Semi-Supervised Deep Learning Approaches for Anomaly Detection in Computer Vision

JACK WILLIAM BARKER

Semi-Supervised Deep Learning Approaches for Anomaly Detection in Computer Vision

Jack W. Barker

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
31st March 2023

Abstract

Anomalies are samples which differ significantly from ordinary appearance or behaviours to such a degree that they lay outside what is considered standard in a given task. Deviations may be due to defective or broken regions of a sample, or due to foreign objects present in samples. Detecting such deviations in samples is the task of anomaly detection. In the task of X-Ray Security Scanning or Factory Line Inspection, missing the detection of anomalous instances, especially in the former, can cause catastrophic impact to safety. Missing anomalies within tasks such as Plant Disease Detection or Wind Turbine Blade Fault Detection are likely to cause increased detriment to the assets of these tasks if they are not caught soon enough. The work presented in this thesis aims to push towards automation of the detection of anomalies in such critical tasks. Firstly, an extensive review is conducted into prior approaches and paradigms which have been presented for anomaly detection. As most tasks in visual anomaly detection do not have the luxury of having copious and diverse anomalous samples, if any, methods have since shifted to semi-supervised learning whereby training is conducted solely across non-anomalous samples. An obvious problem with such training is the detection of subtle anomalies (deviations which vary only slightly from normality) in a given task. This was the motivation behind the PANDA architecture, a generative semi-supervised method presented in this thesis. This method, specifically designed to detect subtle and coarse anomalies obtains state-of-the-art results in AUC score across a substantial pool of challenging datasets. Following from this, a trend in anomaly detection has seen denoising approaches obtaining state-of-the-art and robustness to the task, however, such noising approaches are manually defined and random by nature. This thesis presents a method to add optimised, custom noise for any given anomaly detection task. The results of this method show that even a very basic architecture can obtain close to state-of-the-art performance when using this unique noising approach. Finally, an approach to detect faults in wind turbine blades is introduced in the form of a two-stage detection approach which first establishes a more accurate method of blade detection and extraction compared with prior object detection approaches, and then uses off-the-shelf anomaly detection methods to perform successful defect detection of super-pixel sub-regions of the detected blades.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

Conducting a PhD is a demanding and challenging part of one's life, this was made ever more difficult by the sudden introduction of the COVID-19 pandemic and the widespread lockdowns and restrictions it imposed.

Firstly, I express my sincerest gratitude to my supervisor, Professor Toby Breckon. His supportive and approachable management style helped during this challenging period. I will forever be indebted for both the imparted knowledge I have acquired during valuable and insightful discussions together, and the time and effort he put into ensuring my success throughout the PhD. This has enabled me to push myself and produce work that I can be proud of and will have an impact for future research in this field.

I extend my deepest gratitude to my family for their unwavering boundless support and patience, not only during the time of my PhD, but throughout my entire educational journey. In particular, I thank my sister for being there anytime I needed for support and words of encouragement as well as my mother and father for their selflessness and hard work while raising my sister and I, which set the foundational bedrock for everything that we have achieved. I am also thankful to my girlfriend Isabella who has been my rock while writing this thesis and offered me kindness, support, and encouragement throughout which helped me get through this difficult period.

Also, I am thankful to have met so many close friends during my time at Durham with whom I have countless fond memories and with whom I hope we will continue the friendship long into the future.

I would also like to thank my sponsors for this PhD, Ørsted and Engineering and Physical Sciences Research Council (EPSRC) for their funding for this project.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The work in this thesis contributes to the field of reconstruction-based anomaly detection applied to real-world tasks. On a high level, anomaly detection is the task of recognising samples of a given dataset which deviate significantly from established normality and as such, represent unexpected eventualities or outliers in the scope of a given task. Anomaly detection is a challenging task because of the broad range of variational forms which anomalies may present, representing an unbounded (open-set) distribution of possible deviations from normality. Anomalies may present as defective objects within samples, or as incongruous events during inference. As such, anomaly detection methods must recognise and detect such unseen out-of-distribution samples during inference by learning effective knowledge obtained from seen in-distribution examples during training.

Figure 1.1 illustrates some of the real-world anomaly detection tasks used in the course of this thesis. In the task of X-Ray Security Scanning or Factory Line Inspection, missing the detection of anomalous instances, especially in the former, can cause catastrophic impact to safety. For this reason, human operators who are tasked with detecting anomalous items within these tasks have to be rigorously trained and examined in order to perform their job. Deep anomaly detection approaches could act as a tool for human operators to assist in the detection of anomalous samples; To essentially act as a 'second pair of eyes', both reassuring the human operators that they have not made a mistake during categorisation and catching anything that may have slipped past the operator. Such a system may assist in detecting anomalous instances which may be missed more frequently by human operators during busy periods, or while fatigued at the end of a long shift. Missing anomalies within tasks such as Plant Disease Detection or Wind Turbine Blade Fault Detection are likely to cause increased detriment to the assets of these tasks over time if they are not caught soon enough. To this end, the sooner these anomalies are caught, then the less costly they will be to rectify. Success at this task relies on detecting subtle anomalies which deviate only slightly from normality. Approaches should be able to detect diseased regions, or cracks in turbine blades forming to enable the eradication of the disease so as to not let the entire crop become affected in the former, and

repairing small cracks in turbine blades is far cheaper than replacing the entire wind turbine blade due to the crack spreading throughout the full blade in the latter.

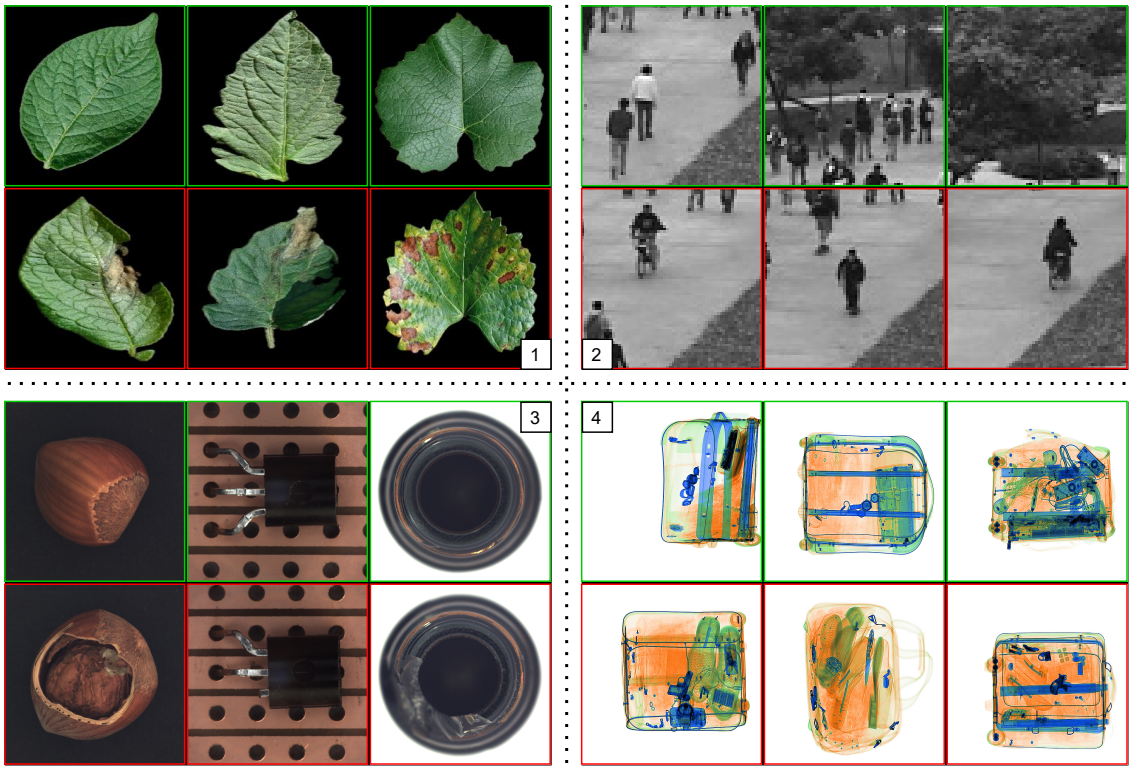

Figure 1.1: Visual samples from anomaly detection tasks: **(1)**Plant Village, **(2)** UCSDPed1, **(3)**MVTEC, **(4)**Durham Threat Item X-Ray dataset. With each segment illustrating both normal, non-anomalous samples (*top row*), together with their anomalous counterparts (*bottom row*) across the tasks

For meaningful real-world application of anomaly detection methods the type I and type II errors or commonly referred false-positive and false-negative errors respectively, must be reduced. Given the null hypothesis, stated as 'the presented sample is non-anomalous'. Type I error occurs when this correct null hypothesis is erroneously rejected such that a normal sample is categorised as anomalous. The type II error is the failure to reject the false null hypothesis leading to anomalous samples being categorised as normal.

Classes within tasks which have high intra-class variance between examples of the normal class can pose a challenge to detect as the features of the instances do not correlate well to one another and so meaningful representation is more difficult

to obtain. The severe diversity in appearance of normal class instances can lead to increased chance of type I error due to such normal class objects being divergent enough to severely increase the given anomaly score, resulting in a wrongful anomalous categorisation. Low inter-class variance occurs in a given task between the normal and anomalous class when the presented anomalies are visually subtle and the distinction between normal and anomalous instances is challenging. This can lead to increased chance of type II errors in categorisation such that anomalous samples are incorrectly classified as normal.

This difficulty is illustrated visually in Figure 1.2. The tomato class within the Plant Village dataset [1] is shown containing samples which have both high intra-class and low inter-class variance. The normal class contains high variability in shape, colour, and texture, whereas the anomalous examples are visually subtle and hardly noticeable. This is especially true when observing the instance containing Bacterial Spot disease where the leaf looks indistinguishable from the healthy leaf.

Our work proposes methods to improve both the detection capability and robustness of reconstruction-based anomaly detection methods. We also evaluate the approaches over datasets of significant importance which can benefit from automation to ease the burden of human operators.

1.1 Motivation

Humans have intuitive skill to recognise deviations from normality in the real world even when never exposed to specific anomalous examples. Deeply rooted through primal instinct, the ability acts as a defence mechanism to avoid danger, triggering the fight or flight response in abnormal situations. A deep knowledge and visual understanding of the world allows us to notice when something seems peculiar. For certain tasks, however, such as factory line inspection or X-Ray security scanning, they require intense training of operators in order to gain expert level knowledge.

Human operators within aviation X-Ray security scanning are required to pass the rigorous X-Ray Competency Assessment Test (X-RAY CAT) [22] under Euro-

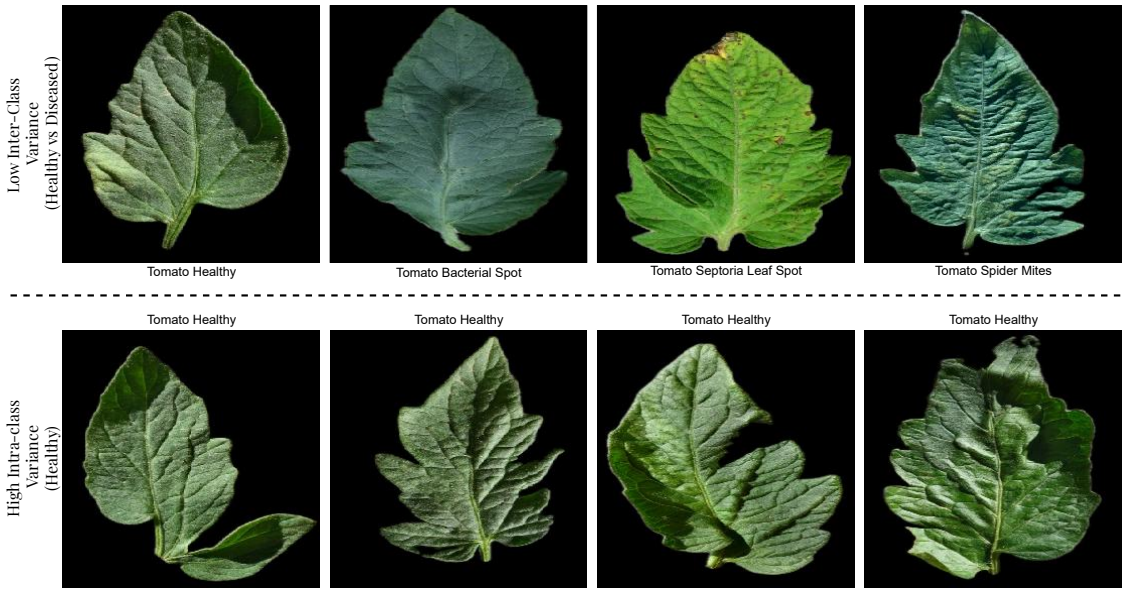

Figure 1.2: Examples from the Plant Village dataset [1] illustrating the low inter-class and high intra-class variance present between samples in certain classes of visual anomaly detection tasks.

pean Civil Aviation Conference document 30 [23] as well as ICAO Annex 17 [24] and EC300 Article 10 [25]. The test conducted by the Federal Aviation Authority in 1987 showed that human operators missed 20% of threat items presented [26]. A further test in 2002 by the Transport Security Administration conducted at Los Angeles Airport showed that security operators failed to detect weapons in 41% of cases [27]. This, together with human operators suffering from fatigue on long shifts [28] motivates the need for automation in such tasks. Of particular interest is the detection of threats concealed inside electronic devices such as laptops [3]. These threat items are far more obfuscated by the visually complex inner electronics of the laptops compared to more coarse threats present in common baggage and as such will have a lower chance of detection than those reported in the aforementioned tests [27,28]. Examples of this obfuscation is illustrated in Figure 1.3. The weapons from the University X-Ray Baggage Anomaly dataset on the top row stand out considerably within the baggage even in more challenging orientations. Observing the threat items concealed within the large electronics on the bottom row, however, are visually very difficult to detect.

It is the goal of anomaly detection methods to be able to detect anomalous instances with these properties as they pose as a more difficult task than merely detecting strong outliers such as those present in leave-one-out tasks [7, 8].

Chapter 2 outlines more motivation in the specific tasks of X-Ray aviation security in large electronics [3], plant leaf disease detection [1], pedestrian footpath monitoring [10], factory line inspection [9], and wind turbine blade fault detection [29] as to the need for assistive automation to better detect visual anomalies in these tasks.

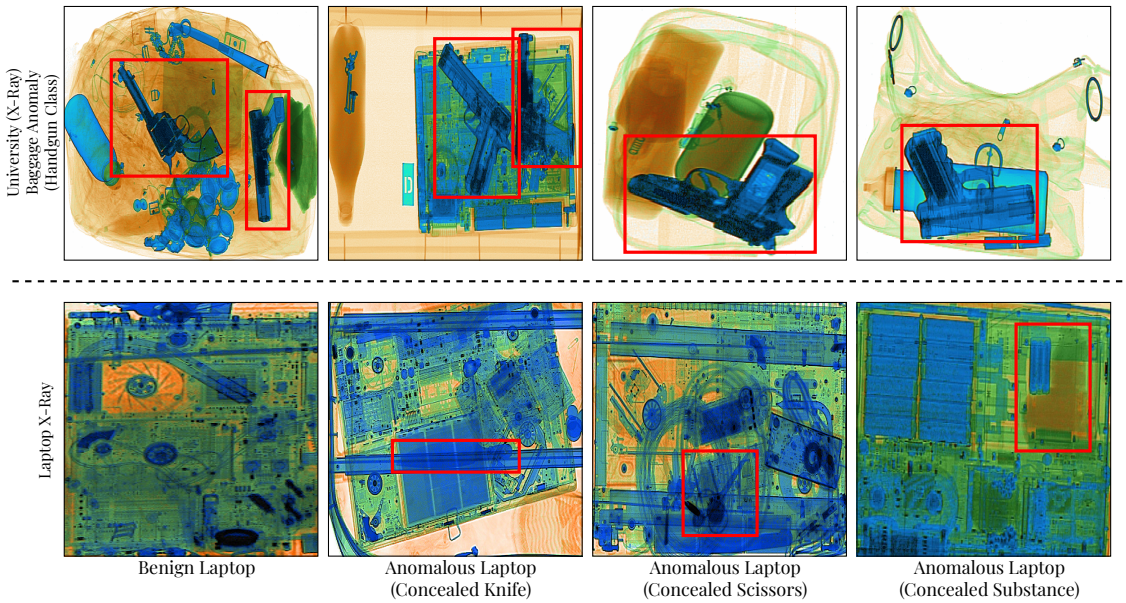

Figure 1.3: Examples from the University (X-Ray) Baggage Anomaly dataset [2] (top) and the Laptop X-Ray dataset [3](bottom) showing the relative difficulty of detecting threat items concealed within large electronics devices (bottom) compared to detecting handguns (top) within baggage. Bounding boxes show location of threat items in each image.

1.2 Thesis Contributions

The contributions of this thesis are as follows:

- A novel method for the semi-supervised detection of anomalies in visual data with increased focus on capturing visually subtle anomalies in real-world example tasks (Chapter 3. We utilise an asymmetric autoencoder generator architecture (Section 3.2.1), trained adversarially with a novel 'fine-grained' discriminator module (Section 3.2.2. Our modifications to improve performance include residually connected dual level feature extractors within the generator module and a perceptual loss function (Section 3.2.3).

- An application of semi-supervised methods to the task of detecting visual surface faults in glass-fibre turbine blades (Chapter 5). Our approach is a dual-stage architecture to firstly, extract blade parts (Section 5.2.1)from a given image with better detection and mask prediction accuracy than prior methods [18–20]. Secondly, the detection of faults is performed with a collection of well-established methods of semi-supervised anomaly detection as well the introduction of U-GANomaly 5.2.4), an upgrade to Skip-GANomaly [6] by utilising a U-Net [30] discriminator [17]. Our experiments show that U-GANomaly outperforms prior methods for semi-supervised anomaly detection.

- A novel approach to training a more robust autoencoder with the use of an adversarial training scheme whereby a denoising autoencoder is challenged to reverse the impact of added learned adversarial noise and corruption to the original data (Chapter 4). We show in our experiments that this training method Significantly improves the performance of the vanilla autoencoder model across real-world tasks and outperforms [31–35] across the task of novel leave-one-out anomaly detection.

1.3 Publications

Work presented in this thesis has been subject to peer review and subsequently accepted for publication in well-established proceedings and outlined in the respective chapters of this thesis as follows:

- **Evaluation of a Dual Convolutional Neural Network Architecture for Object-wise Anomaly Detection in Cluttered X-ray Security Imagery**, Y.F.A. Gaus, N. Bhowmik, S. Akcay, P.M. Guillen-Garcia, J.W. Barker, T.P. Breckon, In Proceedings of the International Joint Conference on Neural Networks, IEEE, 2019. (Introductory research to the task of anomaly detection. Contributes to the problem definition in Chapter 1).

- **On the Impact of Object and Sub-component Level Segmentation Strategies for Supervised Anomaly Detection within X-ray Security Imagery**, N. Bhowmik, Y.F.A. Gaus, S. Akcay, J.W. Barker, T.P. Breckon, In Proceedings of the Conference on Machine Learning and Applications, IEEE, 2019, pp. 986-991. (Contributing to Chapter(s) 3)

- **PANDA: Perceptually Aware Neural Detection of Anomalies**, J.W. Barker, T.P. Breckon, In Proceedings of the International Joint Conference on Neural Networks, IEEE, 2021. (Contributing to Chapter(s) 3,5)

- **Semi-Supervised Surface Anomaly Detection of Composite Wind Turbine Blades From Drone Imagery**, J.W. Barker, N. Bhowmik, T.P. Breckon, In Proceedings of the International Conference on Computer Vision Theory and Applications, 2022. (Contributing to Chapter(s) 5)

- **Robust Semi-Supervised Anomaly Detection via Adversarially Learned Contrastive and Continuous Noise Generation**, J.W. Barker, N. Bhowmik, Y.F.A. Gaus, T.P. Breckon, In Proceedings of the International Conference on Computer Vision Theory and Applications, 2023. (Contributing to Chapter(s) 4)

1.4 Thesis Scope and Structure

Chapter 2 provides a thorough review of the literature within deep visual anomaly detection in images outlining works within the paradigms of probabilistic (Section 2.2), classification (Section 2.3), and reconstruction based approaches (Section 2.3). We then provide an overview of denoising approaches applied to denoising autoencoder methods (Section 2.4.2) which sets the stage for the work presented within Chapter 4 of this thesis. We provide an overview of prior works up until current state-of-the-art within visual anomaly detection.

The work presented in this thesis tackles the challenging and on-going task of accurate detection of anomalous instances within real-world tasks. Of particular interest is the detection of visually subtle anomalies which deviate minimally from normality. Chapter 3 presents a semi-supervised adversarially trained autoencoder architecture bespoke to the detection of such anomalies, achieving state-of-the-art performance during extensive evaluation across a wide array of multi-spectral, multi-modal datasets.

Chapter 4 introduces a novel approach of training a simple denoising autoencoder by adding continuous adversarially learned global noise to images prior to denoising. This work improves on the performance of prior state-of-the-art methods [31, 32, 36] in applying noise methods to denoising autoencoders while being significantly simpler to implement. Our method of continuous global adversarial noise production leads to the creation of a more robust denoising autoencoder and improves anomaly detection capability as shown in our results.

Chapter 5 presents an architecture to the task of detecting surface faults in glass-fibre turbine blades. This solution features a two stage process in which blade parts are initially extracted from input images with high accuracy (Section 5.2.1). These blade parts are then processed with a suite of semi-supervised anomaly detection methods (Section 5.2.3). We also introduce a new anomaly detection architecture (Section 5.2.4) in this chapter which achieves state-of-the-art performance when compared to prior methods in our experiments.

CHAPTER 2

Literature Review

2.1 Introduction

Anomaly detection is the task of recognising deviations from pre-established normality in presented examples in a given task or domain. Accurate and efficient detection of anomalous deviations is crucial for oftentimes critical tasks. Tasks such as factory line quality control inspection or X-ray security screening require specialist rigorous training of human operators to conduct. Automation of anomaly detection in these tasks would be an invaluable tool for current human operators who can suffer from fatigue, boredom or common human error while working long shifts.

In visual data, anomalies may present as defects or deviations in objects from the trained domain, or may instead present as foreign objects which are out-of-distribution from the prior training domain. Examples from the former case as seen in the MVTEC dataset [9] can either feature 1) prominent deviations in which large sections of the objects are either corrupted to a significant degree or are missing leading to visually obvious anomalies or 2) visually subtle deviations which vary slightly from normality and as such lay close to the decision boundary between normal and anomalous categorisation; such samples are challenging to differentiate from training data, increasing the chance of type 1 and type 2 errors occurring during categorisation. Examples of such figures are illustrated in Figure 2.1.

Tasks which instead feature foreign objects as anomalous instances such as X-ray security scanning [3, 37] and pedestrian area monitoring [10] contain more open-set variability of both the normal and anomalous data, making the distinction between normal and anomalous samples a challenging task. Due to this, normal items in X-ray baggage scans which appear strange, but pose no threat and are permitted may be wrongfully flagged as anomalous. The set of anomalous items which pose a threat are a subset region of the open-set distribution of anomalous space and without post-categorising anomalous examples, many false positives would be flagged. This makes these tasks notoriously difficult for anomaly detection methods.

Anomaly detection methods generally train solely across normal data as in many real-world tasks, anomalies are rare occurrences with potential high visual variance.

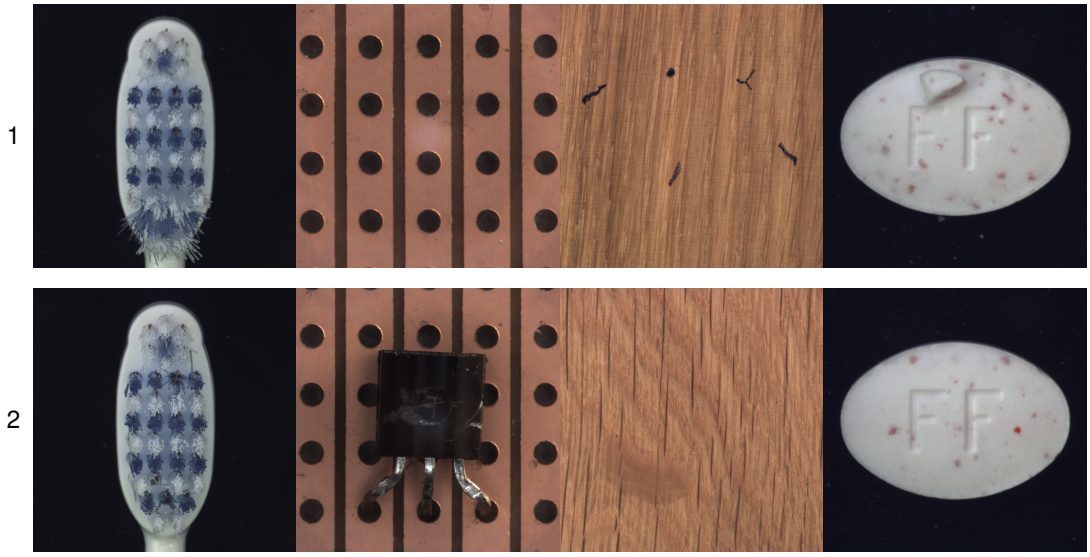

Figure 2.1: Defect examples from MVTEC across Toothbrush, Transistor, Wood and Pill classes illustrating, Row 1: Visually obvious defects, and Row 2: Visually subtle defects.

The collection of normal data is cheap and far more similar to a human approach whereby a human can recognise deviations from normality after being subjected to a set of normal examples [2].

Prior work in anomaly detection can be generally categorised into three categories: probabilistic approaches, classification approaches, and reconstruction approaches [38, 39].

This chapter reviews prior literature primarily in the field of reconstruction-based generative anomaly detection to fit with the research presented in this thesis. A brief overview of works featuring distributional and classification-based approaches will also be provided to give an overview of the field of visual anomaly detection as a whole.

2.2 Probabilistic Approaches

Probabilistic approaches [40] are based on estimating a generative probabilistic model of the underlying data [41]. This involves estimating the Probability Density Function (PDF) of the data during training [42] whereby test samples with low

probabilistic likelihood are likely to be abnormal [43] when measured against this learned distribution established during training. It is expected that abnormal samples presented during inference will not meet such a distribution and as such will be classified as anomalous. Approaches can be split into two paradigms: Parametric approaches such as Gaussian Mixture Models (GMM) [44] and non-parametric Kernel Density Estimators [45–47]; It is, however, well-established that these approaches frequently become highly sensitive across high-dimensional spaces due to the problem of the 'curse of dimensionality' [48]. In visual anomaly detection across high-dimensionality images, this means that any input sample could be a rare event with low probability to observe and as such be flagged as anomalous by a model [49]. Frequently, this is overcome by performing density estimation in feature space across an embedded latent space representation following encoding [50–52] as feature space is considerably easier to conform with the probabilistic assumption [39] of the data and offers more freedom for the model to categorise than sparsely-distributed raw image pixel data [53]. This method of encoding to feature space initially required a two-stage process of first performing dimensionality reduction and then performing density estimation on this low-dimensional latent space [54]. The Deep Autoencoding Gaussian Mixture Model (DAGM) [48] jointly performs these stages to perform both low-dimensionality mapping and distribution modelling in the same stage. Cohen *et al.* [55] state that an approach implementing nearest neighbours can be described as a density estimation approach; they propose the SPADE [55] approach which performs K-nearest neighbours using the Euclidean distance between a learned memory-bank of nominal image-level feature representations and extracted features of test query images. Although successful at both detection and segmentation of anomalies, the linear complexity of the KNN algorithm increases in time and space complexity as the dataset grows [56]. This issue is overcome by the PaDiM approach [56] by modelling deep features collected at different depths of a pretrained network of image patches as a multivariate Gaussian distribution which models correlations between semantic levels of the pretrained network. The PatchCore [57] approach extends the work of PaDiM and SPADE by still using a memory-bank of

non-anomalous feature representations, but implements local neighbourhood aggregation across features to increase receptive field and robustness. PatchCore further implements what they call a 'greedy coreset' approach to reduce the memory-bank size while retaining effective sampling to allow for faster inference speed. The above mentioned methods all have the issue of features which are biased towards large datasets without adaption. The method Coupled-hypersphere-based Feature Adaptation (CFA) [58] solves this induced bias by performing transfer learning on the target dataset to dilute features from a pre-trained model.

The PatchCore approach obtains such a high state-of-the-art value that it practically solves the MVTEC [9] dataset. To this end, we must propose new anomaly detection tasks which pose significant difficulty and variability that is prevalent in the real-world. The MVTEC dataset, although challenging is visually sterile in terms of camera position, object location and lighting which is all kept as fixed as possible. This is seldom true in the real-world and a new task reflecting this and taking this into account would step towards generalisability and improved incorporation of such methods in everyday life.

2.3 Classification-Based Approaches

Although classification-based approaches implementing a binary classification-based approach [3,37] have gained superior results in the task of visual anomaly detection, certain tasks do not always have the luxury of containing an abundance of abnormal samples during training. This is primarily why approaches implementing a one-class classification paradigm in which classifiers are trained solely across the nominal class were introduced [39, 55].

Initially, One-Class Support Vector Machines (OC-SVM) [59, 60] and Replicator Neural Networks [61] were proposed for this task. For SVM-based approaches, a hyperplane is located in feature space which maximally separates the data from the origin. The application of Support Vector Machines (SVM) for one-class classification was first proposed in [60]. This approach utilises a kernel expansion function

to approximate the hyper-plane to evaluate whether a given sample lays within the learned distribution of normality, or is novel enough to be categorised as anomalous. The Support Vector Data Description (SVDD) [62] extends this work by using a hyper-sphere which offers more flexible descriptions than a conventional hyper-plane [62, 63].

As well as SVM-based approaches, Replicator Neural Networks (RNN) [61] have also been introduced to one-class classification-based anomaly detection [61, 64, 65]. RNN which are neural networks similar to autoencoders whereby the function of the RNN is optimised to replicate the input data pattern at the output layer identically to conventional autoencoders. The main difference is that RNN incorporate a staircase-like activation function in the latent layers which quantises the vector of the given layer outputs to a pre-determined number of clusters [66]. The method outlined in [64] propose a method for visual one-class anomaly detection using RNN. However, their approach seems to be identical to reconstruction-based autoencoder methods featured later in this chapter (Section 2.4).

In-addition to SVM and RNN-based approaches, recently, work has implemented Extreme Learning Machines (ELM) [67] to the task of one-class visual anomaly detection [65, 68, 69]. The ELM method utilises a single-hidden-layer feed-forward neural network (SLFN) [68]. However, unlike conventional SLFN, where all weights in the network are updated via back-propagation, ELM models only update the output weights during training [65], allowing for faster learning speed and good generalisation performance [70–72]. The One-Class ELM (OC-ELM) approach proposed in [73] demonstrates the advantage of using ELM over many conventional one-class classifiers. [69]. The single layer ELM architecture has since been extended to a multi-layer framework via a number of approaches including stacked autoencoders [69, 74], sparse representation-based hierarchy [75], deep weights [76], and multi-layer kernel [77]. More recently, the work by Hashmi *et al.* [65] propose speeding up the training of Replicator Neural Networks (RNN), previously mentioned in this section, by using an ELM which significantly outperforms K-nearest neighbours and SVM-based approaches.

