. 48, no. 3, pp. 993–1010, 2015.

[95] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[96] G. R. Arce, *Nonlinear signal processing: a statistical approach*. John Wiley & Sons, 2005.

[97] W. H. Press and S. A. Teukolsky, "Savitzky-golay smoothing filters," *Computers in Physics*, vol. 4, no. 6, pp. 669–672, 1990.

[98] P. Biswas, A. S. Sarkar, and M. Mynuddin, "Deblurring images using a wiener filter," *International Journal of Computer Applications*, vol. 109, no. 7, pp. 36–38, 2015.

[99] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.