It is well-established that SVM-based approaches struggle with more complicated tasks requiring non-linear representations [78–80]. Deep learning methods such as neural networks have proven significantly better at modelling such high-dimensional data and improved the performance of anomaly detection approaches [80, 81]. Although deep learning methods can successfully model higher-dimensional data, they struggle with singular class classification tasks [82]. However, methods have been introduced to counter this issue [80, 83, 84]. Deep methods can be split into two paradigms for one-class anomaly detection: Generative (autoencoders, GAN) [83,85] and Discriminative models [80]. The former, explained further in this chapter in Section 2.4 rely on either using reconstruction error as a direct score of abnormality, or utilising the discriminator module (in GAN-based approaches) as the novelty detector [83]. The later paradigm, discriminative methods, implement an aforementioned 'hybrid' approach which utilises a method similar to the one outlined in Section 2.2 whereby the high-dimensional data is encoded to feature space where it is then processed by a second stage of classification such as SVM. The work by Ruff *et al.* propose the hybrid approach Deep-SVDD [86] of applying deep neural networks to a hyper-sphere of minimum volume. This work is further used in the work of [64] where SVDD is used as a post-training step to fine-tune the model. The authors show in their results, that this improves performance of the model for the task of anomaly detection. The work by Chalpathy *et al.* [80] combines the hybrid approach with autoencoder methods by implementing a classification head with a neural network calling this approach the One-Class Neural Network (OC-NN). Building off this, the One-Class Convolutional Neural Network (OC-CNN) [82] which utilises a CNN feature extractor to embed images to feature space, and then jointly introduces zero-centred Gaussian noise into the latent space by concatenating it with the feature vector to act as a 'pseudo-anomaly class'. A neural network, just as in [80] is then used to categorise the samples as anomalous or normal based on Binary Cross-Entropy. The work by Perera *et al.* [87] differ from this and instead use an external reference dataset to represent the anomalous class. Self-supervised learning can greatly improve the classification-capability of neural networks in one-

class classification tasks by learning features which are more useful for detecting anomalies [88]. Self-supervised learning is achieved through learning via a pretext task. Such pretext tasks could be geometry-based, or style-based (contrastive). RotNet [89] is a geometric-based pretext task where predicting rotations in images is used as a pretext task to learn image representations. This has been proven effective at the task of one-class classification [90]. Another geometry-based task relies on predicting the relative global image position of a given patch of an image [91]. Golan *et al.* [88] propose a geometric approach in which a multi-class model is trained to discriminate between many geometric transformations applied on all images. This in turn learns a better representation in which to categorise between known and unknown classes given only the known class. They demonstrate a huge increase in performance compared with prior state-of-the-art methods. The Classification-based Anomaly Detection for General Data method [39], named GOAD by the authors further improves this method by unifying state-of-the-art methods and works by initially transforming data into sub-spaces to learn a feature space such that inter-class separation is larger than intra-class separation. This enables the distance from the cluster center within features to be directly correlated with the likelihood of the given sample being anomalous.

Within style-based pretext tasks, the work by Zhang *et al.* [92] propose using colourisation as a technique whereby mono (black and white) images are provided to the model which is tasked with colourising them. The state-of-the-art method within one-class classification for anomaly detection [93] argue that naively applying methods such as rotation, or contrastive methods is sub-optimal for detecting defects. As such, Li *et al.* utilise the CutPaste [93] method in which the pretext task takes a random patch from a given image and places it elsewhere in the image which produce spatial irregularity to serve as more realistic pseudo defects. The results of this demonstrate a notable increase in performance compared with the methods [89, 91, 92].

2.4 Reconstruction-Based Approaches

Reconstruction-based methods train solely across normal, non-anomalous data and as such must learn meaningful feature representations to model the manifold of normality [94]. Such learned feature representations successfully reconstruct normal data, but fail to reconstruct anomalous data, hence allowing the reconstruction error to be a continuous measure of deviation from normality [80]. Models can generalise well to general feature representations and as such, leading trained models to be able to extrapolate and to reconstruct anomalous parts in presented samples, despite no exposure to such during training. This essentially equates to convergence to an identity function in which the input is approximately equal to the output. This leads to low reconstruction errors for anomalous inputs which can affect the discrimination between normal and abnormal samples during inference [38, 95, 96]. This can, however, be overcome by regularisation in the form of denoising.

Common reconstruction-based architectures utilise methods such as Autoencoders (Section 2.4.1) and Generative Adversarial Networks (GAN) (Section 2.4.3) as these can well-represent features fitting to the distribution of singular class data.

The work presented in this thesis is primarily centred around reconstruction-based approaches to anomaly detection. Justifications for using this paradigm are such that 1) using only non-anomalous data during training is cheaper to obtain than a wide variety of anomalous examples and 2) such approaches offer explainability as to the magnitude and location of the anomalies present by taking raw-pixel differences between the input and the output reconstruction.

2.4.1 Autoencoder Methods

Autoencoders were introduced in 1987 [97] initially as a Multi-Layer Perceptron (MLP) which consists of two components, the encoder and the decoder trained in series to map an input image to itself with as little distortion as possible [98]. The encoder component maps the input image data to a compressed latent representation z which the decoder then is tasked to map back into an output image. Due to the

compressed nature of z, the architecture is forced to learn a condensed representation of the input in z. A loss function such as Mean Squared Error (MSE) or L1 loss between the input and the output is typically used to train model weights which minimise the distance between the input and output [99].

Convolutional Autoencoders (CAE) [100] replace the dense perceptron layers with sparsely connected convolutional layers together with pooling to take advantage of the local spatial coherence within image input [100]. The introduced stacked convolutional layers can learn more meaningful representations of pixel interactions in a neighbourhood using information in the higher-order features from previous layers in the architecture. They are also more efficient at modelling images due to sparsely connected features rather than the densely connected MLP model.

Due to the unsupervised nature in which CAE learn accurate feature representations across their input, they are well-suited to the task of anomaly detection when trained solely across the normal dataset. Many works have applied CAE architectures to anomaly detection tasks [101–103]. In fully-trained models, the representations of normality are learned such that all normal parts of the input are included in the reconstructed output. The trained network will attempt to encode and decode anomalous regions at test time using representations over the normal data; As such, the network will produce normal outputs on anomalous parts [104]. Due to this, taking the reconstruction error acts as a continuous measure of anomalous deviation [2, 6].

Autoencoders, however, if large enough, tend to overfit to an identity function in which the representations are learned such that the input is close to equal to the output [105, 106]. This is a form of memorisation for the network meaning that each element within the training set maps perfectly to itself and as such, meaningful learned representations are not learned [107]. Due to this being a near-zero solution to the loss function, regularisation must be introduced to deter such learning convergence. This overfitting is not ideal for the task of anomaly detection because, as previously mentioned, autoencoders must 'repair' anomalous regions in a given image back to normal in order for the reconstruction error to be effective at de-

tecting the deviation from normality [32]. To summarise, a model overfitting too close to the identity function will allow anomalous artifacts to be included in the reconstructions, hence the signal of anomalous deviation will be far weaker and the ability of such models to detect anomalies is reduced significantly [108].

To combat this issue, many regularisation techniques have been introduced including L2-norm regularisation [107, 109–111] and denoising approaches including image dropout [112, 113] and other noising approaches [105, 108, 114, 115]. This thesis will primarily focus on regularising through the use of denoising to align with the work of Chapter 4. One of the main contributors to the work of denoising is the Denoising Autoencoder (DAE) [105, 108, 116] which introduces a simple, yet effective [108] solution of regularisation by adding purposeful noise by randomly setting some pixels in the input images to zero and then tasking the autoencoder to reconstruct the non-corrupted input image from the noisy image [107]. We expand more on this topic in the following subsection.

2.4.2 Denoising Autoencoder

Denoising Autoencoders (DAE) are effective regularisers against overfitting to a identity function. Such noise could be dropping out randomly selected pixels from the input image [105], as purposeful noise [117, 118] to the image, or adding noise to the hidden feature vectors in a given architecture during training [110]. Methods utilising image dropout are, as argued by Wager *et al.* [109], theoretically emulating L2-regularization. This has been established as a well-tested approach to prevent unwanted fitting to the identity function as when image pixels are dropped out or corrupted in the input image, the DAE must reconstruct pixel information based on the information of the surrounding pixels in order to be successful at reconstructing the input faithfully [119]; The identity function would allow such dropped out pixels to appear in the output. Recently, the work [107] defines this as partially correct, however; It will not fully prevent DAE from overfitting to identity functions; instead, it is more accurate to say that offering more rigorous noising can prevent overfitting more often [119].

Differing from this notion of predicting the value of the dropped out image pixels and offering a more robust method of image dropout is the Reconstruction by Inpainting for Anomaly Detection (RIAD) [106] method which sets randomly selected patches of a given image to zero and then tasks an autoencoder to reconstruct an output such that it produces original image content in the zeroed out regions and zeros everywhere else. This method is similar to the work of Adey *et al.* [103] which uses a similar training scheme of learning the inverse, or pixel shifts in the noised regions of the corrupted input. Instead of setting select pixel tiles in the image to zero, each tile is instead corrupted using a randomly selected noising approach (salt, salt and pepper, Gaussian blur, Gaussian Noise, rectangle, line, ellipse arc, shading, erosion, or dilation). This method must learn the inverse of the noised image so that when this is added to the noised image, it results in the original image. Kascenas *et al.* [120] utilise a similar scheme of applying randomly generated Gaussian noise to the foreground pixels of the input image. They report that this significantly increases the precision of reconstruction of the DAE module. At the same time, the work by Salehri *et al.*, Adversarially Robust Training of Autoencoders (ARAE) [31] trains an autoencoder to reconstruct crafted adversarial examples which are perceptually similar to the input sample, but the distance in the latent representation is maximised such that the error between the reconstruction and the input image is minimised. This is the first such method that does not use hand-crafted noise to corrupt the input images, instead utilising backpropagation to optimise for the best noise in which to use to satisfy the aforementioned noise selection criteria of their task. Extending this is the One-Class Learned Encoder-Decoder Network (OLED) [32] which further improves on the performance of ARAE by implementing masking with a binary mask based on the activation of a prior autoencoder network named the mask module. The mask module is trained to maximise the reconstruction error which produces optimal obfuscation of the input images. Unlike [103,106], however, OLED [32] and ARAE [31] do not reconstruct the inverse of the anomaly mask, instead opting for the classical full-image reconstruction approach. These methods fit closely with the work in Chapter 4 in which we train a denoising au-

toencoder with a GAN-like noise generator module which is trained to increase the reconstruction error much like in the work of [32], but surpassing it in performance, obtaining the current state-of-the-art in this field of research.

2.4.3 GAN-based methods

Generative Adversarial Networks (GAN) [121] are powerful latent variable models that can be used to learn complex real-world distributions in such a way that it is possible to generate high-fidelity synthetic examples following successful training [122]. GANs consist of two co-trained modules: the generator and the discriminator. The generator module is trained to produce synthetic examples from either input noise [121, 123] or input data [2, 4, 6, 124] and the discriminator module is trained to differentiate between such synthetically generated examples and real data. Training is performed as a mini-max zero-sum game between the two components; The generator must produce examples which reduces the certainty of the prediction of the discriminator, and the discriminator must find discriminative parts within the synthetically generated examples which do not adhere to the domain of the real data. The convergence of this adversarial training objective is theoretically achieved when equilibrium between the generator and discriminator is reached [125] (Nash Equilibrium) meaning the accuracy of the discriminator to distinguish between normal and anomalous samples is 0.5.

The Deep Convolutional GAN (DCGAN) [125] is a method which is more stable than the original GAN architecture [121] proposed prior which can suffer from mode-collapse, catastrophic forgetting, or non-convergence [126,127]. These improvements include: 1) replacing pooling layers with strided convolutions in the discriminator, and fractional-strided convolutions in the generator module. 2) the use of Batch Normalisation [128] in both modules, 3) the use of ReLU [129] activation for all layers excluding the last layer which uses Tanh in the generator and LeakyRelu activation across all discriminator module layers. These changes allow for stable GAN-based training.

GAN-based approaches [2, 4, 6, 124] used in semi-supervised anomaly detection

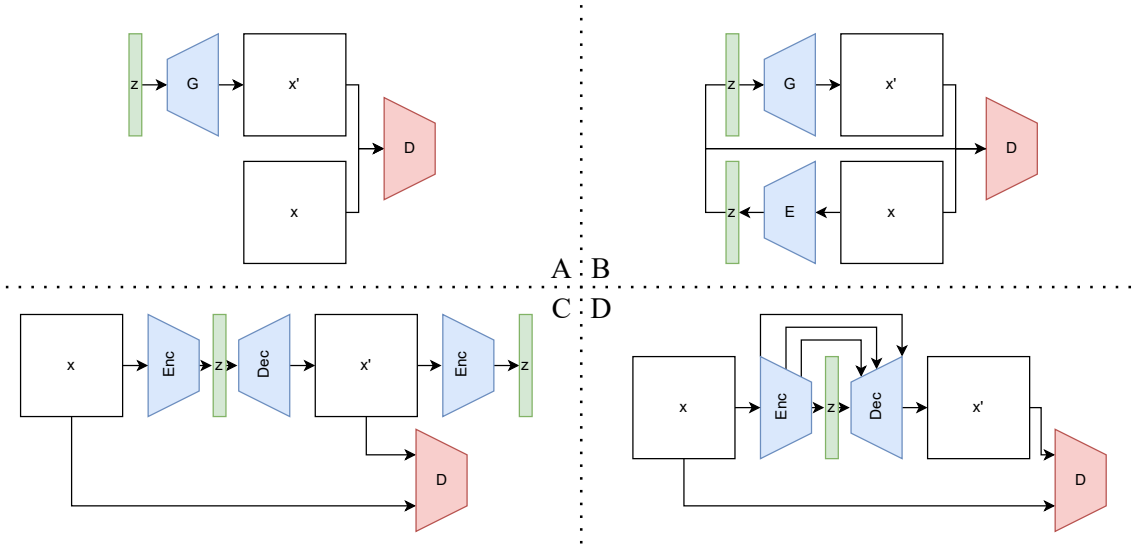

Figure 2.2: Visual comparison of prior methods of GAN-based semi-supervised anomaly detection. A) AnoGAN [4], B) EGBAD [5], C) GANomaly [2], D) Skip-GANomaly [6]

are trained solely across normal, non-anomalous data in a problem. The rarity of anomalies presents an issue of class imbalance as well as little to non-coverage of less-common anomalous instances in a given task [2] which is why training over normal samples is desirable in anomaly detection. As anomaly detection is the task of recognising deviations from normality, then obtained knowledge about the visual appearance of normal examples is relevant to distinguish anomalies in a given task [130]. This following section outlines some patterns observed in the field of visual anomaly detection using GAN and what they implement to improve their anomaly detection capabilities.

The first such work applying an adversarial generative approach to anomaly detection is AnoGAN [4] shown visually in Figure 2.2A. This method trains a GAN in the traditional way by decoding an input noise vector z to a synthetically generated image $G(z)$ which visually matches closely with the distribution domain of the dataset x. The discriminator enforces that the generator produces samples $G(z)$ that are indistinguishable from X. Following convergence of training, the mapping from $z \rightarrow X$ is learned. In order to assign an anomaly score to new samples $\hat{x}$

presented to AnoGAN, the latent space must be searched in order to find the value in the latent space $\hat{z}$ such that $G(\hat{z}) \approx \hat{x}$. In order to find this optimal value of $\hat{z}$, a computationally demanding and time consuming process is required. AnoGAN is inherently slow, but subsequently showed that it is possible to conduct anomaly detection using GAN-based approaches. At a similar time, the Training Adversarial Discriminators (TAD) [131] was introduced to detect abnormal events in crowds. This method uses an image-to-image Conditional GAN (CGAN) [132] which takes input images and random noise as input to generate synthetic reconstructions which should be sufficient to reduce the classification ability of the discriminator module. Soon thereafter, the same authors released a followup paper in which the same architecture from [85] is trained with optical flow between the current frame and the next frame in the video computed using the method presented in [133].

Following from AnoGAN, the method Efficient GAN-Based Anomaly Detection (EGBAD) [5], illustrated in Figure 2.2B, overcomes the issue of compute and time efficiency present in AnoGAN [4] by learning the mapping from x to z while simultaneously training the generator and discriminator modules. This enables the avoidance of the computationally expensive step of finding the latent representation $\hat{z}$ for a new sample $\hat{x}$ at test time. The authors implement a BiGAN [134] architecture for this task which utilises a conventional GAN in the same way as AnoGAN [4] to learn the mapping from z to x as well as an additional encoder (E) which is used to simultaneously learn the mapping from x to z. The pairs $(G(z), z)$ and $(E(x), x)$ are fed into the discriminator module so that the generator can learn accurate mappings from z to x as well as the encoder module learning to map x to z with high aptitude. The results of this work show improved efficiency as well as improved anomaly detection performance [135]. Later, Zenati *et al.* present an improvement to the EGBAD architecture, the Adversarially Learned Anomaly Detection (ALAD) [136] architecture. ALAD is a bi-directional GAN based on the theory of ALICE [137] which incorporates reconstruction errors as a measure of abnormality based on adversarially learned features obtained during training of the bi-directional GAN [134].

Pidhorskyi *et al.* introduce the Generative Probabilistic Novelty Detection (GPND) [138] which also leverage a GAN which utilises reconstruction error as well as one-class classification but use two discriminators to take a more probabilistic approach to anomaly detection. The first discriminator distinguishes between real and synthetically generated images while the second takes the latent representation produced by the encoding of the input sample along with the distribution prior (a normal distribution with 0 mean and standard deviation of 1) as input. At a similar time, Haloui *et al.* [139] introduce the method of Wasserstein GAN [140] (WGAN) which is an approach to encourage the generator module to better approximate the distribution of the input data by reducing the Kantorovich–Rubinstein (Earth-Movers) distance between the distributions of the input and the generated images. WGAN better solves the vanishing gradient and mode collapse issues with DCGAN [125], but is slower than DCGAN to train [141].

Schlegl *et al.* utilise this Wasserstein distance [140] to speed up their AnoGAN approach to present the F-AnoGAN [124] architecture. A two stage training process is introduced where firstly, the WGAN [140] is trained in the same way as AnoGAN [4] in which noise is decoded with the generator module to image space and then a discriminator is used to train the generator adversarially. The second stage takes the generator and discriminator from the WGAN and includes them as the decoder and discriminator respectively of the autoencoder architecture in the next stage while freezing the weights. An encoder module is introduced and is trained using the reconstruction error and a latent loss which is inspired by the loss function in [2]. At a similar time, the GANomaly [2] method shown visually in Figure 2.2C proposes an adversarial autoencoder architecture built using DCGAN [125]. The architecture produces synthetic images which are used as input into the discriminator module. However, following generation, a further encoder module is used to re-encode the synthetic images back into a second latent representation. This constrains the latent priors to not be entirely reliant upon the input images. This approach is shown to greatly out-perform the work in [124]. A further improvement, Skip-GANomaly [6] (illustrated in Figure 2.2D) is introduced as an extension to the

GANomaly architecture [2], introducing residual (skip) connections [142] between layers in the encoder to reflected layers in the decoder in the generator module in the same way as U-Net [30]. Sparse-GAN [143] extends from this and utilises a similar method to GANomaly whereby the synthetically generated images are re-encoded by a secondary encoder. However, it does not only use this process for training like in [2], but predicts anomalies based on the latent space rather than the image space. Following on from this, the One-Class Latent Regularised Networks (OCLRN) [144] suggests that the training of adversarial autoencoder based methods can be stabilised with the use of a dual autoencoder network in the generator. As such, an initial autoencoder reconstructs the input x to the synthetic reconstruction x'. A secondary autoencoder then reconstructs x' to a secondary reconstruction. A loss is calculated between the latent representation output of the encoders of the first and second autoencoder similar to [2].

Some of these methods [6, 139] suffer from successfully passing through anomalous parts into the reconstructed image during inference. As previously mentioned in this thesis, the anomaly score will be lower if anomalous parts can be successfully reconstructed by the network. The One-class GAN (OCGAN) [34] combats this potential issue by using a denoising autoencoder generator together with two discriminators. The first discriminator enforces the latent representation of a CGAN to only produce examples of the input class and not copy visually similar features of anomalous input into the reconstruction. The Anomaly Detection with Adversarial Dual Autoencoders (ADAE) [36] approach combats this by using an autoencoder generator twinned with an autoencoder discriminator which is trained to fail to reconstruct the inputs if they belong to the generated distribution and succeed otherwise. The output of the discriminator is then used for anomaly scoring against input queries at inference.

Class Activation Maps (CAM) are a technique that offer explainability as to model predictions by showing a saliency map of the most important regions of a given input. They are computed as the score of the output of an activation within a given layer and the respective class of the input. CAM as a means of guiding

networks towards being more attentive to anomalous parts has been experimentally introduced to the field in a couple of recent works. Both utilise Gradient-weighted CAM (Grad-CAM) [145] which is an approach which can localise important class-specific regions of the image. It differs from regular CAM by computing the gradient of the classification score with respect to the convolutional features determined by the network in order to outline which parts of the image are most important for classification.

The Convolutional Adversarial Variational Autoencoder with Guided Attention (CAVGA) [146] uses a variational autoencoder that incorporates computed attention for anomaly localisation. The network uses an 'Attention Expansion Loss' to supervise the attention illustrated by Grad-CAM on the last layer of the encoder module to spatially localise anomalies. A similar approach is performed in Adversarial Discriminative Attention for Robust Anomaly Detection [147] (DARAD). In this work, grad-CAM is used to guide training to localise on anomalous regions. However, the CAM are generated from the gradients of the discriminator module and not the generator. A visualisation of how CAM can be used for guiding anomalous decision can be seen in Figure 2.3, from the work of [3]. It can be seen on the left in column A, that the CAM-based classification approaches can localise to anomalies, offering some explainability, but in column B, the CAM seem to be randomly dispersed around the image. Although classification-based, or CAM-guided methods perform well, it is far better to use a reconstruction-based approach as illustrated in C and D for better model prediction explainability of which parts exactly are anomalous in a presented sample during inference.

The Old is Gold [148] method redefines the task of the discriminator from identifying between input and synthetically generated images to instead distinguishing between good and bad quality reconstructions. It accomplishes this by training the generator module together with a frozen low epoch state (old version) of the generator which generates notably lower quality reconstructions. The discriminator is trained with both of these as the input together with pseudo-anomalous data which is the pixel-wise average of two randomly selected input images.

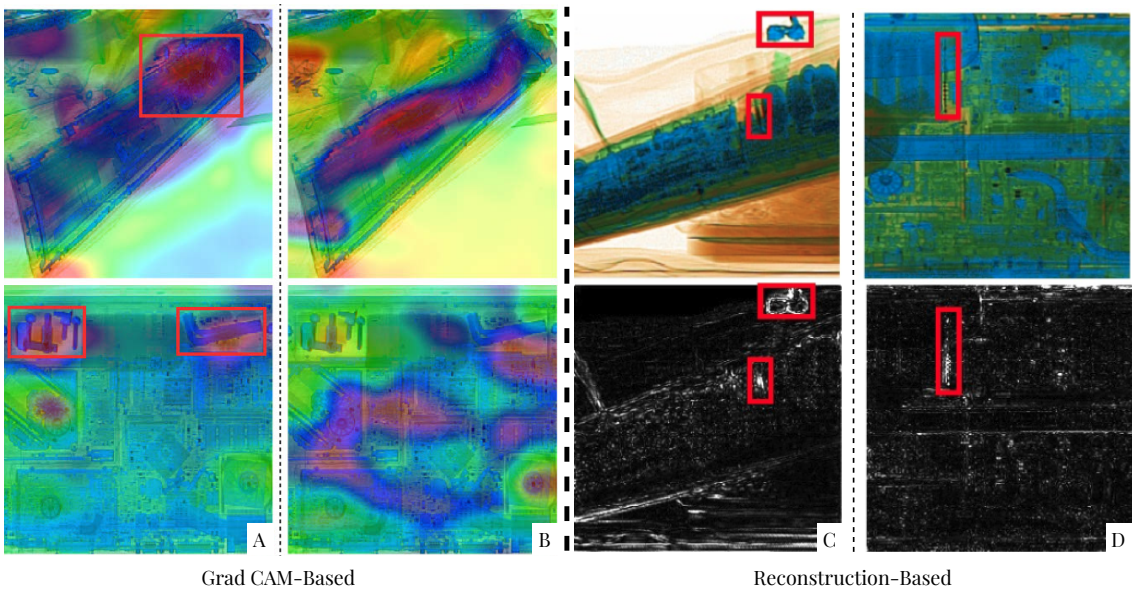

Figure 2.3: Visualisation of anomalous localisation using CAM-based and reconstruction based approaches across the X-Ray Laptop dataset [3]. Images A and B feature GradCAM from the fine-grained classification module in [3] trained with sub-component and object-level components respectively. Images C and D feature anomaly masks produced by the PANDA method in Chapter 3.

Sabokrou *et al.* propose the Adversarially Learned One-Class Classifier (ALOCC) [83] method which tackles the problem with the previously mentioned method of TAD [85] in its inability to reconstruct parts of novel inputs which can subsequently categorise normal images as anomalous. ALOCC also uses a CGAN, like TAD, but trains it as a One-Class Classifier (OCC) in which the network is trained with the target class label of normality in conjunction to the reconstruction error which helps improve the performance of the model across all test images. The Generative-discriminative Feature Representations for Open-set Recognition [149] (GDFR) also trains a one-class classifier using a GAN, however, uses a closed-set classifier on the final stage of categorisation paired with self-supervised training using RotNet [89], which has been explained in the previous section. RotNet is also implemented in the Discriminative-Generative Anomaly Detection (DGAD) [150] while training a GAN to reconstruct input images. The discriminator module not only measures image reconstruction quality, but also outputs a parameter based on image rotation

to encourage the generator to produce a normal angle image.

The issue with the aforementioned methods in this section is that anomaly detectors are task specific [151, 152] and may gain high AUC values between the task specific normal and abnormal class. However, when the normal class is evaluated against vastly out-of-distribution anomalous examples (from a different dataset), the AUC can drop to near random guessing. As such Multiple Class Novelty Detection Under Data Distribution Shift (MCNDDS) [151] is presented to tackle this limitation. Like the previously mentioned method, ALOCC, MCNDDS trains a one-class classifier. However, the GAN architecture uses two decoder networks to reconstruct the source and target input independently. Both decoders sample from the same latent space to enforce domain invariant feature representations. The AnoSeg approach [152] also applies self-supervision to better train a generative method by implementing 'Hard Augmentation' to input image patches with methods such as rotation, perm, color jitter and CutPaste [93]. CutPaste works by replacing a patch of a given input image with a patch from another image to create synthetically anomalous input data. AnoSeg is trained to accurately segment such augmented regions as anomalous thus enabling the generation of accurate anomaly segmentation results at inference.

At a similar time, the Discriminatively Trained Reconstruction Embedding (DRAEM) [153] approach was introduced which also implements self-supervised training based on hard augmentation to create synthetic anomaly masks applied to the input data. Such synthetic anomaly masks are produced by a thresholded Perlin noise [154] similar to [103], which is then multiplied by an anomaly source image (sampled from an out-of-distribution dataset) to produce the anomaly mask. DRAEM is then challenged with reconstructing the clean input images from the noised ones through an autoencoder. A subsequent autoencoder then reconstructs an anomaly mask which is trained to match the thresholded binary Perlin noise mask. Puzzle-AE [155] is a GAN-based method using a U-Net [30] generator in which patches of input are shuffled and the generator is optimised to produce a non-shuffled version of the input. Recently, the Self-supervised Predictive Convolutional Attentive

Block (SSPCAB) [38] which constructs a self-supervised block which incorporates reconstruction-based functionality and masking which, when applied to DRAEM, gains a substantial performance increase.

2.5 Anomaly Detection Datasets

The many datasets used in the process of evaluating anomaly detection methods vary significantly both in difficulty (location, size and frequency of anomalous instances) and from being cross-spectral (Visual, X-ray). Difficulty ranges from visually trivial problems such as classic leave-one-out anomaly detection tasks across MNIST [7] and CIFAR-10 [8] which are akin to vastly out of distribution examples which are unrealistic in real-world tasks. More challenging datasets are real-world anomaly detection tasks which feature both out-of-distribution examples as well as more visually subtle anomalies which lay on the boundary of normal and anomalous. Such examples include factory line inspection [9], agricultural plant leaf disease detection [1], X-ray aviation security scanning [2, 6] and closed circuit public space monitoring [10].

2.5.1 Leave-one-out tasks

Leave-one out anomaly detection is the process of training across a dataset, typically MNIST [7] or CIFAR-10 [8] in such a way that one or many select classes are omitted from the training set and all instances of these classes are assigned as anomalous during inference time [5]. The task is to distinguish between the classes included in the training set and the anomalous classes which has been left out.

The ultimate challenge of anomaly detection models is effectiveness and efficiency during deployment to real-world tasks where the accurate and fast detection of anomalies is crucial [156–158]. By contrast, the *modus operandi* of anomaly detection evaluation in the literature is to solely demonstrate model performance across trivial and unrealistic 'leave one out' tasks on general datasets such as MNIST [7] or CIFAR-10 [8] in which one class from the dataset is labelled as anomalous and all other

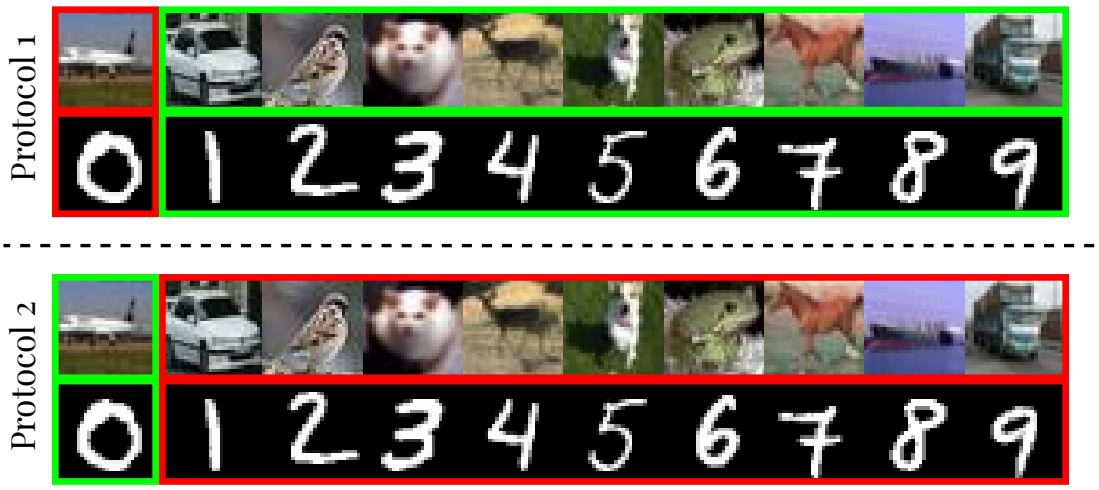

Figure 2.4: Visualisation of the two protocols we used for the MNIST [7] and CIFAR-10 [8]. Above is protocol 1 whereby 9 classes are selected as normal training data and one class is left out as anomalous. Protocol 2 below shows protocol 2 where 1 class is selected as the normal training data and the remaining 9 classes are anomalous.

classes as normal. This evaluation methodology is highly unrealistic as not only are these datasets not intended for anomaly detection, but the act of directly comparing between classes present in the datasets in this way is unlikely to occur in real-world anomaly detection tasks. The relative simplicity of such tasks impose ambiguity to real-world applicability of methods proven effective solely over leave-one-out tasks. Such tasks evaluate the capability of the model to detect vastly out-of-distribution examples which as previously stated are seldom present in real-world tasks. On the other hand, anomalies occurring within real-world problems can be subtle, localised to a small sub-region of the image, exhibit high variance or even be the result of subterfuge by an adversary [1,3,9,10] thus are significantly more challenging. Classic leave-one-out anomaly detection tasks do offer some advantages, however, namely due to the aforementioned rarity of vastly out-of-distribution examples in real-world tasks as well as the simplicity and lightweightedness of the datasets, such tasks can be used as 'toy' tasks which can be used to quickly test performance during development, requiring significantly less compute. Their use as a sole evaluation criteria should be strongly discouraged, however, and a diverse set of real-world

datasets should be used to truly evaluate a methods capability.

Prior methods [2, 5, 6] implement a scheme in which one class is omitted from the training set and the models are trained across all nine remaining classes. A 80:20 split is conducted between training and testing respectively. Methods [31, 32], implement the inverse of this paradigm. They instead opt to omit nine classes out of the training set and instead train solely across one select class. The performance of the latter training paradigm is arguably significantly easier than the prior due to models only having to learn representations over one class. This is reflected in the accuracies of methods across each respective paradigm with the latter having higher average AUC scores.

2.5.2 MVTEC AD

The MVTEC Anomaly Detection dataset is a visual benchmark task with a focus on recognising and detecting faults in factory line products. Detecting such faults is important not only for quality control of the products being produced, but also for the safety of the consumer. Certain anomalous instances such as contaminants within beverage or food products (bottle and hazelnut), or errors such as incorrect pill type, or misprint of text on a pill could be detrimental to the health of the consumer. Detecting such anomalies with high accuracy has been a recent major area of focus for anomaly detection methods. The dataset itself contains 15 classes of objects, each containing both instances of non-defective (normal) samples as well as a selection of common defects which present frequently for a given class. The details of this dataset are outlined in Table 2.5.2 which gives the amount of data samples that each class contains, together with the image resolution of samples in each class as well as an outline of the defects included in each class of the dataset.

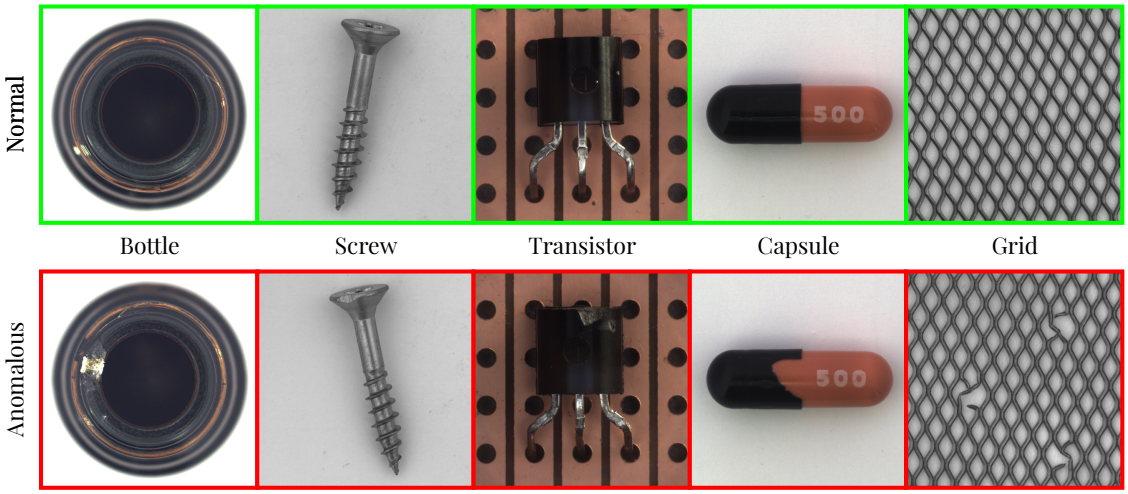

Figure 2.5: Example samples from the MVTEC Anomaly Detection dataset [9] featuring 5 out of 15 classes (Bottle, Screw, Transistor, Capsule, Grid).

	Class	Train Images	Test Images (normal)	Test Images (defective)	Image Resolution (squared)	Contained Defects
Textures	**Carpet**	280	28	89	1024	Colour, Cut, Hole, Metal Contamination, Thread
	Grid	264	21	57	1024	Bent, Broken, Glue, Metal Contamination, Thread
	Leather	245	32	92	1024	Colour, Cut, Fold, Glue, Poke
	Tile	230	33	84	840	Crack, Glue Strip, Grey Stroke, Oil, Roughness
	Wood	247	19	60	1024	Colour, Hole, Liquid, Scratch
Objects	**Bottle**	209	20	63	900	Broken (large), Broken (small), Contamination
	Cable	224	58	92	1024	Bent Wire, Cable Swap, Cut Inner Insulation, Cut Outer Insulation, Missing Cable, Missing Wire
	Capsule	219	23	109	1000	Crack, Faulty Imprint, Poke, Scratch, Squeeze
	Hazelnut	391	40	70	1024	Crack, Cut, Hole, Print Error
	Metal Nut	220	22	93	700	Bent, Colour, Flip, Scratch
	Pill	267	26	141	800	Colour, Contamination, Crack, Faulty Imprint, Pill Type, Scratch
	Screw	320	41	119	1024	Manipulated Front, Scratch (head), Scratch (neck), Thread (side), Thread (top)
	Toothbrush	60	12	30	1024	Contamination, Frayed Bristles, Missing Bristles
	Transistor	213	60	40	1024	Bent Leg, Cut Leg, Damaged Case, Misplaced
	Zipper	240	32	119	1024	Broken Teeth. Fabric (border/interior), Rough, Split Teeth, Squeezed Teeth

Table 2.1: Details of the MVTEC AD dataset taken from [9] outlining per-class amount of training and test images, image resolution, and textual description of defects.

2.5.3 UCSDPed

The University of California, San Diego Pedestrian (UCSDPed) dataset [10] addresses the task of detecting anomalous actions and objects occurring on crowded public pedestrian walkways using a single channel common closed circuit television camera. Anomalies featured in this dataset include pedestrians riding on bikes, pedestrians riding on skateboards, pedestrians venturing off the designated footpath, or small vehicles driving on the pedestrian walkway.

The data itself is split into two subsets corresponding to two separate scenes. UCSDPed1 contains 34 training samples and 36 testing samples taken from a camera which is parallel to the direction of travel of the pedestrian walkway. As such it captures people walking towards and away from the camera, adding perspective distortion to the problem. UCSDPed2, the second subset places the camera perpendicular to the direction of travel of pedestrians on the walkway. This contains 16 training samples and 12 testing samples.

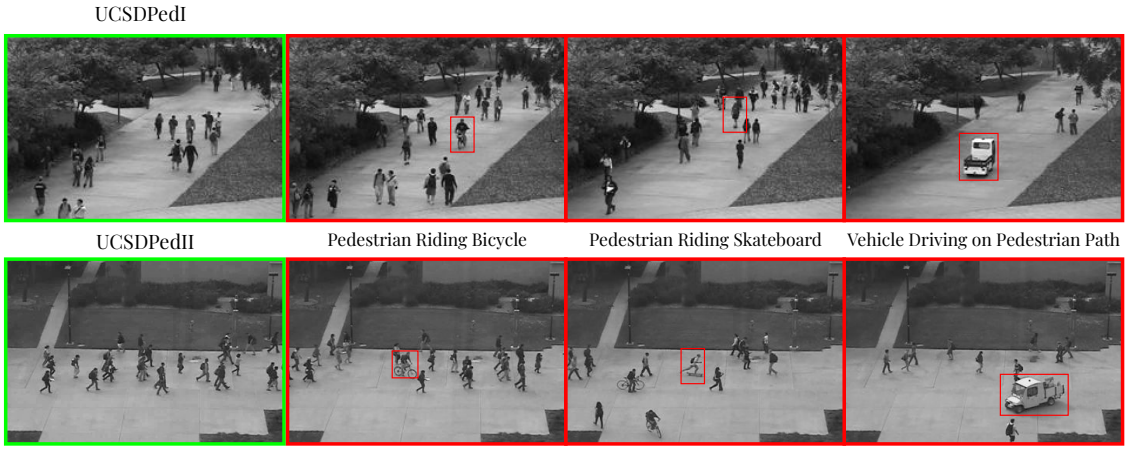

Figure 2.6: Example images from the UCSDPed dataset [10] featuring non-anomalous example (left) and anomalous examples thereafter to the right, namely: Pedestrian Riding Bicycle, Pedestrian Riding Skateboard, Vehicle Driving on Pedestrian Footpath; For both UCSDPed1 (top) and UCSDPed2 (bottom).

2.5.4 Plant Leaf Disease

The UN Department of Economic and Social Affairs (DESA) predicts that the human population size will increase to over 9.7 billion in 2050 [159]. To sustain this population, the World Resources Institute estimates that food production must increase by an estimated 56% [160]. A key contributing factor for potential yield losses up to 16% globally is caused by plant pathogens [161]. The early detection and removal of diseased plants will help to minimise disease spreading to other plants in close proximity and hence maximise the potential yield loss [1].

The Plant Village dataset [1] (outlined in Table 2.5.4) focuses on detecting visual diseases in agricultural leaves caused by pathogens. The dataset contains segmented images of leaves across six crops (Cherry, Potato, Corn, Strawberry, Grape, Tomato) each containing normal (healthy) leaves as well as a selection of leaves with visual diseases common to the respective crop. The diseases vary from visually obvious with vivid discolourations and missing leaf parts, to visually subtle diseases which are almost indistinguishable from their healthy counterparts.

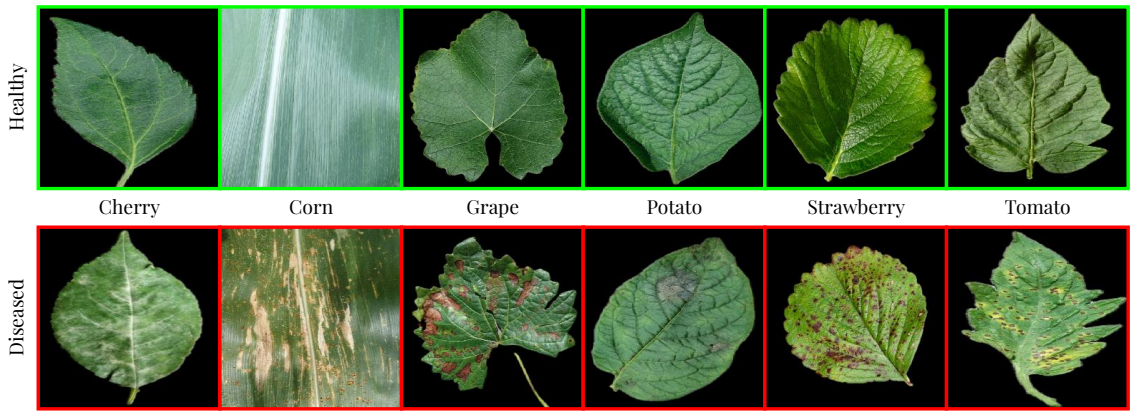

Figure 2.7: Example images from the Plant Village dataset [11] with both healthy and diseased samples from each of the 6 classes: Cherry, Corn, Grape, Potato, Strawberry, Tomato.

Class	Train Images	Test Images (normal)	Test Images (defective)	Leaf Diseases Included
Cherry	684	170	170	Powdery Mildew
Corn	930	232	232	Cercospora Grey Spot, Common Rust, Northern Blight
Grape	338	85	85	Black Rot, Black Measles, Isariopsis Blight
Potato	122	30	30	Early Blight, Late Blight
Strawberry	365	91	91	Leaf Scorch
Tomato	429	318	318	Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Spot, Spider Mites, Target Spot, Yellow Leaf Curl Virus, Mosaic
Total	2868	926	926	

Table 2.2: Overview of the Plant Village [1] dataset outlining the split between the train and test sets. Examples of class-specific diseases are included in the final column.

2.5.5 University (X-Ray) Baggage Anomaly (UBA)

The UK Civil Aviation Authority states that 31.4 million passengers flew between January and March of 2022, a decrease of 42% from the same period in 2019 (before the COVID-19 pandemic) at 44.6 million [162]. Even with such a decrease in passengers during 2022, 606,375 tonnes of cargo was carried in and out of the UK between this period. With so much luggage being moved in and out of the country, it is becoming increasingly difficult to accurately and quickly security screen cargo manually with a human operator. As such, there is a lot of interest in solving this problem of real-time treat item detection in X-Ray baggage data [2, 6] both for increased throughput during peak times in airports while retaining high detection

accuracy of threats.

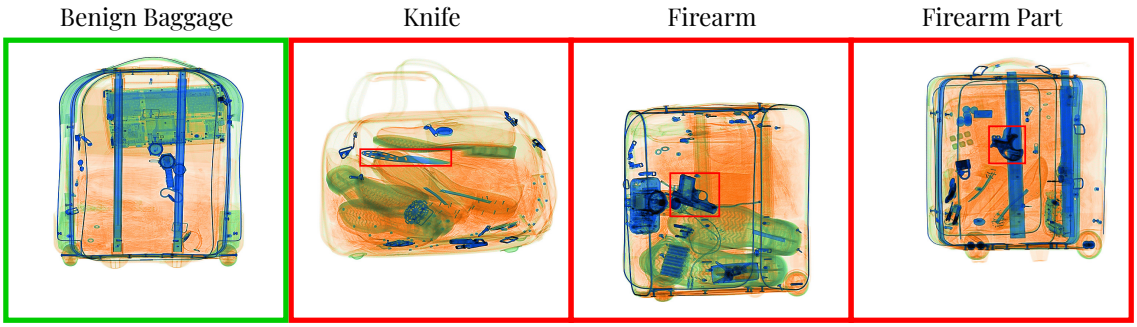

Figure 2.8: Examples from the University Baggage Anomaly (UBA) Dataset [2] outlining one normal sample (left) followed by three samples from anomalous classes: Knife, Firearm, Firearm Part.

The University Baggage Anomaly (UBA) dataset [2] contains 230,275 X-ray images of traveller hand-luggage which is either benign or contains a threat in the form of: Knives, Firearms or Firearm Parts. Imagery is extracted via an overlapping sliding window from a full X-ray image, constructed using single conventional X-ray imagery with associated false color materials mapping from dual-energy [163]. The dataset contains 230,275 benign baggage images and 45,855, 13,452 and 63,496 anomalous images containing Firearms, Firearm Parts and Knives respectively.

Laptop X-Ray

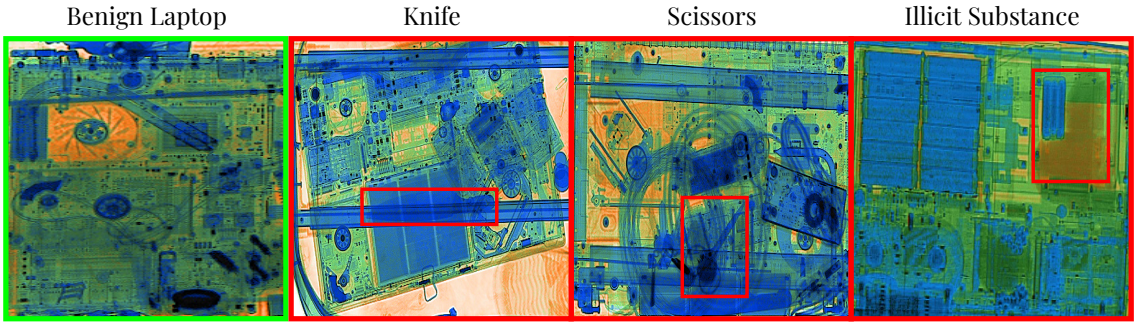

Figure 2.9: Examples of samples from the Laptop X-ray dataset [3] featuring a benign laptop (left) followed by sample images of laptops with concealed threat items: Knife, Scissors, Illicit Substance.

Of particular interest are threat items concealed within large electronic devices such as laptops. Threat items such as illicit drugs, explosives, gun components could be obfuscated by the visual complexity of the X-Ray image produced by scanning such electronic devices. It is for this reason that many airports globally require passengers to remove large electronic items from baggage prior to scanning so that operators can view them without risking obfuscation of malicious parts by other items within the bag. The dataset proposed in the work [3] presents X-Ray scans of normal, benign laptops together with their anomalous counterparts which contain specially engineered inert contraband consisting of plastic explosive, cocaine, and pills of illicit drugs hidden within the electronics of the laptops. Such items are incredibly difficult to detect within the complex structure of the underlying circuitry and interior components of such large electronic items.

2.6 Wind Turbine Inspection Datasets

The task outlined in Chapter 5 focuses on detecting visual surface defects of Glass-Fibre Reinforced Plastic (GFRP) turbine blades. This is an inherently difficult task even for humans to perform [164]. Some real-time blade inspection still requires a human engineer to dangerously abseil down a wind turbine blade to manually inspect the blade for damage [165]. A safer technique has recently been introduced which utilises Unmanned Aerial Vehicles (UAV) or drones for better safety. However, such drones cannot get as close to the blades as human operators. This is trivial given that wind turbines are likely to be in areas with high wind speeds and getting too close to the blade could cause the drone to crash into a blade, causing damage. Even though the onboard cameras of the drone are super high-resolution, it is still harder to spot anomalies in wind turbine blade surfaces from drone imagery [166].

This is especially true when considering the subtle nature in which some visual defects may present on the blade as well as the lack of visual features on some parts of the blade. They may be missed by a human operator sifting through the masses of data manually. As such, the aim of this task is to automate this process of detecting

visual defects on wind turbine blades using models trained to detect them so that they can act as a tool for human engineers to reduce the workload by reducing the amount of inspection images to sift through. This thesis applies a more in-depth explanation of this task in Chapter 5.

This task contains two datasets the Danish Technical University NordTank Wind Turbine Inspection and the Ørsted Offshore Turbine Blade Inspection

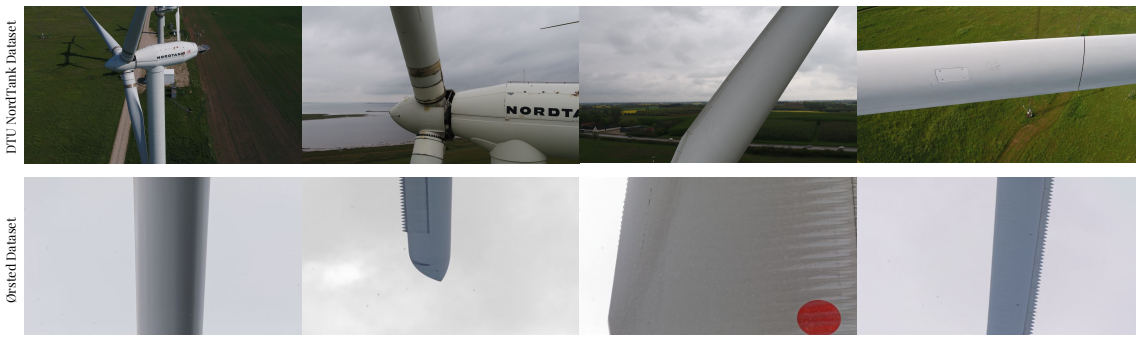

Figure 2.10: Example images taken from the Wind Turbine Blade Inspection Datasets: DTU NordTank [12] (top) and the Ørsted Offshore Wind Turbine Blade [13] (bottom).

2.6.1 Danish Technical University NordTank Wind Turbine Inspection

The Danish Technical University (DTU) NordTank Wind Turbine Inspection dataset [12] contains 1170 images of onshore wind turbines captured at the DTU National Laboratory for Sustainable Energy at Risø, Denmark, from an Unmanned Aerial Vehicle (UAV) mounted camera. The images themselves are of resolution 5280 $\times$ 2970 and were initially devoid of any prior annotation; As such, we manually annotated all turbine blades featured in the images of the dataset in the form of polygon segmentations and bounding box annotations.

We did not use this dataset for detecting anomalies in the blades due to the ambiguity of the defects in this dataset. We did not have expert annotators to give us an indication that a defect was present in the blade, as such, we could not assume that the given blade had a presented defect.

2.6.2 Ørsted Offshore Turbine Blade Inspection

The Ørsted Offshore Turbine Blade Inspection dataset [29], in contrast to the DTU NordTank dataset [12], consists of blade inspection images of offshore wind turbines collected by a UAV mounted camera. The data is collected from the Hornsea 1 offshore wind farm and consists of 2637 non-annotated images of resolution 6720 $\times$ 4480. The images themselves mostly consist of a background of sky which is beneficial as it is mostly featureless compared to the diversity in background of the images within the DTU Nord Tank dataset previously mentioned.

We then computed super pixel segmentations of the blades using the SLIC (Section 5.2.2) approach. These super-pixel segmentations were then categorised into two categories {normal, anomalous} depending on whether they contained any visible defects in them and supported by the rough annotations supplied to us by Ørsted engineers. This process was incredibly labour intensive, but following this, we obtained an anomaly dataset for turbine blade defect detection.

Examples of the defects in this dataset are illustrated in Figure 5.13 in which anomalous regions are present in the super-pixel regions of the turbine blades.

2.6.3 Evaluation Criteria

Calculating Average Precision (AP) of the object detection method requires Precision (P) and Recall (R) as the culmination of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) in the formulas:

$$P = \frac{TP}{TP + FP} \tag{2.1}$$

$$R = \frac{TP}{TP + FN} \tag{2.2}$$

Intersection Over Union (IOU) quantifies how well the predicted bounding box overlaps with the ground truth of the object. This is illustrated in the diagram in Figure 2.11 in which the predicted detection of the turbine blade is measured against

the ground truth. Note that a higher IOU value demonstrates a closer fit to the ground truth. IOU is measured in the following formula:

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2.3}$$

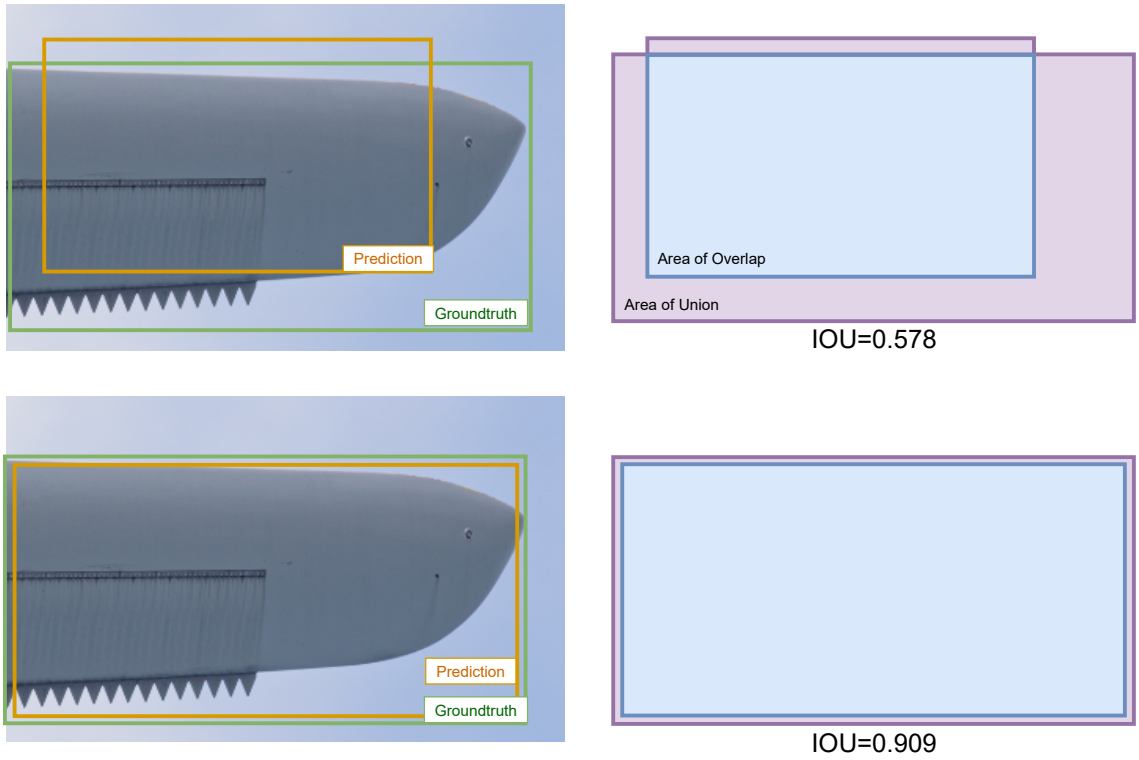

Figure 2.11: Visualisation of how the Intersection Over Union value is calculated for a predicted bounding box by evaluating with respect to the ground truth.

It is common in prior object detection work [18,167–170] to use an IOU threshold of 0.5 as a means of removing those bounding boxes which offer poor detection capability.

A precision-recall graph is constructed from precision (Equation 2.1) and recall (Equation 2.2) as a function of precision with respect to recall. A correct prediction for a given object is correct iff $IOU \geq 0.5$. AP is calculated as the area underneath this obtained curve as:

$$AP = \int_0^1 P(R)dR \tag{2.4}$$

2.7 Anomaly Detection Metrics

Throughout this thesis, we use a number of metrics to evaluate the performance of anomaly detection methods. In this section, we will explain each of them and how to interpret their results.

The Receiver Operating Characteristic or ROC, is a plot of the True Positive Rate (TPR) = $\frac{TP}{TP+FN}$ an the False Positive Rate (FPR) = $\frac{FP}{FP+TN}$ for a binary classification at each threshold setting. The Area Under Curve (AUC) is the area of the space underneath this ROC curve. A higher AUC score indicates better performance as the curve will be pushed higher into the top left corner due to better separability in the two prediction distributions. Conversely, an AUC value of 0.5 indicates that there is no separability of the two distributions and the model is randomly guessing the predictions for new samples.

Within anomaly detection, the Area Under Curve (AUC) of the Receiver Operator Characteristic (ROC) is used due to it being classification threshold-invariant. It is calculated using the precision (Equation 2.1) and recall (Equation 2.2). This produces an ROC curve of False Positive Rate (FPR) with respect to True Positive Rate (TPR). Integrating this line using equation 2.5 yields the Area Under Curve (AUC) value between [0.5, 1) for a given set of predictions. The lower bound of 0.5 implies that the model is randomly guessing the prediction and as such the distributions are near equal. Conversely, the upper bound of ~ 1 implies perfect categorisation of given samples due to absolute distinction between the two distributions. Figure 2.12 illustrates the relationship between distribution overlap and AUC score.

$$AUC = \int_0^1 \text{FPR(TPR)dTPR} \tag{2.5}$$

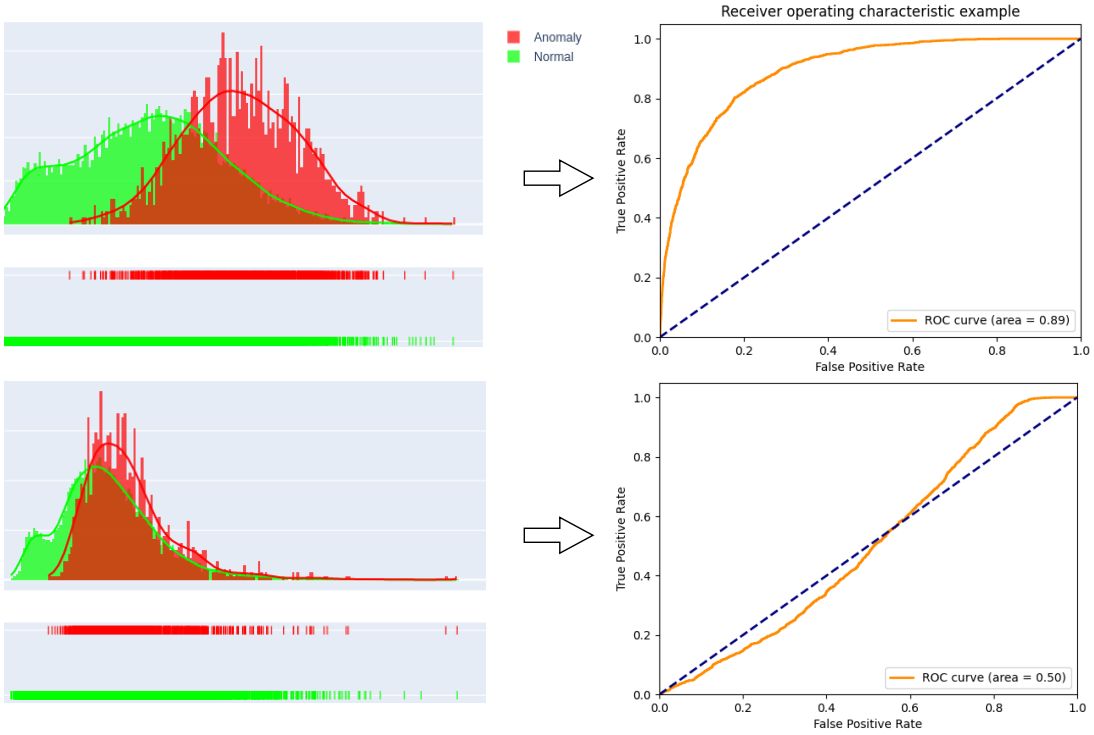

Figure 2.12: The distribution of anomaly score (left) together with their corresponding ROC Curve (right). The above distribution shows a meaningful separation between the distributions and as such obtains an AUC of 0.89 whereas the bottom distribution overlaps massively leading to an AUC of 0.5

2.8 Conclusion

This chapter reviews approaches in visual anomaly detection across three main paradigms in this field namely: probabilistic, classification-based and reconstruction-based approaches; outlining the *modus operandi* of each paradigm and illustrating crossovers which demonstrate that they are not entirely mutually exclusive with how they contribute to solving the task of visual anomaly detection. A strong focus is given to reconstruction-based methods as these are the focus of the anomaly detection approaches presented within Chapters 3, 4 and 5. The justification for choosing reconstruction-based approaches over the other such paradigms is that they not only exhibit strength in predictability during inference, but also that they offer more explainability to their predictions than probabilistic and classification-based

approaches as evidenced in Figure 2.3. As recent methods within reconstruction-based anomaly detection have seen success by including the use of denoising into their training schemes, we evaluate these techniques and build off this to produce the work presented in Chapter 4.

Further to this, we also evaluate all synthetic leave-one-out (MNIST and CIFAR-10) and real-world (MVTEC, Plant Leaf Disease, UCSDPed and X-ray) visual anomaly detection datasets which are used to thoroughly benchmark anomaly detection methods to show their capability to detect anomalies in real-world tasks.

CHAPTER 3

PANDA

3.1 Introduction

Anomaly detection methods have had varying degrees of success across datasets of real-world tasks including, but not limited to: retinal diagnosis [4, 124] where accuracy (effectiveness) is preferred, factory line inspection [9] where the speed of detection is important (efficiency), and airport security scanning [2, 3, 6, 37] where both effectiveness and efficiency must be maximised. However, methods presented to solve these tasks can often attribute their limited success to being domain-specific and are not applied across multiple, diverse (multi-spectral; cross-domain) datasets.

It is unclear how well anomaly detection methods trained in these domains actually perform in the real-world scenarios, but the datasets are set up to give a close indication of how methods would perform 'in-the-wild'.

Figure 3.1: **Top:** Leaves from Plant Village [1] featuring visible diseases. **Bottom:** Anomalous instance segmentation masks generated by PANDA for the respective diseased leaves.

Whilst supervised methods [3,37] by binary classification approaches have proven to obtain superior performance across anomaly detection benchmark tasks, often by following a simplistic anomaly detection by classification paradigm with discrete classes, they require large, labeled-datasets for training. These can be both expensive to obtain, unbalanced in nature, and will always struggle to provide sufficient coverage of rare, low-occurrence anomalies given the potential open-ended scope of the anomalous class space. These challenges of training data adequacy could lead

to misclassification by potential adversarial example attacks against such methods [171, 172].

By contrast, generative semi-supervised methods [2, 6, 124, 173, 174] overcome this issue by learning a close approximation to the true distribution manifold exclusively over the non-anomalous (normal) data samples [4]. Such techniques use generative methods in order to approximate this distribution of normality [2, 6]. Such prior anomaly detection methods are overly focused on more general features to aid in categorising visually obvious anomalies [2, 4] akin to the flawed evaluation methodology of 'leave-one-out tasks', meaning they do not perform overly well with detecting visually subtle anomalies. Methods such as [1, 4, 124] all suffer from slow inference speed as well, which can hinder their real-world applicability in scenarios where high-throughput processing is required. Furthermore, methods such as [2,124] exhibit vastly differing accuracy with each training run over the same dataset due to instability during training leading to a wider confidence interval as demonstrated in our experiments. GANomaly [2] also suffers from high variation in AUC performance during inference across the same dataset using the same weights. We assume this is due to implemented batch normalisation between layers, however, this needs to be explored further. This problem further impedes real-world applicability due to unpredictable detection behaviour at inference.

In this chapter, we introduce PANDA, an Autoencoder Generative Adversarial Network (AE-GAN) based architecture to combat the task of detecting subtle fine-grained anomalies present in real-world anomaly detection applications whilst also retaining time-efficiency at inference. PANDA includes three novel proposals:

- A Fine-Grained Visual Categorisation Discriminator Network (FGVC): to combat the problem of detecting visually subtle, low inter-class variance anomalies present in real-world anomaly detection problems and to provide a harsher critic during training for the GAN generator module.

- A residually connected dual-feature extractor implementation within our generator module that carries lower-level features in given images forward and

combines them residually with higher-level, later features in the architecture.

- A perceptual loss function based on feature error instead of raw pixel-error; Originally used in Style Transfer tasks [175], perceptual loss has not yet been applied to the task of generative anomaly detection.

This work represents the first instance of these techniques being jointly applied to semi-supervised anomaly detection.

3.2 Approach

Our proposed method applies a unique adversarially trained autoencoder architecture to the task of anomaly detection. Our method is visually outlined in Figure 3.2. Our asymmetric generator module (Section 3.2.1) encodes input images to both a low-level (z_{low}) and a high-level (z_{high}) latent representation with the use of encoders ($\{E^0_{low}, E^0_{high}\}$). z_{high} is decoded to z_ϕ before being residually combined through skip-connections with z_{low} before being decoded back to image space. This allows strong consideration of both high-order and low-order features during the decoding process. Secondary encoders E^1_{low} and E^1_{high} are implemented to re-encode the output of both the bottom D^0_{low} and top-level D^0_{high} decoders respectively. This idea is inspired from the GANomaly [2] approach which uses one extra encoder which re-encodes the decoded (reconstructed image) output back to a latent representation and minimises a latent loss between the original encoding and the re-encoding during training. We optimise the secondary encoders in our approach using: $\|(z_{low} + z_\phi) - E^1_{low}(x')\|_2 \simeq \left\|z_{high} - E^1_{high}(z_\phi)\right\|_2 \simeq 0$ which constitutes our latent error term in our overall loss function outlined in Equation 3.1.

While prior methods [4, 5, 124] solely utilise the reconstruction error between the input image x and the reconstruction x', in this work we experiment using a perceptual loss function. This is explained further in Section 3.2.3. Our justification for using perceptual loss is due to its proven success in style transfer [175] and super-resolution [176] tasks where taking perceptual distances between activations

of a given layer are more beneficial than using raw pixel differences. Perceptual loss is able to better reconstruct fine details compared to methods trained with per-pixel loss [175] as such, we theorise that it should be better at approximating fine-grained details of the images in the normal training distribution and apply this during inference.

We train our generator module adversarially with a unique fine-grained discriminator network (Section 3.2.2) which is optimised to assign a true probability that a presented image is normal and not synthetically generated. This fine-grained discriminator module can detect subtle discriminating features between the input images and the synthetic images during training, offering a harsher critic for the generator module and forcing the generator to produce higher-fidelity reconstructions with emphasis on detail. We also use a weighted output of the FGVC discriminator model during inference while performing anomaly scoring (Equation 3.3) due to the ability to recognise key discriminative regions present in normal samples obtained while training.

3.2.1 Generator Network

Our generator autoencoder model is trained adversarially which produces an Adversarial Autoencoder (AAE) model which remains stable during training meaning that high-fidelity reconstructions are obtained to preserve fine-grained details unlike those produced by vanilla autoencoder-based models. AAE differ from traditional GANs because whereas the latter has a random noise vector as the input, an AAE uses images from the training dataset as input. AAE do not exhibit training difficulties such as mode collapse or non-convergence are avoided which occur frequently while training traditional GAN-based architectures [121].

The architecture for our architecture is inspired by the VQ-VAE-2 [177, 178], a generative approach in which a low and high level latent representation is computed and assigned to a discrete code book which is then sampled using an auto regressive approach during inference to produce new images. In their results, they show, perceptually, that this dual latent representation results in higher-fidelity generations

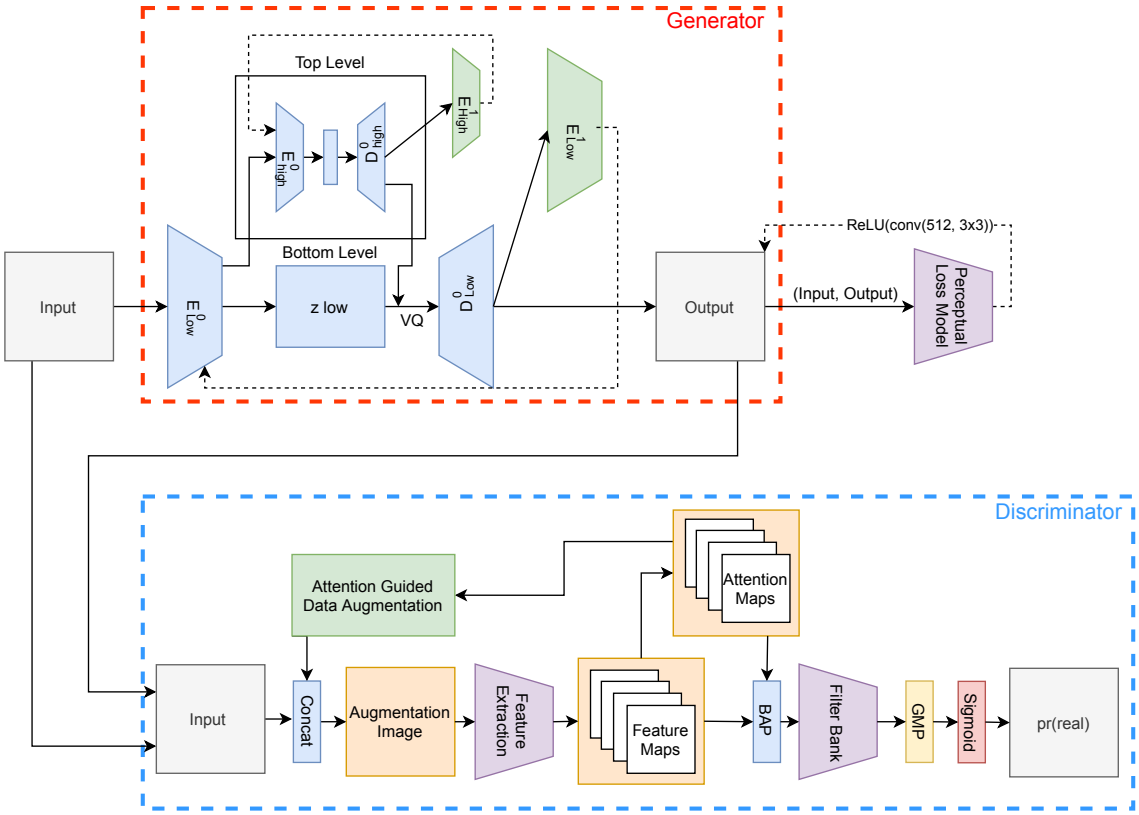

Figure 3.2: Proposed model architecture featuring our PANDA-GAN architecture with the generator network (upper) and the discriminator network (lower) together with the perceptual loss network used for reconstruction.

during inference.

Input images x are fed into an encoder to gather low-level features after being encoded through E^0_{low}. We call this representation z_{low} and it has shape Batch$\times$128$\times$ 64 $\times$ 64 after encoding. The top-level encoder E^0_{high} then further encodes z_{low} to a feature embedding space we call z_{high}. This top-level latent representation has shape Batch$\times$128 $\times$ 32 $\times$ 32 before being decoded by the top-level decoder D^0_{high} to z_ϕ which has the same dimensions as z_{low}. We only implement one higher-order latent representation due to memory constraints and to keep our method more efficient during inference. We show that our method obtains state-of-the-art performance by utilising just one higher-order latent representation. We then residually combine z_ϕ with z_{low} by adding them to preserve the information from both low and high-level features during decoding. The anatomy of our generator module is outlined in figure

3.3.

Our generator module is asymmetric as it contains more encoder layers (convolutional components) than decoder layers (transpose convolutional components) which is implemented in prior work [179–181]. The justification for this model asymmetry is that it increases memory efficiency due to the reduction of parameters in decoding components which also reduces the chance of the model overfitting during training. The rationale behind having increasing the number of encoder layers as opposed to the decoder layers is because it decreases the chance of artifacts in the final reconstruction. [180].

Additionally, we also utilise two secondary encoders, $\{E^1_{high}, E^1_{low}\}$ during training exclusively to re-encode the decoded respective latent representations z_ϕ and x' back into the latent spaces of z'_{high} and z'_{low} respectively. This approach was employed in GANomaly [2] to yield better performance, but on the single latent representation in this architecture. We implement a similar scheme in our architecture on the latent representations in the hope that it yields better performance. Encoders $\{E^1_{high}$ and $E^1_{low}\}$ are solely required during training and so are not enabled during inference time to increase throughput efficiency at deployment time.

Overall our learning objective seeks to minimise over $\forall x \in X$:

$$L_{AE} = L_{rec}(x, x') + L_{\texttt{discriminator}}(x, x') + L_{z[0]} + L_{z[1]} \tag{3.1}$$

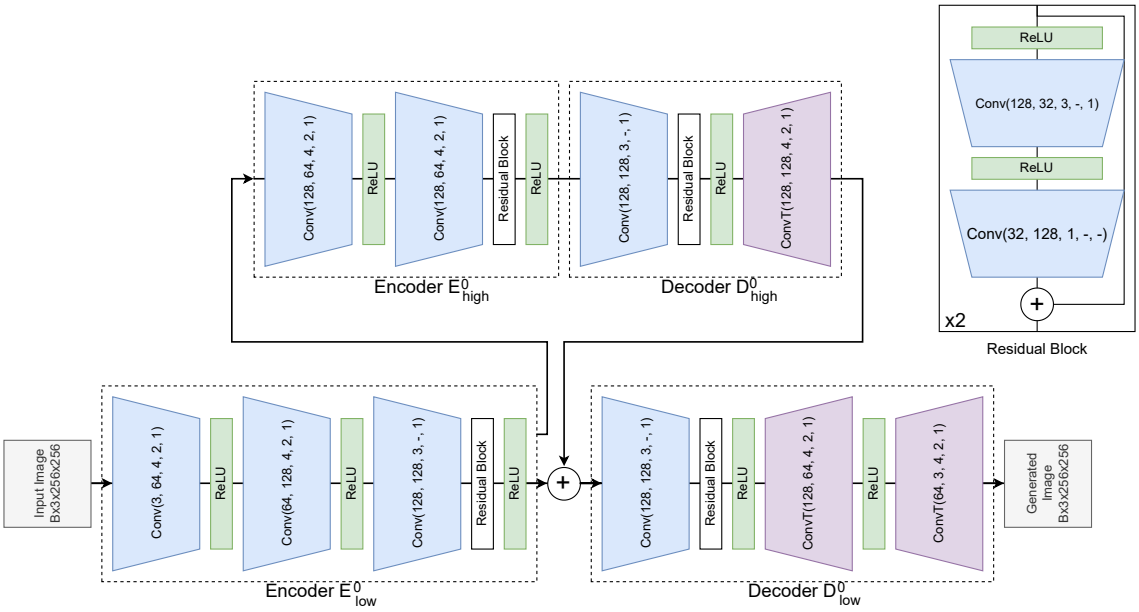

Figure 3.3: In-depth overview of the architecture of the generator module of PANDA.

L_{rec} is either the raw reconstruction error between the pixels of x and x' ($\|x - x'\|_2$), or the perceptual (feature) distance (Section 3.2.3) between the activations of layer 14 in a pretrained VGG19 [182] network. The second component in this loss is the discriminator loss obtained from the discriminator module (Section 3.2.2) This assigns a probability that a given sample x' belongs to the dataset containing x. The final component of this loss is the latent loss $L_{z[i]} = \left\| E^1_{[i]} - E^0_{[i]} \right\|_2, i = \{high, low\}$ between the latent representations z_{high} and $z_{low} + z_\phi$ of the generator, and the secondary encodings produced by E^1_{high} and E^1_{low} respectively.

3.2.2 Discriminator Network

In contrast to prior works in generative anomaly detection which utilise conventional discriminators [2, 4, 6, 124, 183], in this work we incorporate a Fine-grained Visual Categorisation (FGVC) discriminator. In general, FGVC is for use in obtaining specific sub-class classification of objects (e.g. species of bird or model of car) [184]. Typical FGVC datasets are inherently difficult to classify due to highly localised and visually subtle distinguishing features between classes. Within real-world anomaly

detection problems, there exist varying levels in which an anomaly may present ranging from visually obvious to negligibly subtle. Our FGVC discriminator outlined in this work is optimised to detect more subtle anomalies during inference by recognising the discriminating regions within presented images. It also acts as a harsher critic to our generator module during training, promoting emphasis on high-fidelity generation of object parts in detail rather than settling with an approximation of the fine details of objects.

Our discriminator is inspired by the Weakly Supervised Data Augmentation Network (WS-DAN) architecture [185], a proven method in FGVC which obtains superior categorisation performance in the task of FGVC [185].

The WS-DAN architecture contains attention layers which allow the network to focus upon both detailed features and key discriminative object parts during inference when categorising anomalous data. This mechanism also allows attention guided data augmentation within the network leading to higher information gain and optimised augmentation of non-anomalous samples during training. The resulting attention maps are combined with feature representations via Bilinear Attention Pooling (BAP). This combined feature representation is then fed into a discriminative filter bank of 1×1 convolutions followed by a Global Max Pooling (GMP) [186] layer on the resulting feature matrix to reduce dimensionality in the output, and results in a 1×1 patch in the output which is the area of highest discrimination for the discriminator network.

This allows our generator module to refine these areas in the next iteration and thus enable the overall PANDA architecture to reduce the reconstruction error substantially. The final layer of the discriminator module issues a continuous probability score for a presented image through the use of a Sigmoid activation layer. The value of the probability represents the likelihood that a presented sample is an element of the real dataset and not synthetically generated. To prevent vanishing gradients, which is common with logistic functions, we use the residual network, ResNet-50 [142] as our main backbone architecture. ResNets do not suffer from vanishing gradient as the residual connections present in the network allows the

gradient a path to flow to earlier components in the network prior to the logistic function. The pair of real, non-anomalous data examples (x) and the generated, synthetic examples (x') from x are fed into the discriminator to obtain a probability score that each of the images is an element of X.

Overall, the discriminator seeks to optimise:

$$L_C = -log(C(x)) - log(1 - C(x')) \tag{3.2}$$

where C represents the discriminator, or critic network. The pair (x, x') obtains probability outputs $C(x)$ and $C(x')$ respectively. L_c represents $Pr(x \in X|(x, x'))$.

3.2.3 Perceptual Loss Function

We introduce the notion of perceptual loss (PL) to calculate feature error rather than pixel-wise error during reconstruction. Previously introduced to the task of Style Transfer [175], we introduce its usage into the task of Anomaly Detection as a replacement for conventional Pixel-Wise Loss (PWL).

While PWL computes raw-pixel differences between x and x' on low-level and literal pixel value information, PL takes the advantage of taking the error between high-level activation features [175] obtained from pre-trained Convolutional Neural Network (CNN) based classifiers. Feeding the pair $(x_i, x'_i), \forall x_i \in X$ through a pre-trained conventional CNN classifier ($f()$) obtains differing activations $(f(x_i), f(x'_i))$ of a given convolutional feature extraction layer. PL is then calculated as $\|f(x_i), f(x'_i)\|_2$. The PANDA architecture uses a pretrained VGG19 [182] network as the Perceptual Loss model and uses the error between the activations of the 14^{th} layer. We utilise two variants on perceptual loss:

- General Perceptual Loss ($\text{PL}_g()$) : Weights obtained by pre-training a CNN across ImageNet [187].
- Problem-specific Perceptual Loss ($\text{PL}_{ps}()$) : Weights obtained from pre-training a CNN over non-anomalous samples from the specific anomaly detection task dataset.

Justification for using a problem specific loss network is that rather than using general features from ImageNet, we can learn a set of bespoke features unique to our problem set. Across image queries from visually unique datasets such as those in the X-Ray Security Electronics anomaly detection task [3] featured in this work, the queries possess little perceptual similarity to the images featured in the ImageNet dataset. As such, a perceptual loss model trained across ImageNet may only be useful to compare shallow features such as edges whereas deeper features such as textures, patterns or object information will cause a weak and faint perceptual activation signal due to null-exposure of such image features during training. Fine-tuning the perceptual loss model prior to training PANDA across only the normal (non-anomalous) images of a given task causes deeper higher-order features to be learned so that the activation signal becomes stronger. This gives a stronger loss signal to backpropagate during training of the generator module.

Fine-tuning comes with a small added computational overhead, but as our experimental results demonstrate, it isn't always necessary to fine-tune the perceptual loss function if the dataset shares visual similarity to ImageNet. Fine-tuning the perceptual loss function doesn't require much computation; In our experiments we obtained convergence of the loss function after as few as 5 epochs using the hyper-parameters outlined in Section 3.3.1.

3.2.4 Anomaly Scoring

Anomaly scoring is the process of categorising samples as anomalous or non-anomalous via a continuous score of deviation from normality by the gained approximation to the manifold over X (normal samples) based on the knowledge and learned representations of normality that the network has obtained during training.

Anomalous samples will be reconstructed by the generator model from the normal latent representation producing normal (e.g repaired or different class) appearing sample outputs. This allows us to infer a distribution of anomaly scores $N^i_{\texttt{anomaly score}} \sim N(\mu_i, \sigma_i^2)$ where $i = \{normal, anomalous\}$ over both normal and anomalous samples respectively. $N^i_{\texttt{anomaly score}}$ is formed via a weighted sum of the

distributions of the two discriminator scores across both the input samples $N^i_{C(x)}$ and the synthetically generated samples $N^i_{C(x')}$ together with the distribution of reconstruction error $N^i_{L_{rec}(x,x')}$ using the following formula:

$$N^i_{\texttt{anomaly score}} = \beta_0 \cdot N^i_{L_{rec}(x,x')} + \beta_1 \cdot N^i_{C(x)} + \beta_2 \cdot N^i_{C(x')} \tag{3.3}$$

where $\{\beta_0, \beta_1, \beta_2\}$ are real-valued weighting terms for each error component. This allows us to impose varying categorisation power to individual scores during the final anomaly scoring process. The final anomaly score values $a^i \in A^i | A^i = N^i_{\texttt{anomaly score}}$ are negligibly small for both normal and anomalous samples ($1e^{-7} < a^i < 1e^{-6}$) which can make the two distributions difficult to separate; to rectify this, we normalise $\forall a^i \in A^i$ to values $0 \ll a^i < 1$ via Equation 3.4.

$$A^i_{normalised} = \forall a^i \in A^i, \frac{a^i - A^i_{min}}{A^i_{max} - A^i_{min}} \tag{3.4}$$

Once we obtain the distributions $A^i_{normalised}$, an anomaly score can be assigned to any sample presented to the model. Figure 3.7 outlines the distribution of anomaly scores across classes of the Plant Village dataset [1]. Note that generally the model will reconstruct normal samples with more precision than anomalous samples and will assign them a lower anomaly score. As the samples deviate more from normality, the model fails more to reconstruct them and the assigned anomaly score increases accordingly. Both the normal and anomalous distributions approximate to normal distributions with separate means and standard deviations. On average, the distributions of anomaly scores of anomalous samples exhibit a larger variance and mean value than their normal anomaly score distribution counterparts which are more tightly bound with a lower mean value. The boundary between $A^0_{normalised}$ and $A^1_{normalised}$ across the validation or test set can be calculated as a trivial solution to the quadratic equation $a(x) = ax^2 + bx + c$ where a, b, c are defined in formula 3.5 using the mean and standard deviation pairs (μ_1, σ_1) and (μ_2, σ_2) obtained from the distribution over normal ($A^0_{normalised} \sim N(\mu_1, \sigma_1)$) and the distribution over anomalous ($A^1_{normalised} \sim N(\mu_2, \sigma_2)$ data respectively. The solution from this formula can

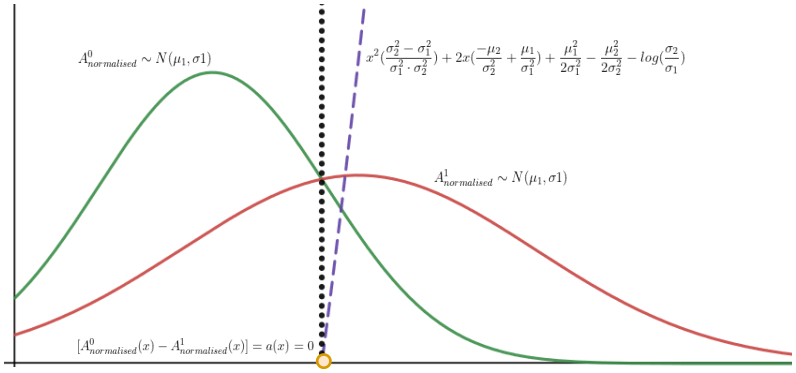

Figure 3.4: Visualisation of anomaly decision boundary ($[A^0_{normalised} - A^1_{normalised}] = a(x) = 0$) between $A^0_{normalised}$ and $A^1_{normalised}$ using Equation 3.5

be used to categorise subsequent presented samples as anomalous or otherwise normal based on the principle of maximum likelihood.

$$a = \frac{\sigma_2^2 - \sigma_1^2}{\sigma_1^2 \cdot \sigma_2^2}, b = \frac{\sigma_1^2 \cdot \mu_2 - \sigma_2^2 \cdot \mu_1}{\sigma_2^2 \cdot \sigma_1^2}, c = \frac{\sigma_2^2 \cdot \mu_1^2 - \sigma_1^2 \cdot \mu_2^2}{\sigma_1^2 \cdot \sigma_2^2} - log(\frac{\sigma_2}{\sigma_1}) \tag{3.5}$$

3.3 Evaluation

We exhaustively evaluate our PANDA approach against prior works [2, 4–6, 14, 21, 85, 124, 188–192] across a series of challenging tasks [1, 3, 7–10] both quantitatively and qualitatively.

3.3.1 Experimental Setup

The experimental setup comprises of the following dataset configurations:

We use the dataset split for {train : validate : test} as following: {13,593 : 2,589 : 12,661} for Plant Village [1], {229 : 25 : 125} for X-ray Security Electronics [3]. The {train : test} split for MNIST and CIFAR-10 is {80% : 20%} across both datasets as was performed in prior work [2, 5, 6]. We utilise the default train/test split for the MVTEC task [9]. The hyper-parameters and data configurations are fine-tuned by systematic grid search in order to obtain the best results across the

Table 3.1: AUPRC results across trivial MNIST [7] and CIFAR-10 [8] leave-one-out tasks.

	MNIST Class										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Avg**
AnoGAN [4]	0.61	0.3	0.54	0.44	0.43	0.42	0.48	0.36	0.4	0.34	0.43
EGBAD [5]	0.78	0.29	0.67	0.52	0.45	0.43	0.57	0.4	0.55	0.35	0.5
GANomaly [2]	0.89	0.65	0.93	0.8	0.82	0.85	0.84	0.69	0.87	0.55	0.79
IGMM-GAN [194]	**0.96**	0.9	0.93	0.82	0.83	0.9	**0.93**	0.9	0.78	0.57	0.85
ADAE [174]	0.95	0.82	**0.95**	**0.89**	0.83	**0.91**	0.89	0.80	**0.93**	0.63	0.86
PANDA	0.83	**0.99**	0.88	0.86	**0.93**	0.89	0.9	**0.91**	0.85	**0.92**	**0.9**
	CIFAR-10 Class										
	Airplane	**Automobile**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**	**Avg**
AnoGAN [4]	0.51	0.49	0.41	0.4	0.34	0.39	0.34	0.41	0.56	0.51	0.44
GANomaly [2]	0.63	0.63	0.51	0.59	0.59	0.63	0.68	0.61	0.62	0.62	0.61
Skip-GANomaly [6]	0.8	0.95	0.45	0.61	0.60	0.62	0.93	0.79	0.66	0.91	0.73
DADUGT [88]	0.75	0.96	0.78	0.72	0.88	0.8	0.83	0.96	0.93	0.91	0.85
CSI [195]	0.89	**0.99**	0.93	0.86	**0.94**	**0.93**	**0.95**	**0.99**	**0.98**	**0.96**	**0.94**
SSOE [196]	0.78	0.97	0.87	0.81	0.93	0.9	0.91	0.97	0.95	0.93	0.9
PANDA	**0.95**	0.85	**1**	**0.92**	0.92	0.9	0.9	0.91	0.89	0.86	0.81

problems presented in this work. Pixel values in input images are normalised to a mean and a standard deviation of 0.5. All models use ADAM momentum [193] except our Perceptual Loss model which uses Stochastic Gradient Descent (SGD) with momentum 0.9. Learning rates used are: 7×10^{-6} - Generator, 1×10^{-5} - Discriminator, and 1×10^{-4} - Perceptual Loss model. Training is performed on an Nvidia 1080TI GPU using a batch size of 15.

3.4 Results and Discussion

In this section we present both the qualitative and quantitative results of our PANDA method against prior methods of Semi-Supervised anomaly detection.

The AUPRC statistical score across the classical 'leave-one-out' anomaly detection tasks (MNIST / CIFAR-10) are outlined in Table 3.1 where it can be observed that our approach (PANDA) performs competitively with prior state-of-the-art approaches on these seminal, albeit unrealistic benchmark tasks. The qualitative results outlined in Table 3.1 provide further evidence against the comparison of model performance solely across these trivial 'leave-one-out' MNIST / CIFAR-10 based anomaly detection tasks. It can be seen that model performance is becoming decreasingly informative due to potential performance saturation among competing approaches. It can also be seen in this table, that methods get vastly different

Table 3.2: Results of models across Leaf disease [1] and X-ray Laptop Anomaly detection [3] image datasets as well as results across UCSDPed1 [10] pedestrian detection and crowd control video dataset using frame-level comparison [21].

Model	Loss	Image Dataset									
		Plant Village [1]					**Laptop X-ray** [3]				
		AUC	95% CI (AUC)	Average Rec_Err	Average Adv_Err	I/t(ms)	AUC	95% CI (AUC)	Average Rec_Err	Average Adv_Err	I/t(ms)
AE [14]	-	0.65	(0.60, 0.70)	0.56	-	6.9	0.21	(0.19, 0.23)	0.80	-	9.4
AnoGAN [4]	-	0.65	(0.65, 0.66)	0.45	0.88	7151	0.41	(0.39, 0.42)	0.4	0.92	7223
EGBAD [5]	-	0.70	(0.65, 0.67)	0.40	0.92	87	0.47	(0.42, 0.43)	0.41	0.94	89
GANomaly [2]	-	0.73	(0.68, 0.73)	0.39	0.75	28	0.49	(0.41, 0.51)	0.34	0.78	273
f-AnoGAN [124]	-	0.77	(0.65, 0.78)	0.12	0.72	65	0.50	(0.49, 0.53)	0.1	0.72	86
Skip-GANomaly [2]	-	0.77	(0.74, 0.77)	0.13	0.74	123	0.51	(0.48, 0.58)	0.11	0.68	112
PANDA-GAN	PWL	**0.78**	**(0.77, 0.78)**	**0.01**	0.99	15.2	0.42	(0.30, 0.48)	0.052	0.987	16.8
	$PL_g()$	0.74	(0.73, 0.75)	0.40	0.99	20	0.45	(0.29, 0.52)	**0.02**	0.66	36
	$PL_{ps}()$	0.75	(0.76, 0.78)	0.20	0.99	20.8	**0.51**	**(0.48, 0.55)**	0.045	0.78	30

Model	Loss	Video Dataset	
		UCSDPed1 [10]	
		AUC	EER
SF [188]	-	0.68	31
MPPCA [189]	-	0.77	40
MDT [190]	-	0.82	25
SRC [191]	-	0.86	19
AMDN [192]	-	0.92	16
PCA-NET GMM [21]	-	0.93	11.2
AED-GAN [85]	-	**0.97**	**8**
PANDA-GAN	PWL	0.95	35
	$PL_g()$	0.95	75
	$PL_{ps}()$	0.93	96

Table 3.3: AUPRC results across MVTEC [9] dataset.

Model	**Classes**															
	Bottle	**Cable**	**Capsule**	**Carpet**	**Grid**	**Hazelnut**	**Leather**	**Metal Nut**	**Pill**	**Screw**	**Tile**	**Toothbrush**	**Transistor**	**Wood**	**Zipper**	AUC_{avg}
AnoGAN [4]	0.8	0.48	0.44	0.34	0.87	0.26	0.45	0.28	0.71	1	0.40	0.44	0.69	0.57	0.72	0.56
GANomaly [2]	0.8	**0.71**	0.72	0.82	0.74	0.87	0.81	0.69	0.67	1	0.72	0.7	0.81	0.92	0.74	0.78
Skip-GANomaly [6]	0.94	0.67	0.72	0.8	0.66	0.91	0.91	0.79	0.76	1	0.85	0.69	0.81	0.92	0.66	0.81
DA-GAN [197]	**0.98**	0.67	0.69	0.90	0.87	**1**	**0.94**	**0.82**	0.77	1	0.96	**0.95**	0.79	**0.98**	**0.78**	**0.87**
U-Net [198]	0.86	0.64	0.67	0.77	0.86	1	0.87	0.68	0.78	1	**0.96**	0.81	0.67	0.96	0.75	0.82
PANDA-GAN	0.83	0.68	**0.98**	**0.95**	**0.95**	0.92	0.75	0.79	**0.95**	**1**	0.85	0.66	**0.9**	0.68	0.62	0.83

results across the classes. This could be due to overfitting by the models on a relatively simple dataset, or could be the effect of this data being more noisily sampled for inference. Random sampling is performed to obtain the test set, so it is not controlled in the same way as other datasets in this thesis.

Across the MNIST task, our PANDA method obtains state-of-the-art results across 40% of classes and obtains performance close to the other prior methods across the other classes while exhibiting uniform performance across all classes. Most noticeably is the result across the digit 9 whereby PANDA is close to 0.3 AUPRC higher than the next best performing prior method (ADAE [174]). Across CIFAR-10, our method obtains state-of-the-art in 30% of classes and matches closely with other such methods (DADUGT [88], CSI [195], and SSOE [196]) while also obtaining close to uniform performance across all classes.

By contrast, Table 3.2 outlines quantitative results across the challenging real-

world benchmark datasets of Plant Village [1], Laptop X-ray [3], and UCSDped1 [10] providing numerous statistical comparatives including Area Under Curve (AUC), the 95% confidence interval of the AUC, inference time (I/t, ms) per image. These datasets [1, 3] feature particularly subtle anomalies by nature and as such pose as challenging tasks for semi-supervised anomaly detection models.

PANDA obtains the highest AUC value across both image based datasets (Plant Village: 0.78- using Pixel-Wise Loss (PWL); Laptop X-ray: 0.51- using Problem Specific Perceptual Loss(PL_{ps})) in comparison to leading state-of-the-art methods [2, 4–6, 14, 124, 182] (Table 3.2). Over multiple evaluations, PANDA also obtains tighter confidence-intervals, which are calculated over 5 consecutive training sessions per model, compared to prior semi-supervised work illustrating our PANDA method can produce more stable and reliable results across the same dataset while other such approaches can suffer from sporadic performance at inference (Table 3.2). Observing the I/t(ms), the PANDA method is also significantly faster than prior methods.

While using the Plant Village dataset in our experiments, we combine all classes from this dataset into a binary categorisation with classes {Healthy, Diseased} and achieve the aforementioned AUC score of 0.78. This is unrealistic as farms generally tend to stay with fixed cycles of the same crop(s), so it would make sense to separate the individual leaf classes {Cherry, Potato, Corn, Strawberry, Grape, Tomato} and train across each of them independently. The results of this experiment are quantitatively outlined in Figure 3.7 which together with the obtained AUC value, shows the distributions of anomaly score for each class. Overall PANDA gains the following AUC values: Cherry:0.93, Potato:0.96, Corn:0.99, Strawberry:0.99, Grape:0.99 and Tomato:0.77. This shows that, for certain classes of leaves, the performance can increase to near-perfect categorisation of visual leaf disease when focusing on one particular leaf while training. All distributions of anomaly score show that the Healthy class distributions have a smaller mean and standard deviation than the Diseased counterparts, providing more confidence upon normal samples through the learned representations of normality than in diseased samples. Of particular interest is the performance over the Corn class where the distributions over the normal and

anomalous anomaly score are noticeably separable with the diseased distribution having a much larger mean and standard deviation than the normal scores. Across the Cherry class, the distributions tend to overlap more and the separation between the two distributions is difficult. The distribution over normal samples across the Tomato class has a dual peak and is not as classically 'normally' distributed like the other classes; This is the worst performing class with an AUC of 0.77 and could be due to the vast differences between the shapes of leaves which deviate significantly between samples within the dataset.

Figure 3.6 illustrates the anomaly segmentation masks generated by PANDA, outlining where visual disease features upon leaves. For each of the independent models trained on their respective class, we demonstrate the segmentation across both normal, healthy leaves (above) as well as diseased leaves (below) to demonstrate the stark difference in the appearance of the anomaly masks between both normal and diseased leaves. Overall, the healthy samples show little to no noise implying that no disease is present in the leaves. For their diseased counterparts, however, PANDA is able to detect the diseased regions with high accuracy. In particular across the Cherry class, the disease appears visually subtle, yet PANDA is still able to accurately detect it. Across the Tomato class, the worst performing class, the diseased parts are segmented, but also on some occurrences, the outline of the leaves (shape) are included in the anomaly segmentation, cementing the previous assumption that the 'double peaked' normal anomaly score distribution and thus the reduced AUC performance could be due to the non-uniform geometry appearance of healthy Tomato leaves within the dataset.

In Table 3.3, the quantitative AUC results across the MVTEC dataset can be observed. This is a challenging dataset due to the large variation in appearance of anomalies present in textures and objects. Some objects (carpet, hazelnut, screw) exhibit visually obvious anomalies, but other objects (wood, metal nut, toothbrush) feature subtle anomalies which are hard to detect. This is reflected in the results of various methods across this dataset with PANDA obtaining superior AUC performance across 6 classes.

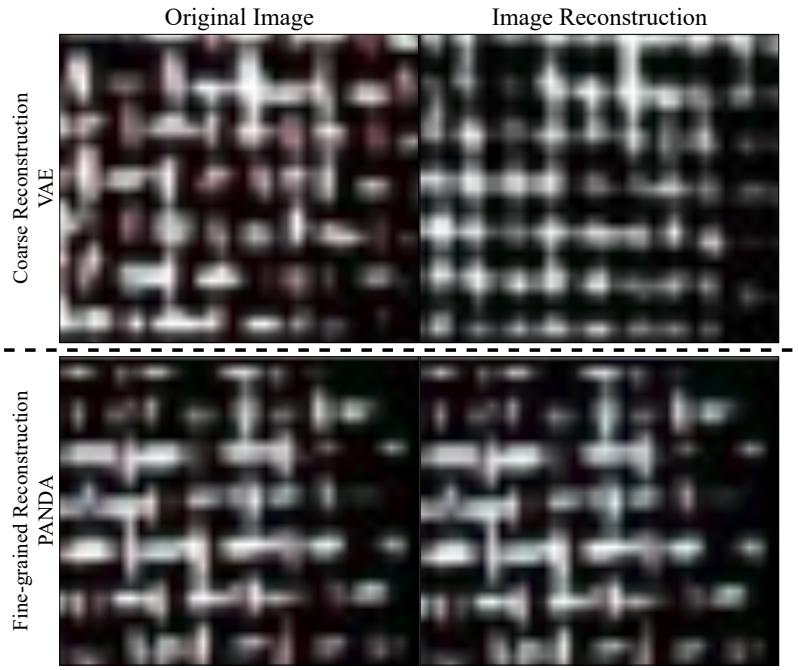

Figure 3.5: Illustration of the difference in detail preservation during reconstruction within the Carpet class of MVTEC [9] between the Variational Autoencoder (VAE) [14] and our PANDA architecture.

During the detection of subtle anomalies, it is important to preserve fine-grained details when reconstructing images. Approximating such details of the inputs produce coarse reconstructions with missing fine-grained image details which will result in inaccurate anomaly detection. Figure 3.5 illustrates this by presenting an image from the Carpet class of MVTEC [9] which is zoomed to 200% and cropped to show the details of the threads in the weave within the carpet. The example above shows a coarse reconstruction of the input image produced by a novel Autoencoder (AE) [14] model which appears to opt to almost uniformly place threads in a grid pattern within the reconstruction. Conversely, Figure 3.5 shows that our PANDA approach preserves the details of all the threads in the weave in the reconstruction. The preservation of these details is vital to detecting subtle errors which, in this class would be missing or damaged threads in a given carpet sample which could be vital for the quality and longevity of a given final product.

Figure 3.8 shows the qualitative results of PANDA with regards to the detection and segmentation of anomalous parts across different classes of the MVTEC dataset.

Within this figure, we show defective samples from 6 classes of the MVTEC dataset, namely {Hazelnut, Bottle, Capsule, Toothbrush, Screw, Wood} featuring a range from both visually obvious to visually subtle anomalies. Analysis of each class is as follows:

- Hazelnut: defects such as holes or missing parts are detected in the segmentation mask. The fourth image includes text that is also correctly detected as anomalous by PANDA.
- Bottle: chipped regions are detected in each example, however, key emphasis is put on the difference in light intensity caused by the chipped parts of the glass reflecting light differently as well as the chipped regions within the glass.
- Capsule: The faults in this class range from subtle such as the first and second image, up to being more visually obvious in the fourth image where a severe hole is detected. PANDA is able to detect all defective parts of the capsule. However on the third image, the text on the capsule is mistaken as anomalous.
- Toothbrush: Faults occur predominantly on the bristles of toothbrushes and as such are profoundly subtle by nature and difficult to spot due to the small size of the bristles. PANDA is able to locate the anomalous parts present in the toothbrushes. However, some noise is also included within the detection.
- Screw: The anomaly detections of this class are very noisy. Although our approach detects the anomalous regions such as the bent tip in the first image and the scratches in the other two, there is a lot of noise particularly around the corkscrew ridges. This seems strange as the Screw class gained the highest AUC of 1 during inference.
- Wood: The random grain patterns of the wood twinned with the visually subtle nature of the anomalies in these examples makes the Wood class incredibly difficult. Our approach does detect the pinholes in the wood, but the segmentations are faint, giving indication of low confidence in anomaly score.

Figures 3.10 and 3.9 show the anomaly mask produced over the Bottle and Hazelnut class of MVTEC [9], respectively. GANomaly [2] generates noisy detections over both classes with a lot of noise. The anomalous regions are successfully detected in both classes however, but it is difficult to pinpoint due to the noisy detection. on the other hand, Skip-GANomaly [6] and PANDA (Chapter 3) obtain very clean detections however, some of the underlying shape, especially the outline of the bottle is still visible in the anomaly mask. Across the Hazelnut examples, the outline of the object is however, not present in the anomaly mask, possibly due to it being a softer edge than the bottle edge. In both classes, both Skip-GANomaly and PANDA are able to successfully isolate the anomalous region however, detection is not perfect as there is still noise present.

3.4.1 Ablation Study

Our ablation study (Table 3.4) produces evaluation over individual components to our novel architecture with respect to variations in both loss function, our network architecture components ($E^0_{high} \cap D^0_{high}$, E^1_{high}, E^1_{low}) and our choice of discriminator architecture across two of the more challenging real-world anomaly detection task datasets. For comparison we include the DCGAN [125] discriminator architecture from GANomaly / Skip-GANomaly [2,6], which is the next best performing approach in terms of AUC across the same datasets (Table 3.2) to compare against our FGVC-based discriminator architecture choice.

Table 3.4: Ablation Study of PANDA-GAN across Plant Village [1] and Laptop Anomaly [3].

Model	**Dataset**											
	Plant Village						**Laptop Anomaly**					
	Loss			**Network Architecture**			**Loss**			**Network Architecture**		
	PWL	PL(g)	PL(ps)	$E^0_{high} \cap D^0_{high}$	E^1_{high}	E^1_{low}	PWL	PL(g)	PL(ps)	$E^0_{high} \cap D^0_{high}$	E^1_{high}	E^1_{low}
PANDA-GAN	0.75	0.75	0.76	✗	-	✗	0.38	0.42	0.43	✗	-	✗
	0.75	0.74	0.75	✗	-	✓	0.42	0.44	0.45	✗	-	✓
	0.75	0.74	0.74	✓	✗	✗	0.46	0.47	0.48	✓	✗	✗
	0.76	0.75	0.76	✓	✗	✓	0.46	0.44	0.43	✓	✗	✓
	0.77	**0.76**	**0.78**	✓	✓	✗	**0.50**	**0.52**	0.50	✓	✓	✗
	0.78	0.74	0.77	✓	✓	✓	0.42	0.45	**0.51**	✓	✓	✓
PANDA-GAN DCGAN Discriminator	0.77	0.75	0.74	✓	✓	✓	0.41	0.42	0.47	✓	✓	✓

From the results of Table 3.4, it can be observed that synergy exists between

components of our generator network obtaining the highest AUC value only when all three of our novel components are activated. Generally we see that the more components we activate in our architecture, the better the performance obtained during our ablation study. Overall, the problem-specific perceptual loss (PL_{ps}) performs better across the Laptop X-ray dataset by a clear margin from the other loss functions tested against. Across the Plant Village dataset, there is negligible difference between the pixel-wise loss and the problem specific perceptual loss. Both performed almost identically and gained a clear advantage over the general perceptual loss function (PL_g).

3.5 Conclusion

This chapter provides a thorough overview of the PANDA architecture. This is a method which is bespoke for the task of real-time detection of subtle visual anomalies present in real-world tasks which is still a difficult ongoing problem in the field of anomaly detection. Our method is an Adversarially trained Autoencoder based architecture which is able to reconstruct input with high-fidelity while remaining stable during training. We introduce three novel concepts to this method which include (1) A Fine-grained Visual Categorisation (FGVC) discriminator network to provide a harsher critic to the generator while training which promotes the generation of finer details of the image. (2) A residually connected dual feature extraction method within our generator module which carries low-level and high-level features forward in the architecture. (3) A perceptual loss function which captures differences in activation pattern between images and their reconstructions produced by the generator module.

Our exhaustive experimentation in this work show state-of-the-art performance across both real-world datasets and trivial leave-one-out anomaly detection tasks across MNIST [7] and CIFAR-10 [8] obtaining a state-of-the art AUC score across 40% and 30% of classes of the respective leave-one-out datasets. We also experiment on more realistic tasks namely: Plant Village [1], Laptop X-ray [3], UCSDped1 [10] and the MVTEC [9] datasets. Across Plant Village, our method is able to outperform all prior methods with an AUC score of 0.78. However, when splitting the individual leaf classes and training on them independently, we are able to obtain near-perfect anomaly categorisation of presented leaves. The Laptop X-Ray dataset is challenging due to the complexity of the images present in this task. PANDA is still able to gain the best performance with an AUC of 0.51, even though this result is only just better than random guessing. Across the UCSDped1 video task, PANDA gains competitive performance with an AUC of 0.95 compared to prior methods, illustrating the cross-domain ability of PANDA across video data. PANDA is able to gain state-of-the-art performance across 40% of classes of the MVTEC dataset. We also show that

PANDA is able to generate clean anomaly segmentation masks indicating where in given samples is anomalous with high precision as evidenced in Figures 3.6 and 3.8.

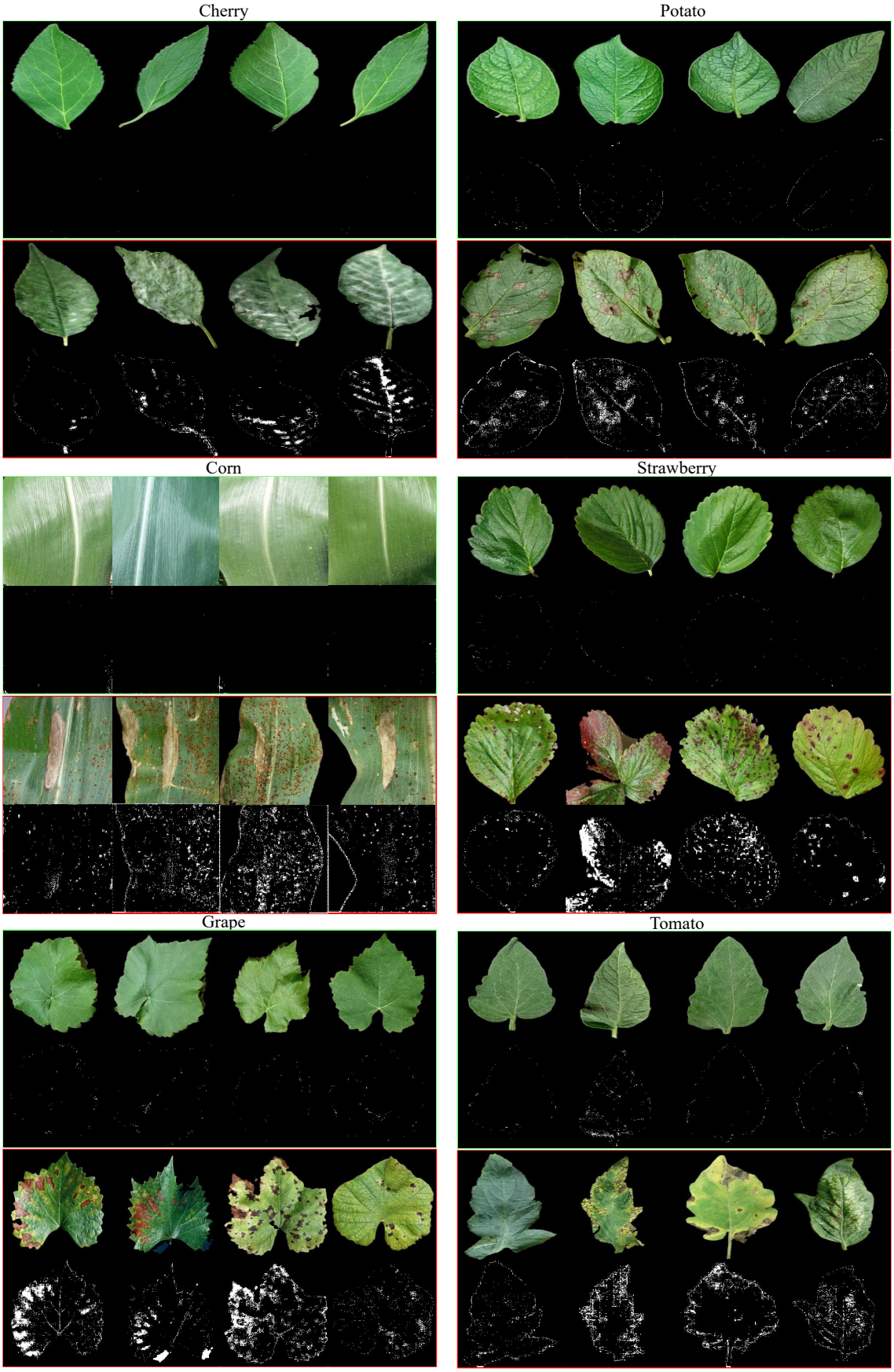

Figure 3.6: Anomaly segmentation masks across classes of the Plant Village [11] dataset outlining both healthy and diseased examples.

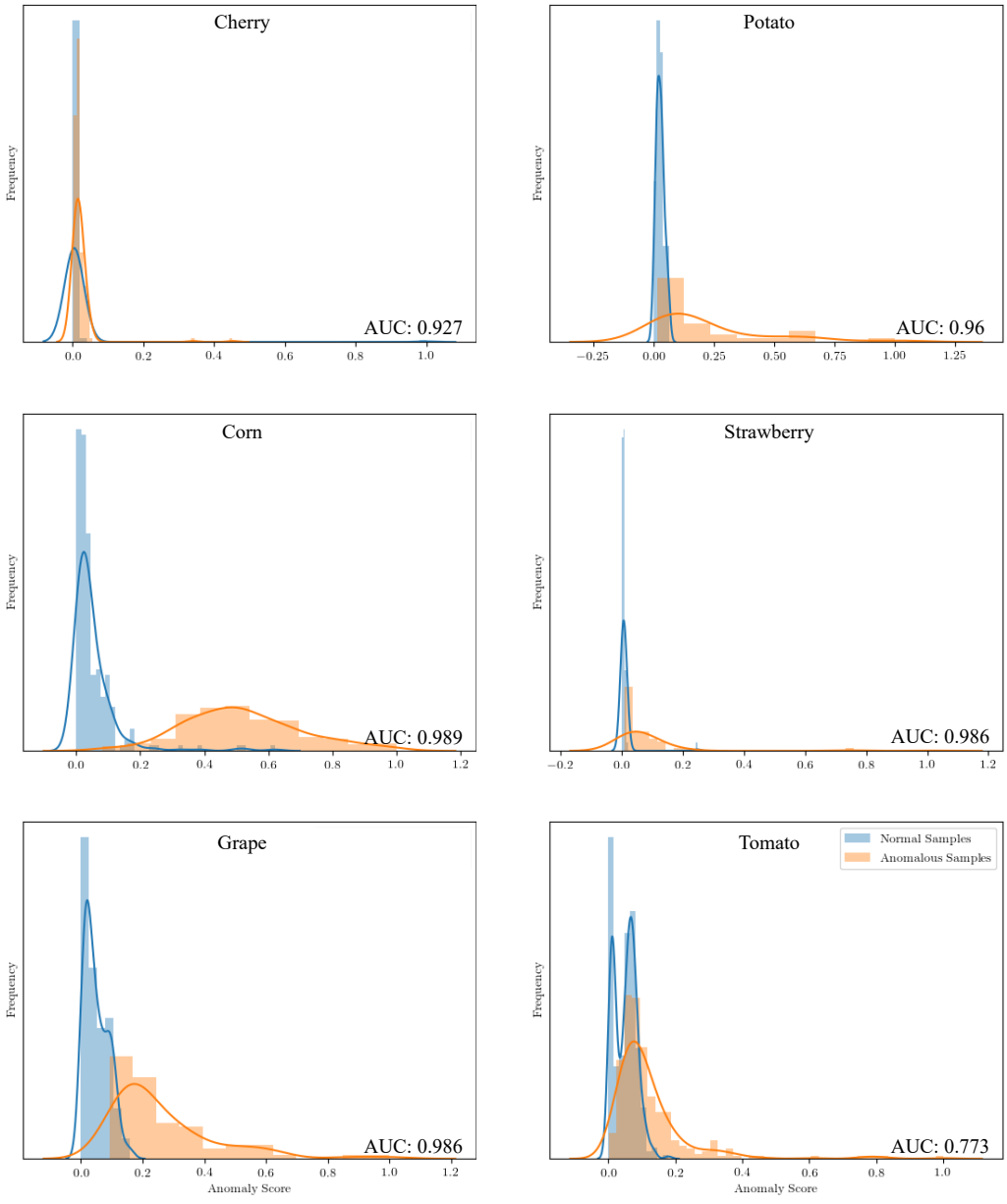

Figure 3.7: Anomaly score distributions together with AUC results of PANDA across classes of the Plant Village dataset.

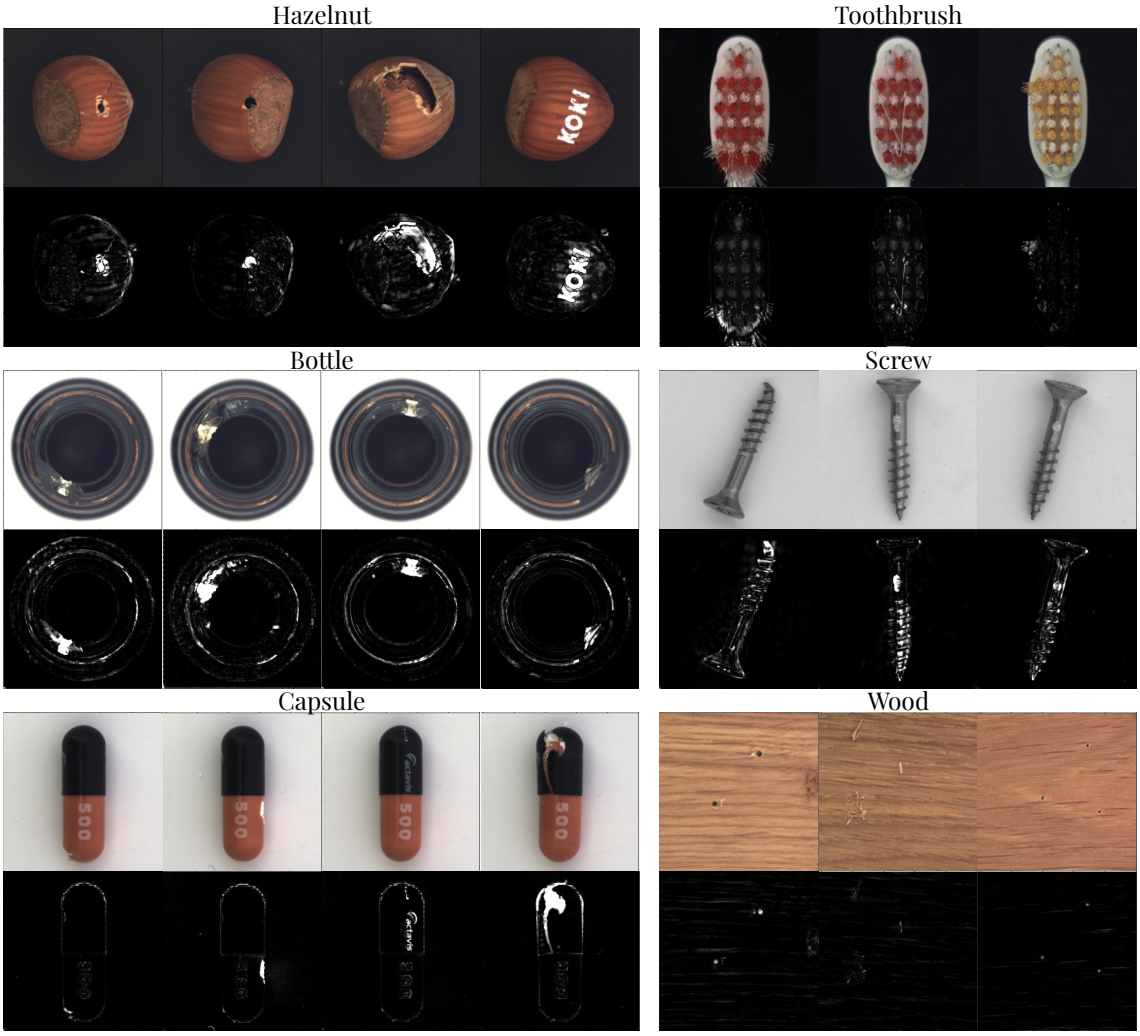

Figure 3.8: Anomaly segmentation masks obtained from PANDA of defective samples from classes (Hazelnut, Bottle, Capsule, Toothbrush, Screw and Wood) within the MVTEC dataset [9].

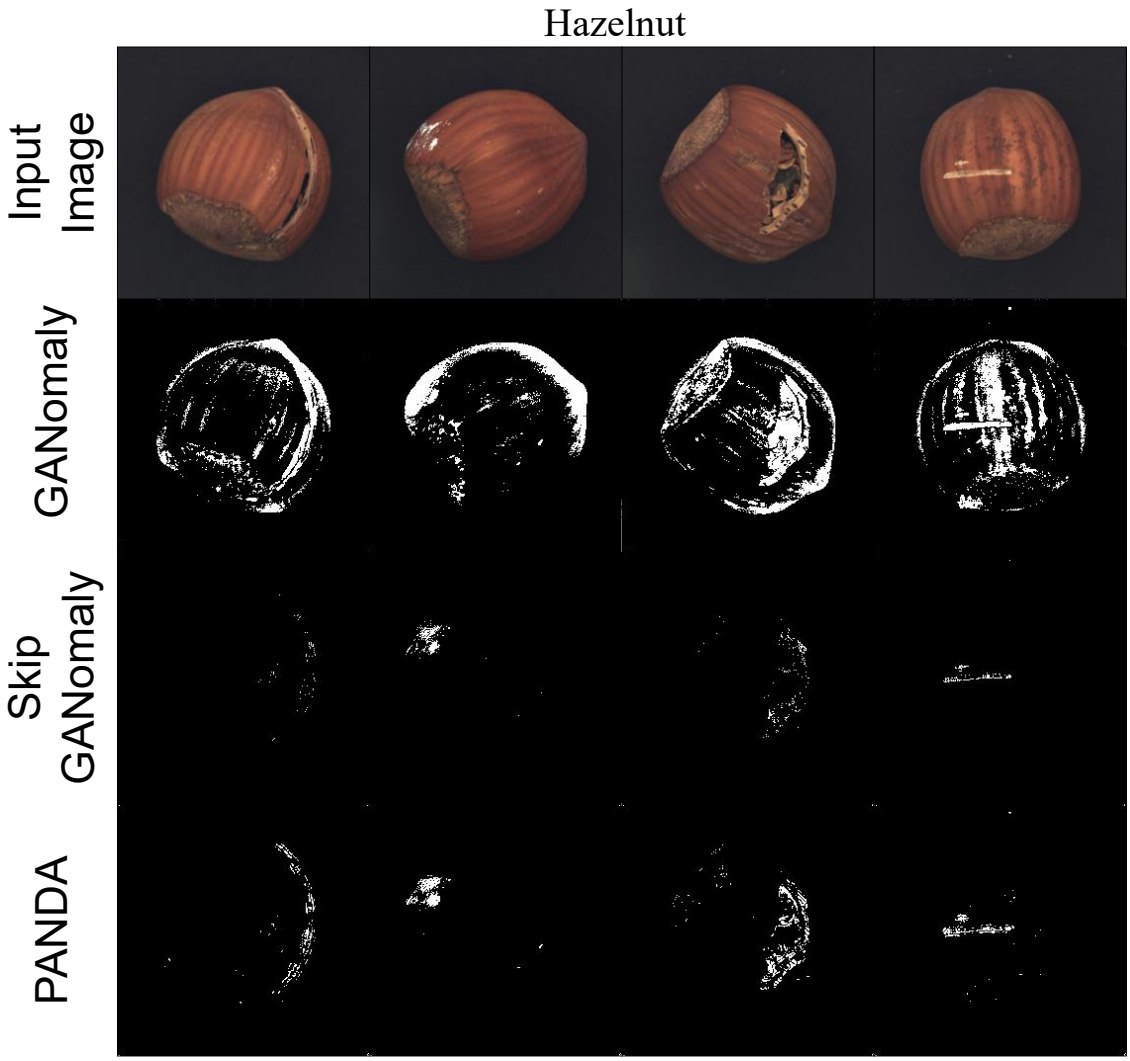

Figure 3.9: Comparison of anomaly mask quality between GANomaly [2], Skip-GANomaly [6] and PANDA (Chapter 3 across the Hazelnut class of the MVTEC dataset [9].

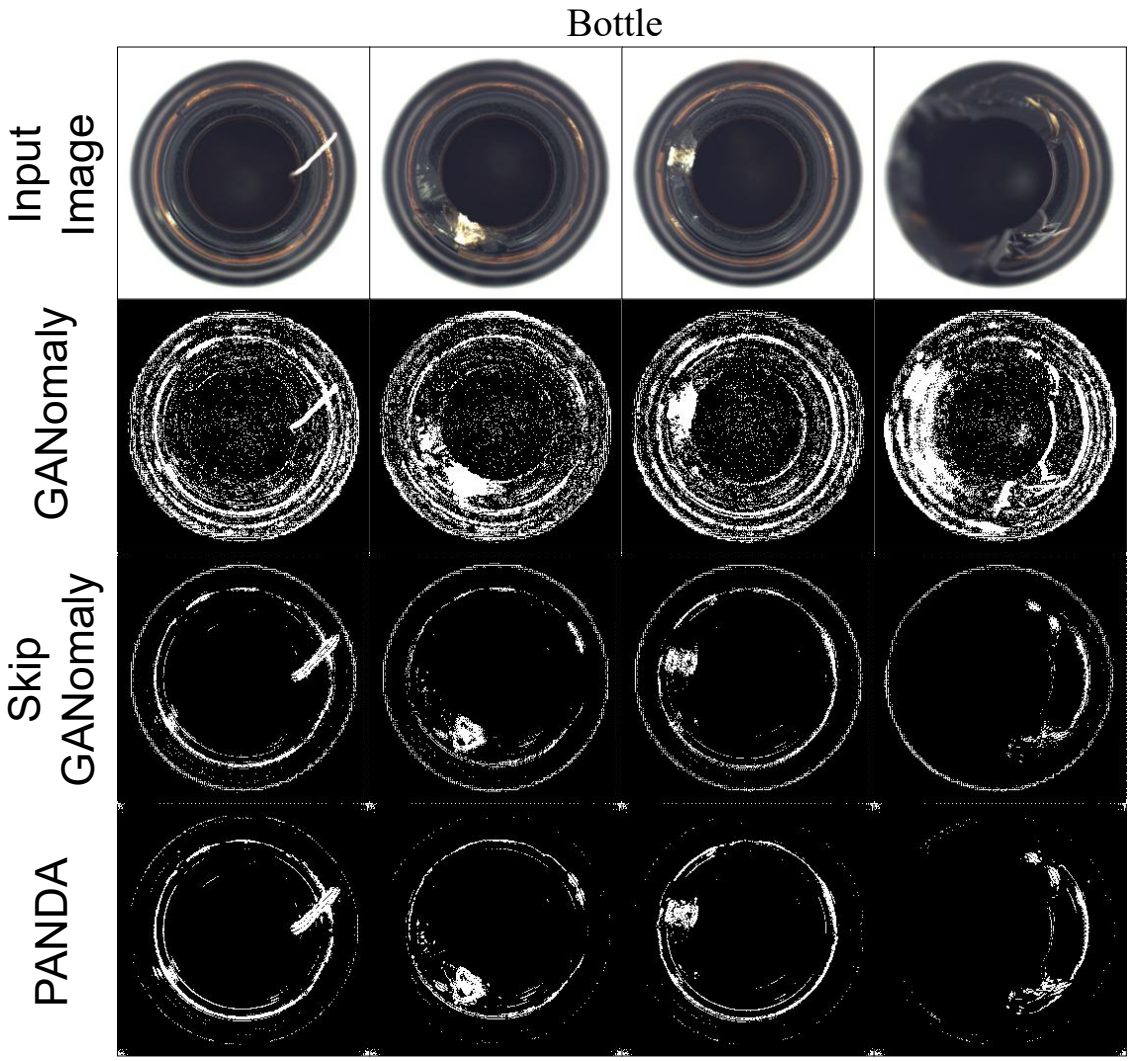

Figure 3.10: Comparison of anomaly mask quality between GANomaly [2], Skip-GANomaly [6] and PANDA (Chapter 3 across the Hazelnut class of the MVTEC dataset [9].

CHAPTER 4

Adversarially Learned Contrastive Noise for Robust Generative Semi-Supervised Anomaly Detection

4.1 Introduction

The task of anomaly detection is challenging due to deviations from normality being continuous and sporadic by nature. Anomalous space is an open-set continuous, infinite distribution of possible deviations from normality; meaning that strictly supervised classifiers, although performing well across tasks in anomaly detection [3,37] are restricted by their limited exposure to abnormal examples during training. As such, it is impossible for datasets to contain every possible deviation in the anomalous data thus, supervised (classification-based) approaches cannot generalise to the continuous nature in which anomalous samples may deviate from normality, meaning that there will always exist anomalous deviations in anomaly space which present as adversarial examples to such supervised methods.

Generative-based anomaly detection methods [2,4–6,124] train solely across normal examples in order to approximate the underlying distribution of normality. They work by learning meaningful features to solely represent normal samples which will cause a relatively small reconstruction error after decoding; conversely, the model will fail to reconstruct anomalous samples fully due to null exposure of the anomalous parts during training. As such, the reconstruction error between input and output provides a sound metric to measure anomalous deviation of presented samples. The benefit of this (semi-supervised) training is that normal (non-anomalous) data is often relatively inexpensive and plentiful to obtain within real-world anomaly detection tasks [2, 3]

Autoencoders (AE) are well-suited to the approximation of the underlying data distribution. They exhibit stability during training unlike their Generative Adversarial Network (GAN) [199] based counterparts which exhibit training difficulties such as mode-collapse or convergence instability [200]. Although AE are stable, they risk converging to a pass-through identity function (1) [15] for which the mapping from input x to output x' is a null function such that $\lim_{y \to 0} y = \mathcal{L}(x, x') \Rightarrow x \simeq x'$ where $\mathcal{L}$ is the reconstruction error. Although this can still learn underlying information about the distribution of the training data, this over-fitting negatively

affects performance in tasks such as semi-supervised anomaly detection.

Such functions allow a pass-through of features within presented samples which may not have been seen during training. The power of semi-supervised generative anomaly detection methods is obtained from their ability to fail to reconstruct anomalous parts of a sample as well as the normal parts [2, 201]. This repairing of anomalous parts causes severe shifts in pixel values of a given sample in anomalous samples, allowing the amount that the pixels shift between input and reconstruction to act as a meaningful way in which to measure anomaly score of a presented sample. During overfitting to an identity function. However, anomalous regions will be mapped through to the reconstructions by the model, meaning that the pixel shifts will only vary slightly and the anomaly scores will be lower for anomalous samples at inference.

To prevent this, Denoising Autoencoders (DAE) [15] are trained to produce unperturbed reconstructions from purposefully noised input. This applies a level of regularisation to the AE such that it cannot easily converge to a trivial solution and allows an AE to become invariant of noise in the input as well as yielding more robust and meaningful representations across normality [31, 32]. Bengio *et al.* [15] states that corrupting an observed random variable X into $\hat{X}$ using conditional distribution $C(\hat{X}|X)$ is actually training the denoising autoencoder to estimate the reverse conditional $P(X|\hat{X})$. The noising process can be defined as a conditional distribution process which corrupts a given input X into a noisy version $\hat{X}$. Examples of corruption include Gaussian noise [120, 202, 203], or masking noise such as dropout applied to pixel values of the input [106]. Such methods add notable and proven regularisation to denoising autoencoders, but are stochastic by nature and thus offer randomly assigned noise during training without considering the input distribution with which the denoising autoencoder is being trained to reconstruct. Subsequently, these methods do not tailor the added noise to the problem set trained on, instead opting for a *'one-size-fits-all'* approach with arbitrarily added noise. If the task is to denoise certain pixel-level occurring noise such as speckle or artifacts from compression within an image, then methods implementing randomised pixel-level noise

perform well [203–205]. Within reconstruction-based anomaly detection, however, anomalous instances seldom feature at the individual pixel level and instead tend to cluster into local regions of anomalous pixels. As such, a reconstruction-based approach needs to be able to reconstruct out-of-distribution neighbourhoods of pixels such that the final image represents a non-anomalous 'repaired' version of the anomalous sample.

Adding noise to input images in the task of semi-supervised anomaly detection has been explored previously [31, 32, 206]. The Adversarially Robust Autoencoder (ARAE) [31] works by forcing perceptually similar samples closer in their latent representations by crafting adversarial examples during training by perturbing samples in the dataset that are constrained with respect to the input samples to be 1) perceptually similar to the input, but have 2) maximally distant latent encodings. Although results of ARAE are competitive with prior methods [4, 33, 34, 59] despite having a simpler architecture, producing perturbations which fit with such tight requirements is very computationally demanding and as such, ARAE requires more compute overhead during training.

One-Class Learned Encoder-Decoder (OLED) [32] dynamically corrupts the input data during each step within training by masking through the use of a second network called the mask module. OLED works similarly to the Context Autoencoder (CAE) [206] where instead of being corrupted by noise, patches of the input images are randomly masked and the CAE must learn to inpaint this randomly masked region in conjunction with the reconstruction task [32, 206]. OLED, however, tackles the disadvantage of CAE whereby random patches are masked during training hence maximally important regions of the image are not consistently masked leading to sub-optimal representations [32].

The OLED mask module creates masks that mask important regions of the input by maximising the reconstruction error of the sequential denoising module placed after the masking module. This enables consistently optimal masking of the input through the output of the masking module in order to enable the later denoising module to reconstruct masked regions based on the context of non-anomalous sur-

rounding features. The produced masks from the OLED mask module are, however, discrete due to thresholding the output activations of the mask module with a step function. This produces masks which appear the same across every dataset and are not tailored to the input dataset being trained on, also masking important regions as zeros removes all information from important regions in the image. As such, the denoising module will have significantly lower exposure to such regions during training which may be required during inference.

In this chapter, we extend the notion of denoising perturbed input for use in reconstruction-based anomaly detection and overcome disadvantages present in prior approaches by using a simple denoising module $G_{denoise}$ which is tasked to classically reconstruct perturbed input corrupted through adversarially learned patches of noise which are additively applied to the input. Such patches are produced through the noise generator module G_{noise} which is trained to produce optimal and continuous-valued obfuscation to the input during training. At each training step, G_{noise} is updated with the gradients of the input to produce bespoke noise masks which obfuscate the input as to maximally increase the error between the input x and the denoised reconstruction x' produced by the denoising module. Conversely, the denoising module $G_{denoise}$ is tasked to reduce the error between x and x' from perturbed input, hence to create denoised output from the corrupted input.

The G_{noise} module is only ever exposed to the gradients of the input and not the raw input itself to prevent it from fitting to the identity of each input image, producing zeros or ones everywhere in the input into $G_{denoise}$ which would trivially destroy all information in the input making it impossible for $G_{denoise}$ to reconstruct individual inputs faithfully. The noise produced by G_{noise} is additively applied to the input via a weighted sum so that the relative pixel intensities cannot be used to discriminate between clean and obfuscated image parts by the model. Further theories of why the G_{noise} model does not produce noise that completely destroys the image is that the example produced, may cause a more damaging effect in the weights of $G_{denoise}$, meaning that the network is maximally disrupted by these masks, however, after further refinement and exposure to these, the $G_{denoise}$ model actually

improves the robustness.

4.2 Approach

Common autoencoders (Figure 4.1 A) simply map a given image x to a compressed representation, and then use this representation to map back into a reconstruction of the original image x'. Conventional autoencoders suffer from overfitting when the mapping from $x \rightarrow x' \sim 1$, thus $x \sim x'$.

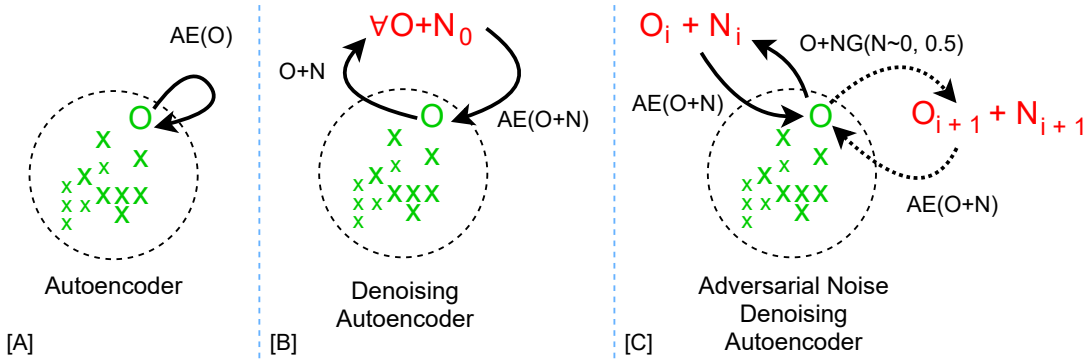

Figure 4.1: Comparison between prior methods (A [14], B [15]) and ours (C).

To prevent this, Denoising Autoencoders (DAE) (Figure 4.1 B) initially corrupt the input images x to x+noise. The DAE then reconstructs $x \rightarrow x'$ from $x + noise$. The noise in these methods is typically drawn from a Gaussian, Speckle or Random distribution. Methods implementing such denoising schemes into reconstruction-based anomaly detection have offered insights into how useful this noise is to the task of reconstruction-based anomaly detection. The work by Adey *et al.* [201] utilises a suite of noising approaches for use in their unique denoising approach. It can be seen in their results, however, that using Gaussian and Speckle noise actually negatively impacts the ability of the anomaly detection. This could be due to the aforementioned issues with pixel-level noising approaches within anomaly detection whereby anomalous regions seldom present as pixel-level deviations and instead present as neighbourhoods of anomalous pixels within images.

The method presented in this chapter, the Adversarially Learned Continuous Noise (ALCN) (outlined in Figure 4.1 C) utilises a unique continuous adversarially

learned noise which maximally obfuscates the input at each step prior to training the denoising module. This training method allows the simultaneous training of both modules so that the noise module can add continuous increasingly bespoke noise to the input while the denoising module increases in knowledge of how to reconstruct clean images from such noise.

The proposed method within this chapter is outlined in Figure 4.2. In our approach, we utilise a Denoising Autoencoder Generator ($G_{denoise}$) network which takes noisy images as input and outputs clean images, together with a GAN-like Noise Generator (G_{noise}) network which takes random samples from a Gaussian distribution as input and outputs the noise mask which maximally increases the reconstruction error when added via weighted sum to the input.

These modules are adversarially trained concurrently using Algorithm 1. In a given step, the weights of G_{noise} are updated first with gradient ascent with respect to the reconstruction error of $G_{denoise}$ from the previous training step so that at the next step, G_{noise} updated to produce a noise mask which maximally increases the reconstruction error for the $G_{denoise}$ to then reconstruct via gradient descent on the reconstruction error. The level of corruption in the current step differs only slightly from the corruption from the previous step. Additionally, over time, the noise produced by G_{noise} fits closer to the style of the input distribution, producing more bespoke noise for a given task.

Training across dataset $x \in \mathbb{R}^{B\times C\times H\times W} \in X$ where $\{B, C, H, W\}$ represent the batch size, number of channels, height and width respectively, starts by training the noise generator G_{noise}. A linear vector of size $B \times 256$ random variables ϕ is sampled from a standard Gaussian normal distribution $\phi \sim N(\mu : 0, \sigma : 1)$. We tried other sizes of input linear vector, namely, {64,128,256,512} and found that 256 was the best performing size in our experiments. This vector is fed through G_{noise} to produce noise n of shape $\mathbb{R}^{B\times C\times H\times W}$. The added Sigmoid layer ($\frac{1}{1+e^{-l}}$) on the final layer of G_{noise} binds the noise values continuously between $[0, 1]$. We combine the noise mask n to the input image x using a weighted sum by using the linear blending operator $x_{perturbed} = \alpha(x) + (1 - \alpha)(n)$ where α is randomly sampled on each step

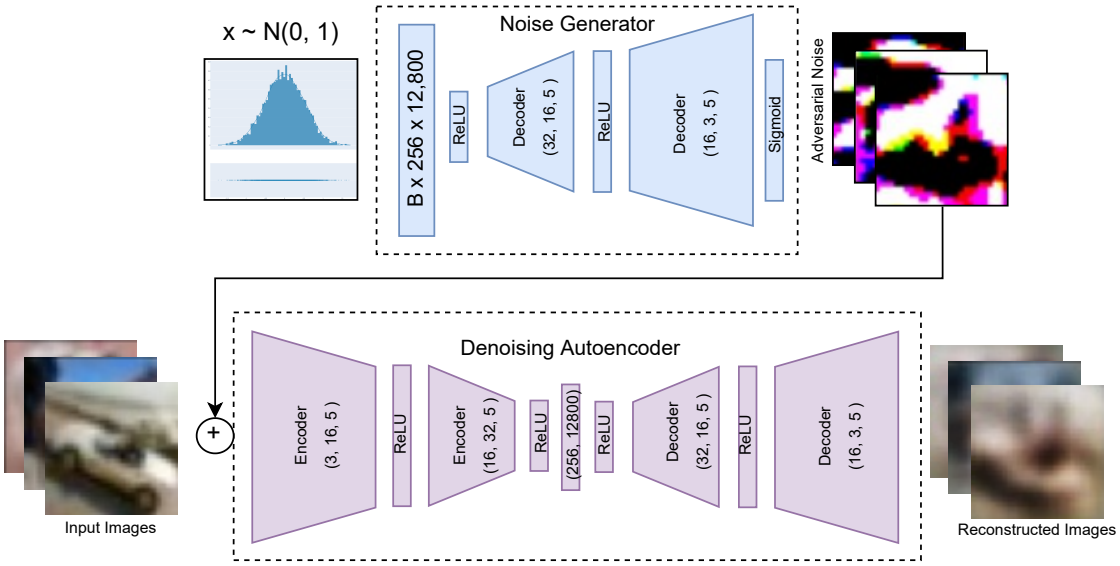

Figure 4.2: Overview of adversarial noise learning architecture featuring: top-Noise Generator Module G_{Noise}, bottom- Denoising module ($G_{denoise}$).

within bounds $\alpha \rightarrow [0.2, 0.9] \in \mathbb{R}^+$. Figure 4.3 illustrates how differing values of α affect the visibility of the added noise. The linear blend operator ensures that the magnitude of the values of $x_{perturbed}$ match with the pixel intensities of x and n. Values of x are normalised with 0 mean and unit variance meaning that the values of $x_{perturbed}$ are such that $G_{denoise}$ is prevented from discriminating between the noise corrupted pixels and the original image pixels based on differing pixel intensity.

Although setting alpha to be static during training could allow G_{noise} to theoretically perfectly optimise the generated noise n to destroy all information in image $x \in X$ such that all values in $x_{perturbed}$ are set to 1 such that $n = (\frac{1-\alpha \cdot x}{1-\alpha})$. This would be very rare, as G_{noise} does not have access to the input data at any point, only the gradients of the input with respect to the reconstruction error. However, it is interesting to think about this aspect within a theoretical situation.

The $x_{perturbed}$ cannot converge to all zeros where $n = -(\frac{\alpha \cdot x}{1-\alpha})$ due to the logical argument that the values of noise n produced by G_{noise} are bound to [0,1] $\in \mathbb{R}^+$ because of the Sigmoid layer on the output of G_{noise} and x is such that $\forall x_i \in x \rightarrow \{0, 1\}, \exists x_i \in x \,|\, x_i = 1$ implying that if $(x_i \in x = 1)$ then $n = \frac{-\alpha}{1-\alpha} \Rightarrow n < 0 \; \forall \alpha \; \therefore$ $n \notin \mathbb{R}^+$. Put simply, there are values of 1 within the images such that if noise n were

Algorithm 1 Adversarial Noise Training

W$\{G\} \leftarrow$ init ▷ Initialise G randomly
W$\{N_G\} \leftarrow$ init ▷ Initialise N_G randomly
Train One Epoch:
for mini-batch: $x \subset X$ **do**
    weights$\{N_G\} \leftarrow$ True
    weights$\{G\} \leftarrow$ False
    $\alpha \leftarrow [0.2, 0.9]$ ▷ Randomly select α
  z $\leftarrow N(\mu = 0, \sigma = 0.5)$ ▷ $|z| = \{|x|, 256\}$
  output $\leftarrow G((1-\alpha)N_G(z) + \alpha x)$
  W$\{N_G\} \xleftarrow{\text{backpropagate}} \text{Optim}_{N_G}(-\mathcal{L}(x, output))$
  weights$\{N_G\} \leftarrow$ False
  weights$\{G\} \leftarrow$ True
  output $\leftarrow G(N_G(z) + x)$
  W$\{G\} \xleftarrow{\text{backpropagate}} \text{Optim}_G(\mathcal{L}(x, output))$
end for

to produce all zeros after adding, then the value of noise would have to be negative and this is not possible with a Sigmoid function.

To ensure convergence to such a trivial hypothetical solution of convergence $n = (\frac{1-\alpha \cdot x}{1-\alpha})$ is even more unlikely, randomness can be applied to certain components in the architecture: 1) setting the value of α to be randomly continuously sampled for each step during training will allow for no solution to the hypothetical convergence; and 2) the input of G_{noise} being sampled from a Gaussian distribution $N(0, 1)$ which applies some level of randomness during sampling of the noise.

The $x_{perturbed}$ is then used as input to $G_{denoise}$ to reconstruct x from $x_{perturbed}$, reversing the corruption caused by G_{noise}. The corrupted image $noise_{x,n}$ is encoded to the latent vector z and then subsequently decoded into a synthetic reconstruction x'.

Adversarial learning is accomplished by the mini-max optimisation between the $G_{denoise}$ and G_{noise} modules. Weights of $G_{denoise}$ are optimised to minimise $\mathcal{L}$, the reconstruction error between x and x' whereas the weights of G_{noise} are conversely optimised to maximise $\mathcal{L}$. Loss terms in the overall loss are given scalar regularisation terms λ_0 and λ_1 for losses $\mathcal{L}_{G_{denoise}}$ and $\mathcal{L}_{G_{noise}}$ respectively. The overall optimisation function in this work is:

$$\underset{G_{denoise}}{argmin}\ \underset{G_{noise}}{argmax} = \mathcal{L}_{G_{denoise}}(x, x')\lambda_0 + \mathcal{L}_{G_{noise}}(x, x')\lambda_1 \tag{4.1}$$

Figure 4.3: Visualisation of the output of the linear blend operator between a sample Bottle from the MVTEC [9] and the corresponding adversarial noise at increasing levels of α.

This method of training encourages the noise generator to produce masks which optimally corrupt the input. Such optimal noise makes the denoising process more difficult as the denoising module must not only learn meaningful features of the input data, but also how to reconstruct maximally corrupted out-of-distribution parts within the input into clean input. It encourages the denoising module to not carry forward out-of-distribution (anomalous) features to the reconstruction during inference which is important for the task of reconstruction-based anomaly detection. Our experiments show a clear improvement in performance while using this training scheme with adversarially learned noise.

4.3 Implementation Details

Our method is compared across the MNIST [7] and CIFAR-10 [8] datasets due to their inherent simplicity while training as well as giving sufficient bench-marking for the evaluation between the techniques included in this work. Evaluation is conducted in two protocols following from established methods for 'leave-one-out' anomaly detection tasks. During protocol 1 (1 vs. rest), one class is regarded as anomalous and remaining classes are normal as performed by: [2, 4–6, 13, 124]. Protocol 2 (rest vs. 1) as performed by: [31–35] is the opposite in that one class is

normal and the nine remaining classes are anomalous.

The split ratio for the data is 80 : 20 for training and testing respectively as conducted by [2,5]. During training, the Adam optimiser is used for both $G_{denoise}$ and G_{noise} with learning rates 1×10^{-5} and 8×10^{-3} respectively, we chose these learning rates by hyper-parameter optimisation during training. An image resolution of 28×28 is implemented throughout 'leave-one-out' anomaly detection tasks [7,8]. We implement a larger resolution of 256×256 across MVTEC [9] and Plant Village [11]. A batch size of 4096 is employed across MNIST and CIFAR-10 and a batch size of 16 is used across MVTEC and Plant Village during training on an Nvidia GTX 1080 TI GPU. We evaluate our method using the Area Under Receiver Operator Characteristic (AUROC) metric.

4.4 Results

Extensive comparison of the results of our method compared to prior methods are outlined in Tables 4.1, 4.2, 4.3, 4.4. Tables 4.1 and 4.2 outline the quantitative results of the ALCN method applied to the DAE model across both MNIST [7] and CIFAR-10 [8] 'leave-one-out tasks' across both protocol 1 (9 normal/1 anomalous) and protocol 2 (1 normal/9 anomalous). Across the real-world anomaly detection tasks outlined in this paper [9, 11], Table 4.3 outlines the quantitative results of our method across the MVTEC-AD [9] industrial inspection dataset and Table 4.4 presents the results across the Plant Village dataset [11].

4.4.1 Leave One Out Anomaly Detection

Protocol 1: Table 4.1 outlines the results of each approach across the MNIST and CIFAR-10 datasets. We begin by comparing our vanilla DAE approach without any noise regularisation and this results in an AUC_{avg} of 0.69 across MNIST and 0.61 across CIFAR-10. The performance of the DAE is weak compared to other methods in the table. This result acts as a control to show how our adversarial noising approach can help to gain better anomaly detection capability during inference.

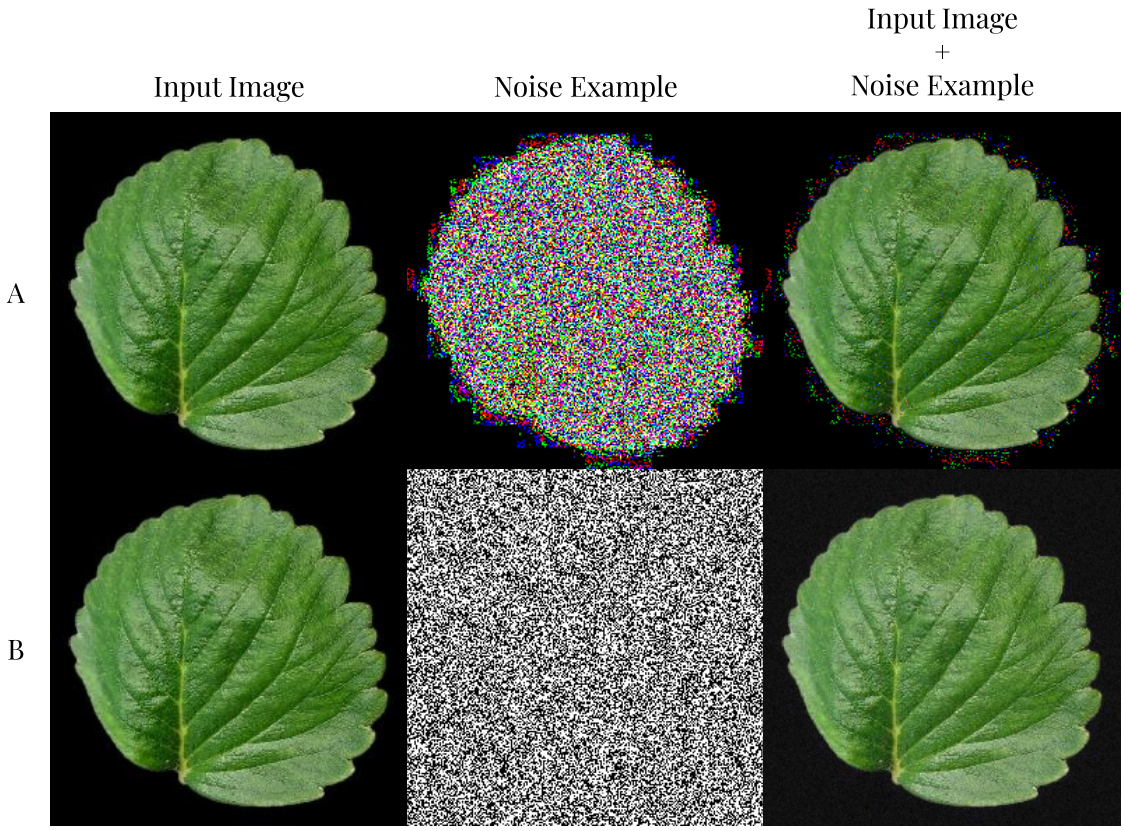

Figure 4.4: Overview of the appearance of different noising techniques implemented in this work, namely A) Speckle noise, B) Gaussian Noise

Other controls in this experiment involve applying Gaussian noise which obtains an AUC_{avg} of 0.70 on MNIST and 0.57 on CIFAR-10 as well as Random Speckle noise for which the DAE architecture obtains an AUC $_{avg}$ of 0.68 and 0.60 across MNIST and CIFAR-10 respectively. It can be seen that both of these noising approaches hardly impact performance of the DAE model. This could be attributable to the fact that the Gaussian and Random Speckle noise act on a pixel level. As the DAE is unlikely to perform a pixel-perfect reconstruction of input features, there will always be some level of blurring in the synthetic reconstruction produced by the DAE. For this reason, such noised pixels may not carry enough significance through the backward gradients of the network implying that the pixel-level noise negligibly impacts the convergence of the DAE during training. This can be observed qualitatively in Figure 4.4 which shows in rows A and B for Speckle and Gaussian respectively, the resulting image + noise sample shows little to no perceptual deviation from the

input image.

Our ALCN approach applied to the DAE architecture achieves the best AUC score across 90% of the classes with an average AUC of 0.89 and produces the best scores on 60% of classes of CIFAR-10 with an average AUC score of 0.67. This shows that our adversarial method of learning synthetic noise and applying it to the training scheme can assist in improving anomaly detection capability across the trivial tasks of MNIST and CIFAR-10 used for their relative aforementioned simplicity to act as a sanity check and initial benchmarking for our approach compared to other techniques.

Protocol 2: Table 4.2 presents the results across the protocol 2 variant (1 normal/9 anomalous) across both MNIST and CIFAR-10. This inference paradigm is used in a number of works on denoising techniques for reconstruction-based leave-one-out anomaly detection. However, it offers an easier challenge than Protocol 1. The performance across methods [31–35] on this protocol are approaching the performance ceiling of the task and thus, comparisons of model performance are not as useful as protocol 1; despite this, as methods tailored to denoising-based anomaly detection [31,32] are inferred using this protocol, we too evaluate across this protocol for direct comparison.

In Table 4.2, it can be seen that the DAE + ALCN method obtains an AUC_{avg} of 0.989 across MNIST and an AUC_{avg} of 0.742 across CIFAR-10, outperforming all prior methods including the next best model OLED [32] which uses discrete, thresholded noise, as previously stated in this work. This gives illumination as to the benefit of using continuous contrastive noise while training over using discrete univariate masking of maximally important regions within the input images. The noise produced by our approach also acts as a masking for important image regions, but also takes into account the nature of the input distribution while crafting continuous valued noise.

Model	MNIST [7]										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	AUC_{avg}
VAE [14]	0.55	0.10	0.63	0.25	0.35	0.30	0.43	0.18	0.50	0.10	0.34
AnoGAN [4]	0.61	0.30	0.54	0.44	0.43	0.42	0.48	0.36	0.40	0.34	0.43
EGBAD [5]	0.78	0.29	0.67	0.52	0.45	0.43	0.57	0.40	0.55	0.35	0.50
GANomaly [2]	0.89	0.65	0.93	0.80	0.82	0.85	0.84	0.69	0.87	0.55	0.79
ADAE [174]	0.95	0.82	0.95	0.89	0.82	**0.91**	0.89	0.80	0.93	0.63	0.86
DAE	0.84	0.97	0.79	0.64	0.53	0.61	0.66	0.55	0.71	0.57	0.69
DAE+Random Noise	0.84	0.93	0.66	0.66	0.52	0.62	0.72	0.56	0.75	0.53	0.68
DAE+Gaussian Noise $\sim N(0, 0.5)$	0.88	0.97	0.77	0.66	0.55	0.62	0.75	0.55	0.71	0.57	0.70
DAE + ALCN	**0.97**	**0.97**	**0.96**	**0.89**	**0.85**	0.88	**0.92**	**0.80**	**0.93**	**0.76**	**0.89**
Model	**CIFAR-10 [8]**										
	Plane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**	AUC_{avg}
VAE [14]	0.59	0.40	0.52	0.44	0.46	0.50	0.38	0.51	0.64	0.49	0.49
AnoGAN [4]	0.51	0.49	0.41	0.40	0.34	0.39	0.34	0.41	0.56	0.51	0.44
EGBAD [5]	0.58	0.52	0.39	0.45	0.37	0.49	0.36	0.54	0.42	0.55	0.47
GANomaly [2]	0.63	0.63	0.51	**0.58**	0.59	0.62	0.68	**0.61**	0.62	0.62	0.61
ADAE [36]	0.63	**0.73**	0.55	**0.58**	0.50	0.60	0.60	**0.61**	0.62	0.67	0.61
DAE	0.50	0.68	0.61	0.55	0.69	0.53	0.62	0.60	0.63	**0.71**	0.61
DAE+Random Noise	0.63	0.53	0.54	0.54	0.65	0.59	0.64	0.55	0.66	0.63	0.60
DAE+Gaussian Noise $\sim N(0, 0.5)$	0.57	0.68	0.57	0.54	0.65	0.54	0.55	0.52	0.57	0.53	0.57
DAE + ALCN	**0.77**	0.71	**0.62**	0.57	**0.72**	**0.62**	**0.72**	0.60	**0.66**	0.69	**0.67**

Table 4.1: Quantitative results (class name indicates AUC, AUC_{avg} of all classes) of models across MNIST [7] (upper) and CIFAR-10 [8] (lower) datasets (Protocol 1).

	MNIST	**CIFAR-10**
Method	**AUC$_{avg}$**	**AUC$_{avg}$**
DSVDD [35]	0.948	0.648
OCGAN [34]	0.975	0.733
LSA [33]	0.975	0.731
ARAE [31]	0.975	0.717
OLED [32]	0.985	0.671
DAE + ALCN	**0.989**	**0.742**

Table 4.2: Quantitative results (AUC_{avg}) of models across MNIST [7] (left) and CIFAR-10 [8] (right) datasets (Protocol 2).

4.4.2 Real-world Tasks

Next we review the performance of the DAE+ALCN approach across challenging, fine-grained real-world tasks which act as a more realistic environment in which to evaluate and compare the performance of models.

MVTEC-AD Industrial Inspection Dataset

In this experiment we compare our DAE + ALCN method against prior semi-supervised anomaly detection methods across the MVTEC-AD task [9] which is a task where the complexity and variability of the anomalies present in this dataset vary dramatically from both visually obvious to visually subtle between examples.

We use this dataset for this reason. The goal is to not only reconstruct small anomalies such as blemishes or scuffs, but also have the ability to reconstruct entire missing, or damaged parts of the anomalous object in question. For this reason, it offers the perfect environment to test the experiment of whether the adversarial denoising (ALCN) approach could enable the DAE to reconstruct both of these such anomalous instances while not carrying the anomalous parts through to the reconstruction following decoding.

The results of this experiment are shown in Table 4.3. It can be observed that DAE + ALCN obtains the highest average AUC score of 0.83, outperforming all other methods on 10 out of the 15 classes in MVTEC-AD. Although we have a significantly simpler architecture than the prior methods outlined in this table, our method still improves the anomaly detection capability significantly such that we are able to surpass the performance of such methods.

Model	MVTEC-AD [9]																
	Bottle	**Cable**	**Caps.**	**Carpet**	**Grid**	**H'nut**	**Leath.**	**M'nut**	**Pill**	**Screw**	**Tile**	**T'brush**	**T'sistor**	**Wood**	**Zipper**	AUC_{avg}
VAE [14]	0.66	0.63	0.61	0.51	0.52	0.30	0.41	0.66	0.51	1	0.21	0.30	0.65	0.87	**0.87**	0.58
AnoGAN [4]	0.80	0.48	0.44	0.34	0.87	0.26	0.45	0.28	0.71	1	0.40	0.44	0.69	0.57	0.72	0.56
EGBAD [5]	0.63	0.68	0.52	0.52	0.54	0.43	0.55	0.47	0.57	0.43	0.79	0.64	0.73	0.91	0.58	0.60
GANomaly [2]	0.89	0.76	0.73	0.70	0.71	0.79	0.84	0.70	0.74	0.75	0.79	0.65	0.79	0.83	0.75	0.76
Skip-GANomaly [6]	0.93	0.67	0.71	0.79	0.65	0.90	**0.90**	0.79	0.75	1	**0.85**	**0.68**	**0.81**	0.91	0.66	0.80
DAE_{256} + ALCN	**0.94**	**0.84**	**0.86**	**0.84**	**0.97**	**0.92**	0.62	**0.86**	**0.75**	**1**	0.79	0.65	0.73	**0.93**	0.70	**0.83**

Table 4.3: Quantitative results (class name indicates AUROC, AUC_{avg} of all classes) of models across MVTEC-AD [9] dataset.

Plant Village Dataset

The Plant Village dataset [11] is challenging due to the large intra-class variance present in this dataset. Leaves of a given plant can vary vastly in appearance with respect to shape and colour, meaning it is challenging to map the underlying distribution. Establishing meaningful features of healthy leaves which are invariant to the diverse geometry and colour differences of healthy leaves while still being able to detect anomalies present in diseased leaves as severe leaf discolouration and missing leaf parts make this task challenging. As a result, producing robust features which can leave healthy parts of the leaf unchanged while reconstructing severe deviations back into healthy parts would enable better success with this task.

With this in mind, it would be optimal for the noise generator of our network to somehow distort the geometry and discolour the images to severely obfuscate the leaf in the image. This would enable the denoising module to be robust against severe obfuscation while learning to reconstruct the underlying leaf geometry as it is supposed to be. This is the case and can be observed in Figure 4.4 where the noise attempts to heavily distort the geometry of the leaf (at the bottom left of the leaf) as well as adding a vivid tone of green to the image to attempt to heavily obfuscate the colour and geometry of the leaf. Our DAE network is then forced to reconstruct the original leaf geometry and colour from the obfuscated leaf which leads to increased robustness with reconstructing presented anomalous instances at inference.

The quantitative results of methods across this dataset are presented in Table 4.4. Our DAE + ALCN method obtains an AUC_{avg} of 0.77 which is the same as that of Skip-GANomaly [6]. Both methods surpass prior methods across this dataset.

Figure 4.5 illustrates the results of an input which is an out-of-distribution example through both the Skip-GANomaly [6] and DAE+ALCN into models trained on only another specific class singular (Figure 4.5, left label). The objective being that Skip-GANomaly [6] and DAE+ALCN should reconstruct out-of-distribution examples within the original class distribution. However, it can be seen in Figure 4.5 that Skip-GANomaly [6] successfully reconstructs an out-of-distribution exam-

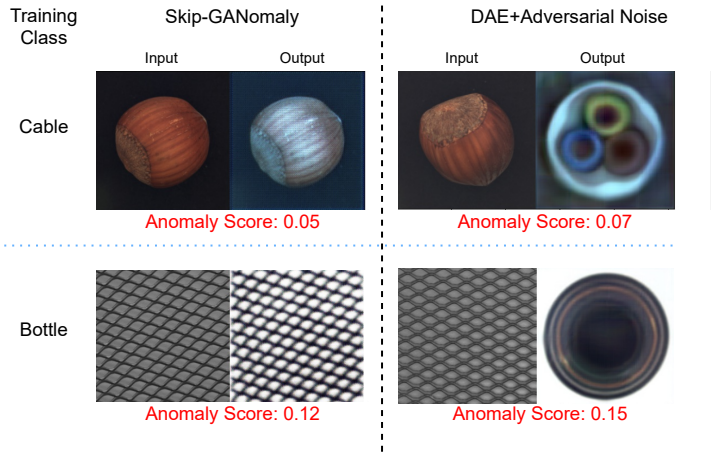

Figure 4.5: Comparison between Skip-GANomaly [6] and DAE+Adversarial Noise of feeding vastly out-of-distribution (Hazelnut and Grid) examples through models trained on a different class (Cable and Bottle). Anomaly Score in this figure is computed as the Mean Squared Error of the input and the output.

ple, giving weight to the conclusion that it has converged to a near pass-though identity function and copies information from input to output (i.e. hazelnut/grid observed in both input + output), despite the fact the model has never been exposed to these class examples during training. For Skip-GANomaly, this leads to low anomaly scores of 0.05 and 0.12 for Cable and Bottle respectively. By contrast, our DAE+ALCN architecture, manages to reconstruct such out-of-distribution examples back into the training classes thus resulting in the anomaly scores 0.07 for Cable (0.02 larger than Skip-GANomaly [6]) and 0.15 for Bottle (0.03 higher than Skip-GANomaly [6]). This shows that given vastly out-of-distribution examples, the DAE+ALCN network is more robust to misclassification and less prone to a pass-through identity-like reconstruction output.

We have also included the results of the individual classes of the Plant Village dataset in order to show the performance of the model on a more realistic scenario. The results of this are outlined in Table 4.5. The results show that the ALCN method is able to perform competitively to the PANDA approach by obtaining higher on the cherry class at 0.967 compared with the 0.927 exhibited by PANDA, however, falls short on all other classes. The resulting average AUC for this experiment resulted in 0.937 for PANDA and 0.903 for the ALCN approach.

Overall these experiments show that using our adversarial noise as a regularisation technique can enable even a simple architecture such as the Denoising Autoencoder outlined in Figure 4.2 to obtain competitive results than more complex model architectures.

Model	**Plant Village [11]**
	AUC_{avg}
AE [14]	0.65
AnoGAN [4]	0.65
EGBAD [5]	0.70
GANomaly [2]	0.73
Skip-GANomaly [6]	0.77
ALCN	**0.77**

Table 4.4: Quantitative results (AUC_{avg}) of models across Plant Village [11] dataset.

4.4.3 Model Complexity

An outline of model complexity together with inference time per batch is outlined in Table 4.6. The DAE+ALCN architecture has 9.87 Million parameters which is slightly larger than EGBAD [5] which is at 8.65 Million but still orders of magnitude smaller relative to that of AnoGAN [4]. The magnitude of our model comes from the noise generation module in addition to the DAE module required during training. Our DAE+ALCN architecture has an inference speed of 4 milliseconds per batch which is significantly faster than the other methods, but is trivially not faster than the sole DAE architecture.

Qualitative Results

Figure 4.7 shows the evolution of the adversarial noise produced by the Noise Generator over training across the Plant Village [11] dataset. It can be seen that the noise

Class	**Cherry**	**Corn**	**Grape**	**Potato**	**Strawberry**	**Tomato**	**AVG**
PANDA	0.927	0.989	0.986	0.96	0.986	0.773	0.937
ALCN	0.967	0.984	0.957	0.891	0.97	0.649	0.903

Table 4.5: Quantitative results of AUC across the Plant Village individual classes.

		Model				
		DAE	**AnoGAN**	**EGBAD**	**GANomaly**	**DAE+ALCN**
Parameters (Million)		1.12	233.04	8.65	3.86	9.87
Inference Time/Batch (Millisecond)	**MNIST [7]**	2.36	667	8.02	9.7	**4.54**
	CIFAR-10 [8]	2.73	611	9.55	10.53	**5.23**

Table 4.6: Comparison of model complexity (number of parameters (millions)) and inference time (milliseconds).

starts off as a green square with little to no resemblance of the plant data. As training continues, the noise that is generated becomes similar to the input distribution and bares visual similarity.

Figure 4.6 illustrates the qualitative results of DAE + ALCN across different datasets. The first column for each example shows the input images to the model. The second column illustrates the adversarial noise which is added to the input resulting in those images (3rd column). This adversarial noise + input is then fed into DAE and the resulting output after denoising (4th column); Of particular interest are the noise examples across the MNIST [7] and Plant Village [11] datasets in which the noise produced is perceptually very similar to the style/shape of the input data of the respective task. The adversarial noise of the digit 2 from the MNIST dataset especially appears perceptually like a digit 2. This gives light on the nature of the maximal obfuscation noise produced by the optimised noise generator such that it attempts to obscure the shape of the objects within the class. In Plant Village, the examples are the same colour of the leaves, but bear hardly any similarity to the geometry of the leaves themselves giving a different nature of noise that is used on this task. For the noise produced on the MVTEC and CIFAR-10 task, the noise produced appears random and not stylised to the task dataset so although the noise can be tailored to the input training set distribution, the noise produced may also produce randomised noise that bears no similarity.

Despite the severe perceptual obfuscation that can be seen in the Image+Noise column of Figure 4.6, the DAE component of the architecture is able to successfully reconstruct the original input images that resemble the original unperturbed input images as illustrated in the Output column of Figure 4.6.

4.5 Conclusion

The work outlined in this chapter presents a novel approach of regularisation by adding adversarially learned noise to input images while training a denoising autoencoder to perform reconstruction-based anomaly detection. Replacing the prior stochastic, individual pixel approaches of obfuscating images such as Gaussian and Random Speckle noise during the task of semi-supervised anomaly detection has been explored in recent literature. However, the work presented in this chapter resolves weaknesses within prior work.

Such weaknesses include constructing computationally expensive adversarial noise examples during training as performed in ARAE [31]. Our ALCN method by contrast, optimises the noise during training gradually on each iteration so that it remains dynamic and efficient. Due to the noise being produced by a Generative Adversarial Network (GAN) which is trained in conjunction with the denoising autoencoder, the noise produced between two consecutive iterations is updated smoothly while the α value in the weighted sum can vary the corruption of the noise subtly or severely at random during each step. The noise hence forces a semi-maximal feature distance in the DAE latent representation while the perceptual appearance of the image has changed negligibly (under a large α value). This is what the ARAE method obtains during training, but introduces strict requirements for such noise during each step in training, as previously stated.

Although the OLED [32] method produces noise which is dynamic at each step and inpaints the optimally important regions in the image with a mask, the mask is singular valued and discrete. Although this method achieves results better than the ARAE approach on the MNIST dataset, the performance is significantly lower than ARAE across the CIFAR-10 dataset. It could be that the MNIST task benefits from the uni-valued mask due to itself being singular valued. However, applying the same such noise on the CIFAR-10 dataset may not be optimal due to not taking into account the task-specific appearance of the input training data. The ALCN approach in this chapter combats this disadvantage by producing dynamic noise

tailored to the distribution of the input data. It can be seen in Figure 4.6 under the MNIST row(s) that our ALCN approach produces noise which is perceptually similar to the noise that OLED [32], giving further evidence that perhaps the noise that OLED produces is optimal for the MNIST task, but not the CIFAR-10 task. This is further evidenced in the results presented in Table 4.2 where OLED achieves 0.99 across MNIST and 0.67 across CIFAR-10 whereas the ALCN approach achieves 0.99 (similar to OLED) across MNIST and 0.74 (surpassing the result of OLED) across CIFAR-10. As such, a *one-size-fits-all* noising approach such as OLED will produce noise which is not optimal to all tasks which ALCN combats.

The study presented in this chapter provides evaluation across numerous datasets ranging from trivial and unrealistic leave-one-out anomaly detection across MNIST and CIFAR-10 achieving results which surpass prior work [2, 4, 5, 36] with vastly more complex architectures and parameter counts as well as against the DAE architecture with both Gaussian and Random Speckle noise. Achieving the best results across 90% and 60% of the classes across both MNIST and CIFAR-10 respectively across protocol 1 and AUC_{avg} values of 0.89 and 0.67 across the respective tasks. As previously mentioned, we surpass the performance of prior methods in denoising approaches [31–35] applied to reconstruction-based anomaly detection across protocol 2 of MNIST and CIFAR-10 achieving AUC_{avg} values of 0.99 and 0.74 respectively.

Across real-world datasets, MVTEC-AD and Plant Village the 256×256 resolution DAE+ALCN method achieved the best result in 66.6% of classes in MVTEC, but achieved the best AUC_{avg} score of 0.83 surpassing the next best performing approach of Skip-GANomaly at 0.80 in the same metric. Across the Plant Village dataset, DAE+ALCN achieved an AUC of 0.77 which is on-par with the result obtained by Skip-GANomaly.

This chapter presents the DAE+ALCN approach, a relatively simple yet effective method of regularisation applied to a simple off-the-shelf denoising autoencoder model which uses a GAN-like module which produces tailored continuous adversarial noise at each iteration during training. When this noise is added to the input, the DAE has to not only be able to reconstruct the input features, but also fix by re-

construction, corrupted input parts obfuscated by the noise. The analysis produced in the results section show that this is an effective method to boost performance of even a simple method to bypass performance of more complex architectures.

4.6 Limitations and Further Improvements

Although the noise produced by the DAE+ALCN approach improves performance of a simple denoising autoencoder, the noise affects the entire input images and the user has very little control over this currently. It could be advantageous to only select the most important parts of the image for noise to be applied to or limit the noise to few regions to further tailor the noise to the distribution of the types of anomalies likely to present to the model at inference; for example, if examples are likely to be small, then the threshold could be set to high so that noise is only applied to maximally important and thus more focused regions.

This could be performed by a thresholding approach applied to the output of the noise generator module similar to that of the step-wise thresholding function within OLED [32]. The issue with thresholding this way would be that low-valued noise applied to important regions could be thresholded by the function and as such would not be applied to the respective input. A method of attention-guided noise would tackle this problem; the generated attention map would be thresholded instead of the noise and then this thresholded attention map could be directly multiplied by the adversarial noise. This would allow maximally important regions of the input to be maximally obfuscated by the noise at each step.

The approach by Adey et al [103] utilises an effective approach of training a denoising module to reconstruct intentionally corrupted image patches I_n using a filter bank of noising approaches. Instead of the DAE reconstructing the image, however, it is optimised to produce the output A, which is an anomaly map which can be thought of as the pixel-shifts that are required to repair the noise. The reconstruction $I' = A + I_n$ is compared to the unperturbed input I. This approach achieved superior results across the MVTEC-AD textures dataset as it is far more efficient to

produce the inverse of the noise rather than the entire image during reconstruction. With this in mind, the user-defined filter bank of noising approaches as used in the work by Adey could be replaced by the ALCN noise generator approach in this work. In essence, the denoising module would have to 'repair' the obfuscation caused by the noise produced by the ALCN noise generator module by the DAE producing the direct inverse of the noise. This would directly connect both the DAE and the ALCN to truly adversarially train them together using Algorithm 1.

A disadvantage of the DAE+ALCN method is that it adds a significant compute and space overhead to current models during training due to it containing a secondary network (Noise Generator) to be trained jointly with the autoencoder.

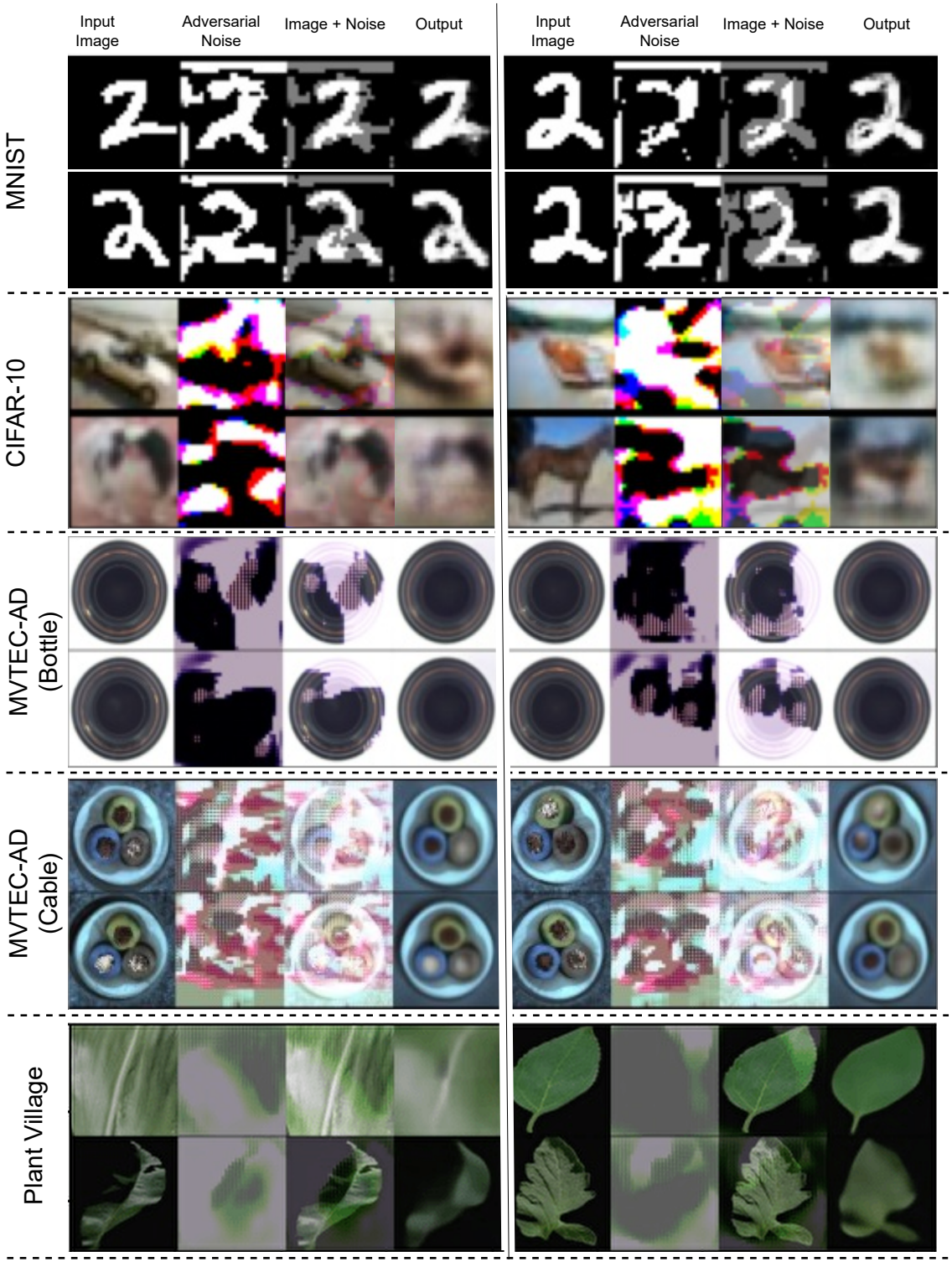

Figure 4.6: Examples of generated adversarial noise together with the output after denoising this noise.

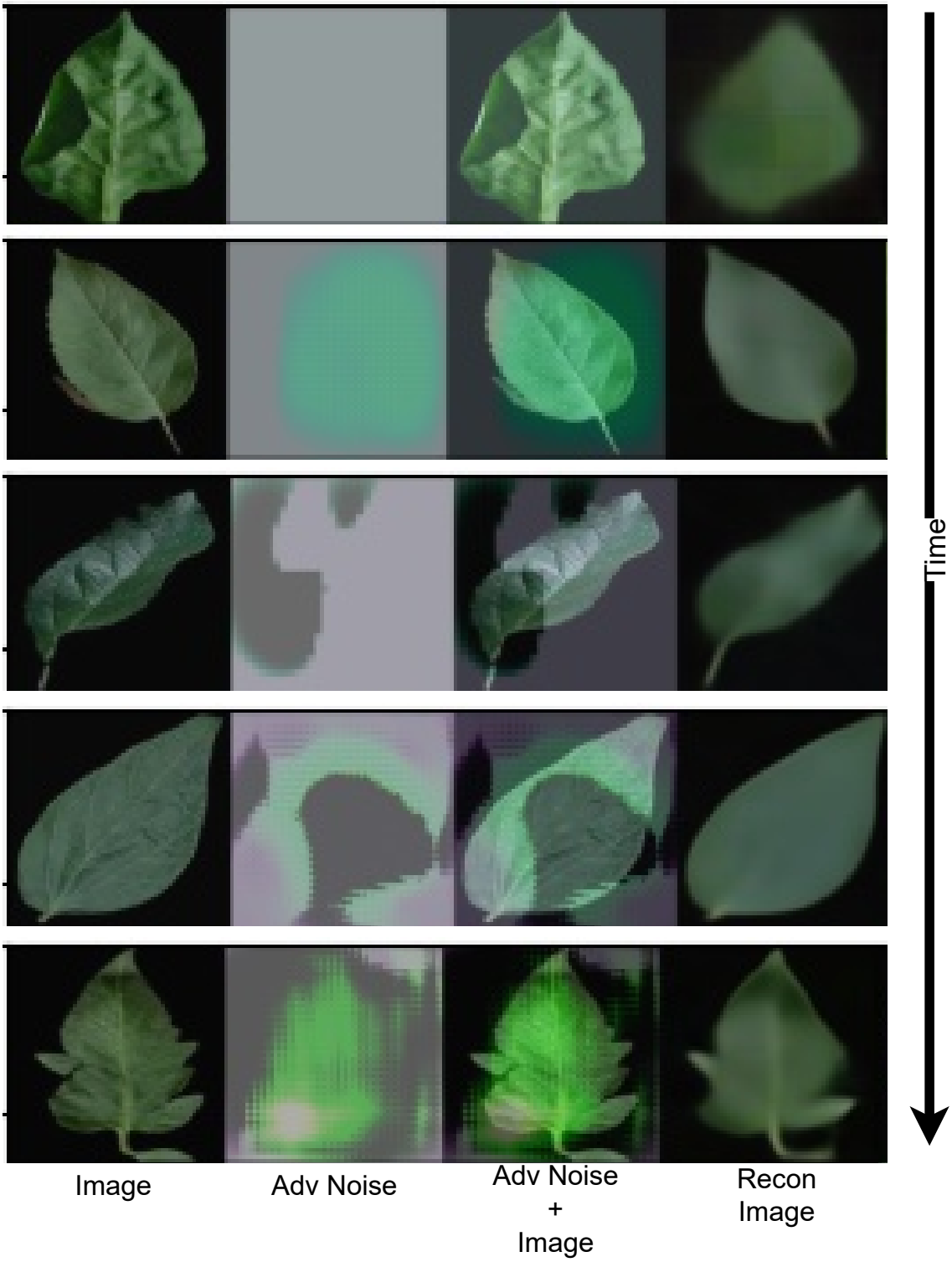

Figure 4.7: Demonstration of how the adversarial noise evolves during training across the Plant Village [11] dataset.

CHAPTER 5

Semi-Supervised Surface Anomaly Detection of Composite Wind Turbine Blades From Drone Imagery

5.1 Introduction

Global energy demand is increasing significantly. Between 1971 to 2010, demand for energy increased 2.4 fold (+134%) and is predicted to increase by +204% by the year 2030 [207]. The '1992 - Kyoto Protocol', introduced by the United Nations Framework Convention on Climate Change (UNFCCC), entered into force in 2005. The Kyoto Protocol regulates 192 member countries to limit and reduce Greenhouse Gas (GHG) emissions in line with agreed individual targets. Fundamental to reducing GHG emissions is to transition from fossil fuels such as coal, oil, and natural gas to renewable sources of energy such as nuclear, wind, solar and tidal to name only a few. Renewable energy sources emit negligible CO_2 emissions and can supply for the increase in demand for power.

The Global Wind Energy Council (GWEC) estimates a 17-fold increase in wind power generation, providing as much as $25-30\%$ of global electricity by the year 2050 [208], equating to 123 petawatt-hours (PWh) of electricity annually [209]. Unlike the reliability of fossil fuel-based energy sources to produce energy on demand, however, wind energy is temperamental. Low wind speeds do not provide sufficient lift forces for turbine blades to rotate whereas high wind speeds exceeding $> 25m/s$ $(90km/h)$, commonly force many modern turbines to shut down as a safety measure [210].

Few locations provide reliable and sufficient supply of wind to meet energy demands. Offshore wind farms are now favoured due to factors which include: the availability of large continuous areas suitable to major projects, and the reduction of visual or noise impact. This promotes construction of broad, widespread wind farms featuring multitudinous, larger turbines at offshore sites which generate significantly more power than their smaller, onshore counterparts. An example as to the scale of modern offshore wind farms is the Hornsea 1 wind farm which contains 174 turbines spread across an area of 407km^2. Due to exhaustive usage and weather-related degradation, such turbine installations including the exposed turbine blades must be routinely inspected for damage. A common cause of failure is turbine blade damage such as: erosion, kinetic foreign object collision, lightning or other weather

related phenomenon, and delamination to name only a few.

Wind Turbine Blades are typically made from fibre-reinforced composites due to such materials exhibiting heterogeneous [211] and anisotropic properties [212]. Typically they are constructed from Glass Fibre Reinforced Plastic (GFRP) materials [211]. GFRP offers the material properties of being both strong (able to withstand an applied stress without failure), and ductile (able to stretch without snapping). These properties are desirable for wind turbine blades due to the strain of operational forces (constant torque forces from lift and rotation) as well as natural forces from weather fronts and foreign object collision during operation. Over time, these forces can cause damage to the blades which may require a turbine to halt operation for a period of time, or even necessitate operational cessation of the turbine, which are both costly. This is why they must be routinely and regularly inspected in order to prevent such events [213]. In the example of the Hornsea 1 farm, each turbine on the farm has 3 blades equating to 522 total blades each with an approximate surface area of 600 m^2. Due to the sheer area, quantity, and size of turbines in new offshore wind farms, engineers and inspectors experience tremendous challenges in inspecting turbine blades for damage to prevent subsequent costly failures.

Within commercial wind energy generation, the monitoring and predictive maintenance of wind turbine blades in-situ is a crucial task, for which remote monitoring via aerial survey from an Unmanned Aerial Vehicle (UAV) is commonplace [12]. Turbine blades are susceptible to both operational and weather-based damage over time, reducing the output energy efficiency of turbines. In this study, we address automating the otherwise time-consuming task of both blade detection and extraction, together with visual surface fault detection within UAV-captured turbine blade inspection imagery. We propose BladeNet, an application-based, robust dual architecture to perform both unsupervised turbine blade detection and extraction, followed by super-pixel generation using the Simple Linear Iterative Clustering (SLIC) method to produce regional clusters of visually similar parts. These clusters are then processed by a suite of semi-supervised detection methods to detect visual sur-

face anomalies on the blade. Our dual architecture detects surface faults of glass fibre composite material blades with high aptitude while requiring minimal prior manual image annotation. BladeNet produces an Average Precision (AP) of 0.995 across our Ørsted blade inspection dataset for offshore wind turbines and 0.223 across the Danish Technical University (DTU) NordTank turbine blade inspection dataset. BladeNet also obtains an AUC of 0.639 for surface anomaly detection across the Ørsted blade inspection dataset. We compare our segmentation-driven U-Net [30] approach to common-place object detection methods: Mask RCNN [18], YOLACT [19] and Cascade Mask RCNN [20]. We note in our findings that these prior methods do not produce segmentation masks which tightly bind to the edges of the turbine blade, opting instead to oscillate and miss important details such as the vortex generators and blade tip. As the edges of the blades are more susceptible to damage, it is vital that they are included in the detection. BladeNet is able to capture a tight fit to the Ørsted turbine blades which leads to successful detection of surface abnormalities in the second stage of the architecture.

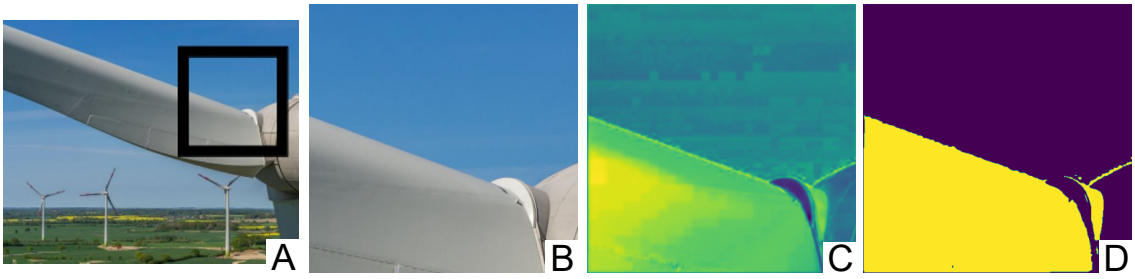

Figure 5.1: Transfer detection of an out-of-dataset turbine blade illustrating the robust ability of our method A) Image of wind turbine with marked region on the blade and nacelle, B) Cropped region of turbine blade, C) Raw model output, D) Threshold model output producing final blade detection.

5.2 Approach

Our approach is two-stage with the operations of: 1) Blade detection and extraction and 2) Semi-supervised surface anomaly detection. The first stage, The BladeNet U-Net detection pipeline is outlined in Figure 5.2 and performs blade detection and

extraction (Section 5.2.1) to solely obtain turbine blade parts from a given image. Figure 5.1 outlines this detection process step by step starting from a given image through to obtaining a mask solely containing blade parts in the data. Extracted blades are then subsequently processed with the Simple Linear Iterative Clustering (SLIC) [16] method (Section 5.2.2) which is illustrated in Figure 5.7 to generate super-pixel clusters which are used as input to the second stage which performs semi-supervised anomaly detection via a suite of well-established approaches to detect visible surface anomalies present on the blade (Section 5.2.3).

5.2.1 Blade Detection and Extraction

Accurate detection and extraction of turbine blades in any given image is crucial to the success of the semi-supervised anomaly detection approaches downstream [2, 6, 13, 124] (Section 5.2.3). If background or any such artifact in the image is introduced, then it could corrupt the learned representations over normal blade data obtained by the semi-supervised methods. Example cases are:

- If the predicted mask is too big and includes the sky around the blade, if there are artifacts in the sky around the turbine blade such as birds or aircraft, then the images could be flagged as anomalous even though the blade itself is healthy, thus leading to false positives during inference.

- The opposite implies that parts of a given blade are missed from the extraction and not included within the normal class training data. This would mean that the semi-supervised anomaly detection methods may not have exposure to adequate amounts of certain blade parts leading to missing representations during training. Such missing blade parts would also miss being classified during inference time leading to ambiguity as to the health of the blade part.

When detecting large objects such as turbine blades in high-resolution (6720 $\times$ 4480) drone imagery, conventional instance segmentation models [18–20] output masks which appear wavy when placed over the object in the original image. This is outlined qualitatively in Figures 5.11 and 5.10. The masks of Mask R-CNN [18],

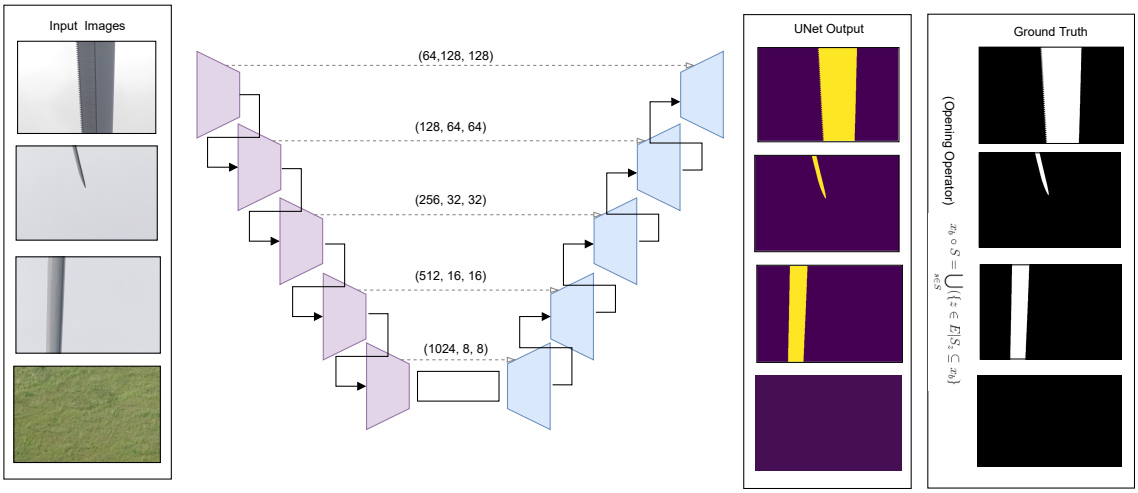

Figure 5.2: Outline of the BladeNet UNet segmentation module which returns the instance segmentation mask of blades in the input images to match the opening operator.

YOLACT [19] and Cascade Mask R-CNN [18] all exhibit oscillating detection boundaries around the straight edges of the blades as well as fail to capture important sections of the blade such as the tip and triangular edges of the blades which are more prone to damage and as such have more potential to feature anomalies. This oscillation of the segmentation masks is due to the resizing of the predicted mask which is 15×15 for (Cascade) Mask R-CNN and 138×138 for YOLACT up to the full resolution which exacerbates the loose fit of the mask boundary due to the exaggeration of edges in the small mask. Detection methods also use discrete polygon annotations for objects which under-sample the true outline curves of an object which can fail to capture them with enough precision.

Our approach extracts turbine blade parts from a given image and discards background and unwanted artifacts by utilising a Fully Convolutional (FCN) U-Net [30] architecture for one-class instance segmentation. This architecture is outlined in Figure 5.2. Five convolutional encoders are used to encode images to a latent representation of shape $1024 \times 8 \times 8$. This latent representation is then decoded using Five convolutional transpose layers connected in series as well as residual connections from their encoder counterparts. The output is a 1-channel mask outlining a pixel-

wise segmentation of where a blade is present in a given image. This process is illustrated in Figure 5.1. Firstly, fixed image patches are taken from the original image (Figure 5.1: A and B). These patches are then used as input into the U-Net module to produce a raw pixel-wise mask outlining maximal activation on blade parts (Figure 5.1: C). A threshold value is then applied to this raw output, producing a clean and denoised segmentation mask (Figure 5.1: D) of turbine blade parts in the original patch.

To create 'pseudo ground truth' for our model, we utilise morphological operators and negative example sampling. Using our Ørsted turbine blade inspection dataset X_b; for each $x_b \in X_b$ where $x_b \in \mathbb{R}^{B\times 3\times H\times W}$, the Opening Morphology Operator $x_b \circ S = \bigcup_{s\in S}(\{z \in E | S_z \subseteq x_b\})$ as a combination of erosion $x_b \ominus S$ followed by dilation $x_b \oplus S$ provides pseudo ground truth for $\forall X_b$ which closely approximates the true edges of the wind turbine blades in X_b. Figure 5.3 illustrates successful pseudo-ground truth segmentation masks which closely match the Ørsted turbine blades in the given imagery.

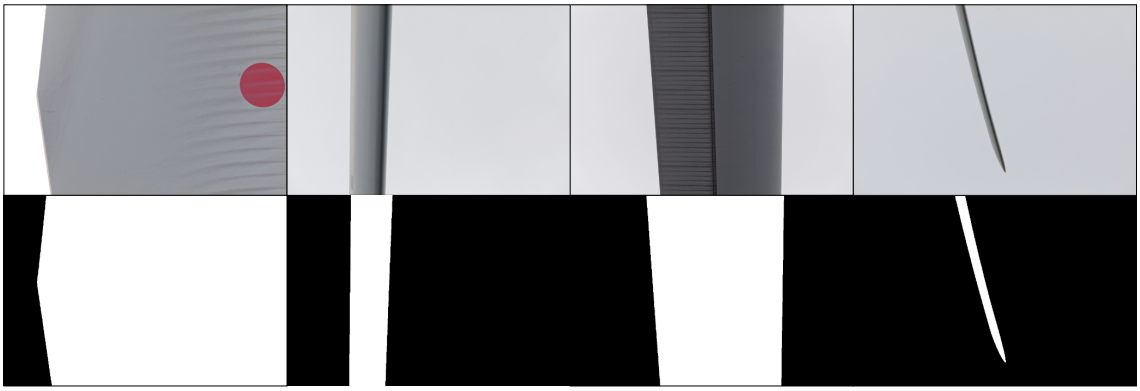

Figure 5.3: **Top Row:** Original images from Ørsted turbine blade dataset. **Bottom Row:** Pseudo-ground truth after applying Opening operator on the original images in the top row. Note that the red circle on the turbine blade on the left-most image of the turbine blades (above), is considered normal by the Ørsted and these circles are put on the turbine blades on purpose, the authors have not included the circle on this image.

This method of generating ground truth offers a fast and efficient way to segment the turbine blades. However, the Opening operator is fragile and very sensitive to

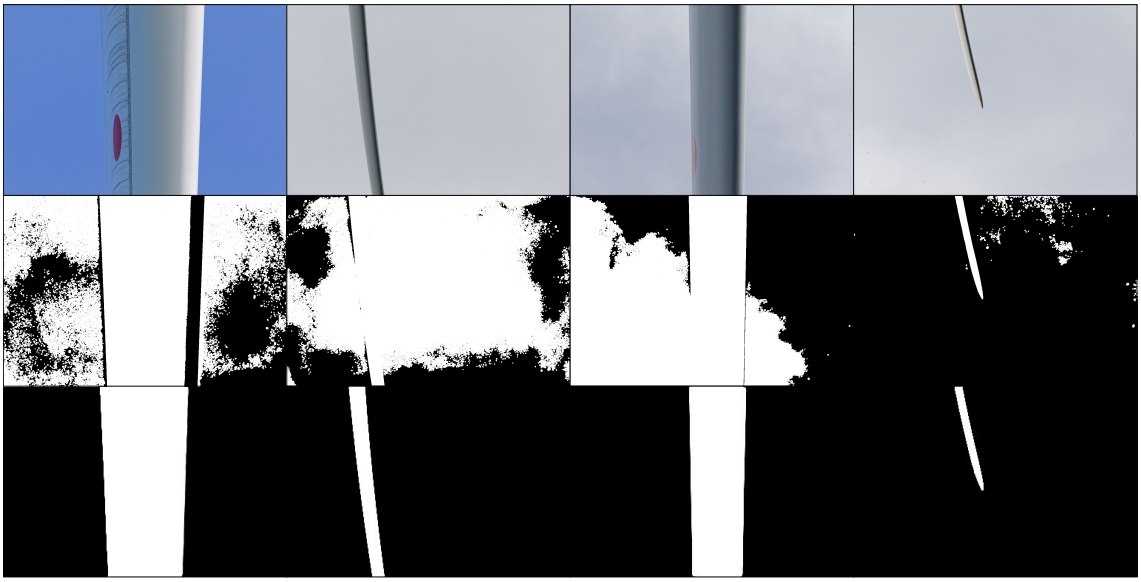

Figure 5.4: **Top Row:** Original images from Ørsted turbine blade dataset. **Middle Row:** Incorrect pseudo-ground truth after applying Opening operator on the original images in the top row. **Bottom Row:** Correct segmentation of turbine blades via the trained U-Net detection module of BladeNet.

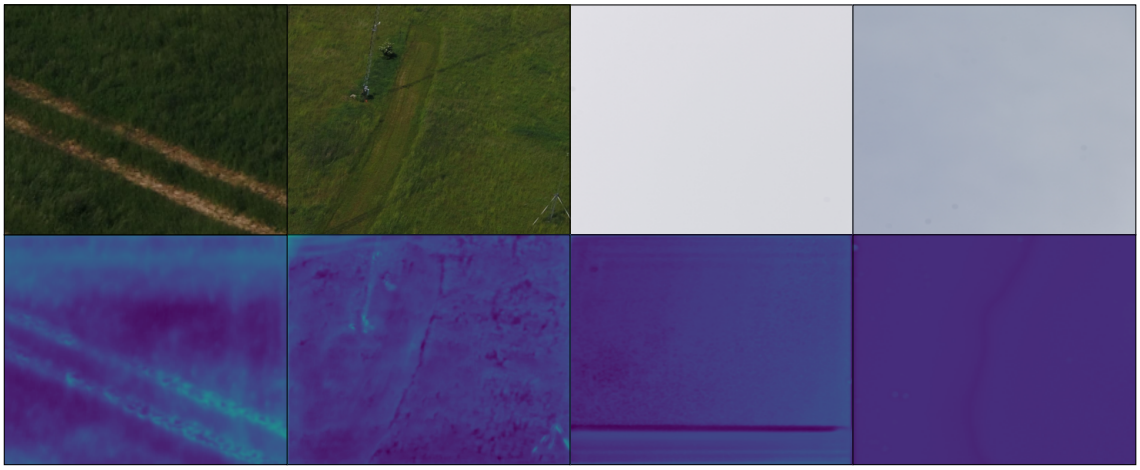

Figure 5.5: **Top Row:** Images from the NordTank turbine blade dataset which do not feature turbine blade parts (negative samples). **Bottom Row:** Raw output from BladeNet showing null detection of the image above.

changes in the images. Due to the fragility of the operator, the pseudo ground-truth must be manually screened ahead of training to remove unsuccessful cases from the training set. Figure 5.4 illustrates failure cases of the operator which must be removed prior to training as they feature high levels of noise in the detection and

missing blade parts. The fully trained U-Net segmentation model trained on clean ground truth obtains clean detection of these failure cases, however.

Negative class examples $x_n \notin X_b$ consisting of images of sky and ground are introduced during training with a ground truth tensor of zeros of shape $\mathbb{R}^{B\times 3\times H\times W}$, indicative of no blade presence in the image. Figure 5.5 shows the negative sampling of both ground and sky images together with their detection output. By performing this, BladeNet learns what it must pay attention to, and ignore in a given image and also makes the model more robust to background artifacts.

5.2.2 Superpixel Extraction

In this work, we implement Simple Linear Iterative Clustering (SLIC) [16] for generating sub-region patches of the full blade rather than using conventional sliding window patches.

Approximately n clusters of neighbouring pixels are generated by stepping over an image of resolution $N = X \times Y$ with an interval $I = |\sqrt{\frac{N}{|n|}}|$ and taking a set of $|n|$ centre points $C = \forall n \in I, \{x_n, y_n\}$. Each centre $c_n \in C$ is refined by taking the best matching pixels from the neighbourhood of $2S^2 < X \times Y | S \in \mathbb{N}$ surrounding pixels utilising euclidean distance upon both the pixel colour vector $(L \times a \times b)$ and the pixel coordinates as: $D_s = \sqrt{(l_n - l_i)^2 + (a_n - a_i)^2 + (b_n - b_i)^2} + \frac{m}{S}\sqrt{(x_n - x_i)^2 + (y_n + y_i)^2}$ where m is the spatial proximity factor of the method.

SLIC patches contain pixels which share visual characteristics to other pixels belonging to the same super-pixel. Super-pixels increase the likelihood that an anomalous region in the image, or key region of interest for a given blade will not likely be situated across the edge of two neighbouring patches. If an anomalous region is split across two patches, then it not only decreases the size of region by the size of the overlap, but the edge of the patch restricts the features of the area surrounding the anomalous region to only the edge of the image hence the model will not be fully utilising the spatial information of the anomalous region.

The patches generated by SLIC are likely to contain the full defect, or no defects at all due to the technique of clustering together similar neigbourhood values of

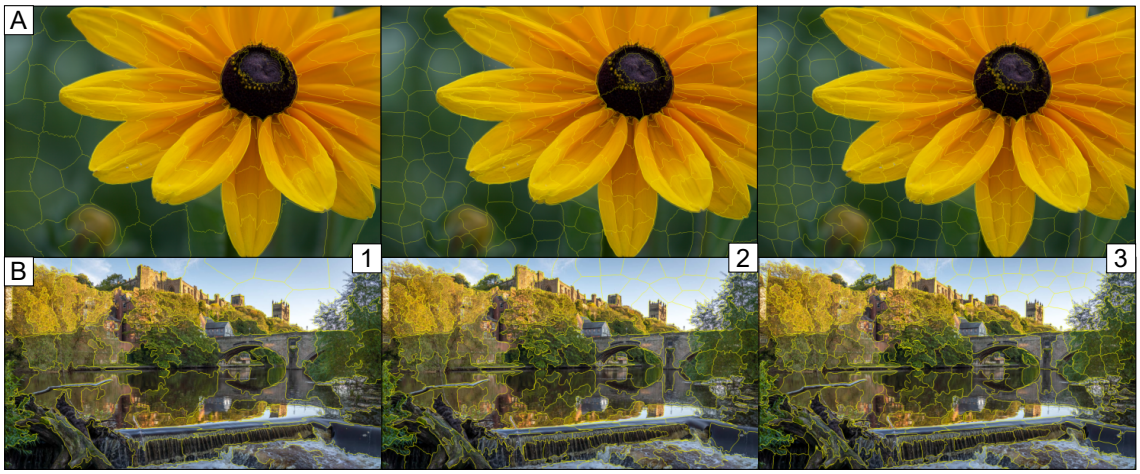

Figure 5.6: Example of Simple Linear Iterative Clustering (SLIC) [16] Superpixel Segmentation across flower (**A**) and Durham (**B**) using σ=5 with number of segments as 100, 200, and 300 (**1, 2, 3**)

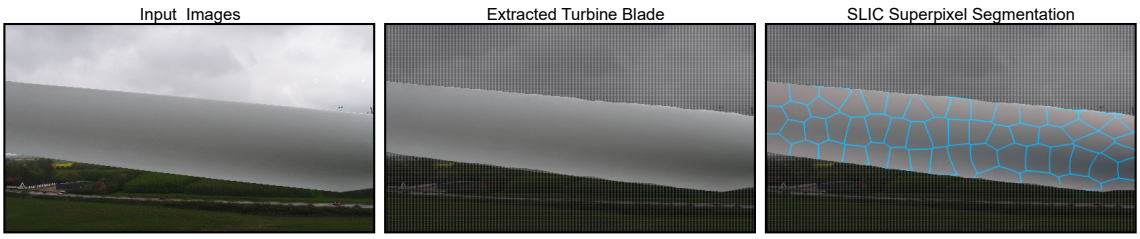

Figure 5.7: **left:** original input image from the NordTank dataset. **centre:** The extracted blade parts after detection and instance segmentation. **right:** SLIC superpixel regions with sigma=5 and number of segments set to 100 across the extracted blade.

pixels. This is illustrated in Figure 5.13. This makes it easy to sift through the images, and quickly decide which are normal and which contain defects. This task would be a bit more difficult to perform with the sliding window approach due to the fact that the anomalous region could be on the boundary between two neighbouring windows, minimising the size of the anomalous region by half and making it harder for the model to detect such anomalies.

5.2.3 Anomaly Detection

Semi-supervised anomaly detection is performed by using super-pixel regions which have no visible defects present to train well-established anomaly detection models. These work by generatively mapping normal images to a latent representation such that when a visual defect presents itself, the representation will differ from normality and the presented example will be flagged as anomalous.

In this work, we utilise the self-supervised anomaly detection algorithms AnoGAN, GANomaly, Skip-GANomaly and PANDA [2, 4, 6, 13] to provide benchmark performance across this task of detecting surface faults in composite blade materials due to their proven success across prior anomaly detection tasks. We propose the novel semi-supervised anomaly detection approach of U-GANomaly outlined in Section 5.2.4 which outperforms the benchmark performance across this task.

5.2.4 U-GANomaly

The introduction of residual (skip) connections to the generator of GANomaly [2] produces the Skip-GANomaly architecture. Identical to the GANomaly architecture with the exception of extending the Generator module into a U-Net [30] with the use of residually combining information from the prior encoders with the respective decoders. This ensures that early low-level features in the encoding process are carried forward while decoding which better models normal representations. The limitation of Skip-GANomaly, however, is the use of the discriminator from the Deep Convolutional Generative Adversarial Networks (DCGAN) method [125]. As this is a novel classifier-based approach, it suffers from learning a representation that is able to penalise the Generator based solely on the most discriminative differences between the real and synthetic data [17]. As such the discriminator focuses on either global structure or local details [17]. This limitation is tackled in the work by Schonfeld *et al.* [17] by introducing a decoder module to the discriminator architecture which decodes the representation prior to categorisation into a 1 channel map outlining a pixel-wise discrimination score map for a given input image. This resulting map

produces a continuous per-pixel discrimination score which is used in conjunction with the classification after encoding to feedback more information to the Generator during training. By introducing this U-Net discriminator to Skip-GANomaly, we create a fully residual, double U-Net network for semi-supervised anomaly detection. Our experiments show that this approach outperforms all benchmarks produced by prior methods. The anatomy of this approach is outlined in Figure 5.8. We choose to implement this approach into the GANomaly [2] because, this architecture has been proven to be effective in the literature on a number of tasks and the repository is easy enough and robust enough to implement such changes to accommodate for this new architecture.

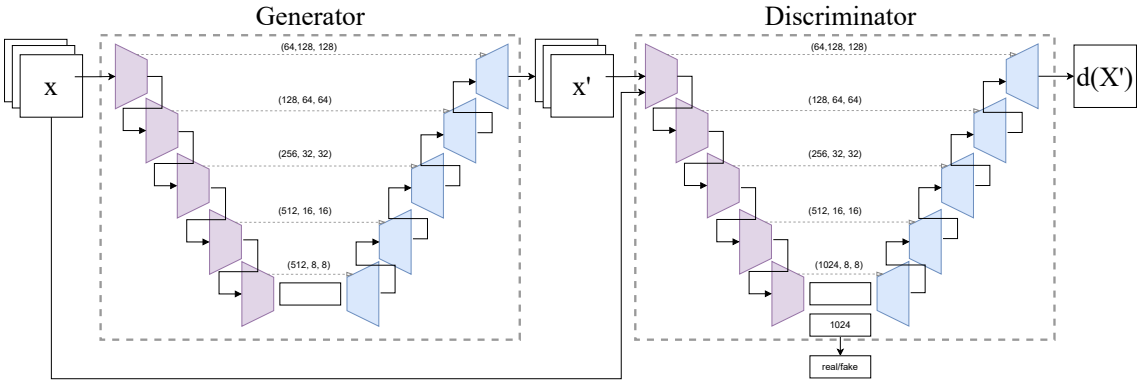

Figure 5.8: Overview of the U-GANomaly architecture featuring the dual U-Net architecture featuring the Generator module from Skip-GANomaly [6] and U-Net Discriminator module from [17].

5.3 Experimental Setup

We evaluate the performance of the BladeNet architecture by individually comparing each component. We start with evaluating the capability of the blade detection and extraction (Section 5.4.1) component and then compare the anomaly detection methods across the extracted blades to detect anomalous regions on the blade surfaces (Section 5.4.2).

The two datasets used in this chapter are the Ørsted turbine blade inspection

dataset and the DTU NordTank blade inspection dataset. The Ørsted turbine blade inspection dataset consists of drone inspection imagery of offshore wind turbine blades from the Hornsea 1 wind farm. It contains 3941 images of offshore turbine blades from varying perspectives in differing weather and backdrop with resolution 6720×4480. The DTU NordTank dataset is supplied by [12] and contains 701 high resolution (5280×2970) images captured by a drone of "Nordtank" wind turbines located at the DTU Wind Energy onshore test site at Roskilde, Denmark.

We evaluate BladeNet against established benchmark methods. We train our detection method solely across the Ørsted turbine blade inspection dataset with the pseudo-ground truth together with negative image samples. In total, 1310 images were rejected due to having poor ground-truth (Figure 5.4) leaving us with 2631 Ørsted images (33.2 % reduction) for training. We use a 20:80 split for testing and training respectively. After training, we report the performance after inferring across both the Ørsted dataset and DTU NordTank dataset separately using the same learned model parameters to demonstrate the robustness of the BladeNet detection approach.

All training was performed on a Titan X GPU. The hyper-parameters of the experiment are taken from the original work by [18–20]. Binary Cross Entropy (BCE) with logits loss with a learning rate of 0.001 was utilised for the U-Net blade detector along with RMS Prop optimiser with weight decay of $1e^{-8}$ and momentum of 0.9. Image scaling by 0.2 was also performed to preserve memory usage with a batch size of 10. Data augmentation is performed via rotation (degrees 90, 180, 270), flipping with probability 0.5, and random crop. The hyper-parameters of prior anomaly detection methods are taken from the original work by [2, 4, 6, 13, 14]. The Generator module of U-GANomaly uses the same hyper-parameters as Skip-GANomaly [6]. The hyper-parameters of the Discriminator module are also taken from the original Skip-GANomaly [6] implementation but, an additional BCE loss term is implemented upon the decoded output produced by the U-Net Discriminator [17, 30]. The Adam optimiser with learning rate of $2e^{-}4$ and epsilon of $1e^{-8}$ was used across a batch size of 64 during training.

5.4 Evaluation

We gauge the performance of our method in two parts; first, by evaluating the blade detection and extraction capability (Section 5.4.1) and then assessing the efficacy of semi-supervised anomaly detection (Section 5.4.2) across the extracted blade parts.

5.4.1 Blade Detection and Extraction

The quantitative performance outlined in Table 5.1 shows that Mask R-CNN performed equally in Average Precision (AP) with YOLACT at 0.983 across the Ørsted dataset. However, YOLACT obtained a greater AP value of 0.023 on the transfer to the DTU NordTank dataset. Cascade Mask R-CNN surpassed the performance of YOLACT across the Ørsted dataset and achieved the best time efficiency of 520.12 ms of all models in the study, but performs worse than Mask R-CNN across the DTU NordTank dataset with AP of 0.002. Our method, BladeNet performs the best quantitatively, obtaining an AP of 0.995, 0.1 higher than the next best performing (Cascade Mask R-CNN) and an AP of 0.223 on the transfer DTU NordTank dataset, far out-performing all prior methods. Although it outperforms however, this does not in any way deem the method suitable for real-world application, only that, compared with the other metrics in this chapter, the bladenet approach performs better, as illustrated by the detections in Figures 5.10 and 5.11.

BladeNet produces clean and sharp masks which fit the blades closely and manage to detect the sharp triangular parts of the mid-body blade and the blade tip with high precision. These masks can be seen in Figures 5.12 where clean segmentation masks are produced which fit closely to the edges of the blades additionally, Figures 5.11 and 5.10 show that when zooming in on the edge of the mask predictions, BladeNet remains tight with the true edge of the blade, missing only slight parts on the inside of the blade; As the edges are accurately captured, they can be used to extract the blade instead of the multiplication operation and still include these missing parts in the extracted blade whereas other methods such as Mask R-CNN and Cascade Mask R-CNN fit the turbine blades poorly, exhibiting waving

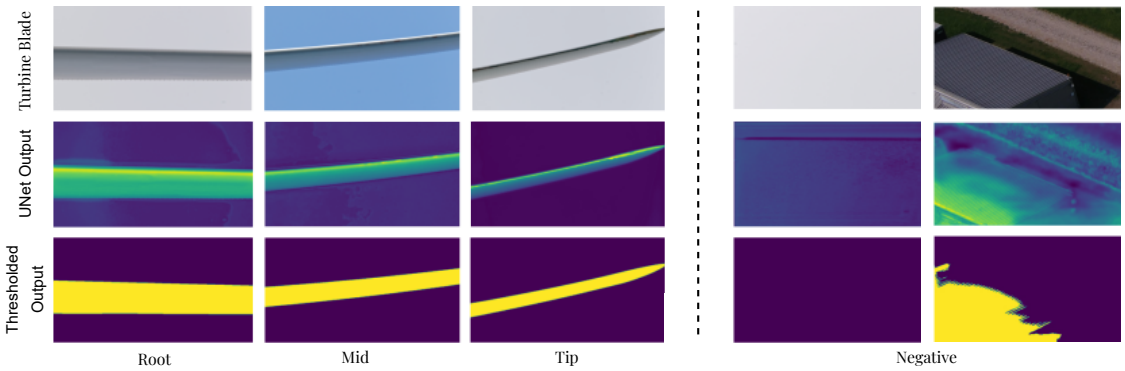

Figure 5.9: BladeNet output of turbine blade detection using inference of U-Net semantic segmentation module trained on data obtained from Ørsted turbine blade inspection showing both blade detections (left), and negative images containing no turbine blades (right).

segmentation mask predictions over the same images which miss out important sections of the blade edge and tip which are more prone to anomalies (edge erosion). Using these methods would impose null-categorisation of such parts of the blade in the next stage and hence impose false-negative error due to anomalous regions going undetected. Figure 5.12 further shows the capability of BladeNet at detecting numerous Ørsted turbine blade parts from different poses and angles with high accuracy.

In Figure 5.9, detection across both the Ørsted and DTU NordTank dataset is illustrated. BladeNet is able to detect the blades from the Ørsted dataset with high-aptitude, but the detections across DTU Nordtank have noisier detections. It is interesting that for the negative sample on the DTU NordTank dataset, BladeNet mistakenly predicts that the metal corrugated roof of the building is a turbine blade due to the colour and straight edges of the roof, resembling that of a turbine blade.

Table 5.1: Average precision (AP) at IoU = 0.5, number of parameters in Millions.

		Ørsted Dataset		Ørsted Model → DTU NordTank Dataset	
	Params	**AP**	**Time (ms)**	**AP**	**Time(ms)**
Mask R-CNN	43.9	0.983	590.36	0.005	537.31
YOLACT	34.7	0.983	549.06	0.023	478.04
Cascade Mask R-CNN	77	0.985	**520.12**	0.002	**314.61**
BladeNet	**17.3**	**0.995**	3439.21	**0.223**	1791.43

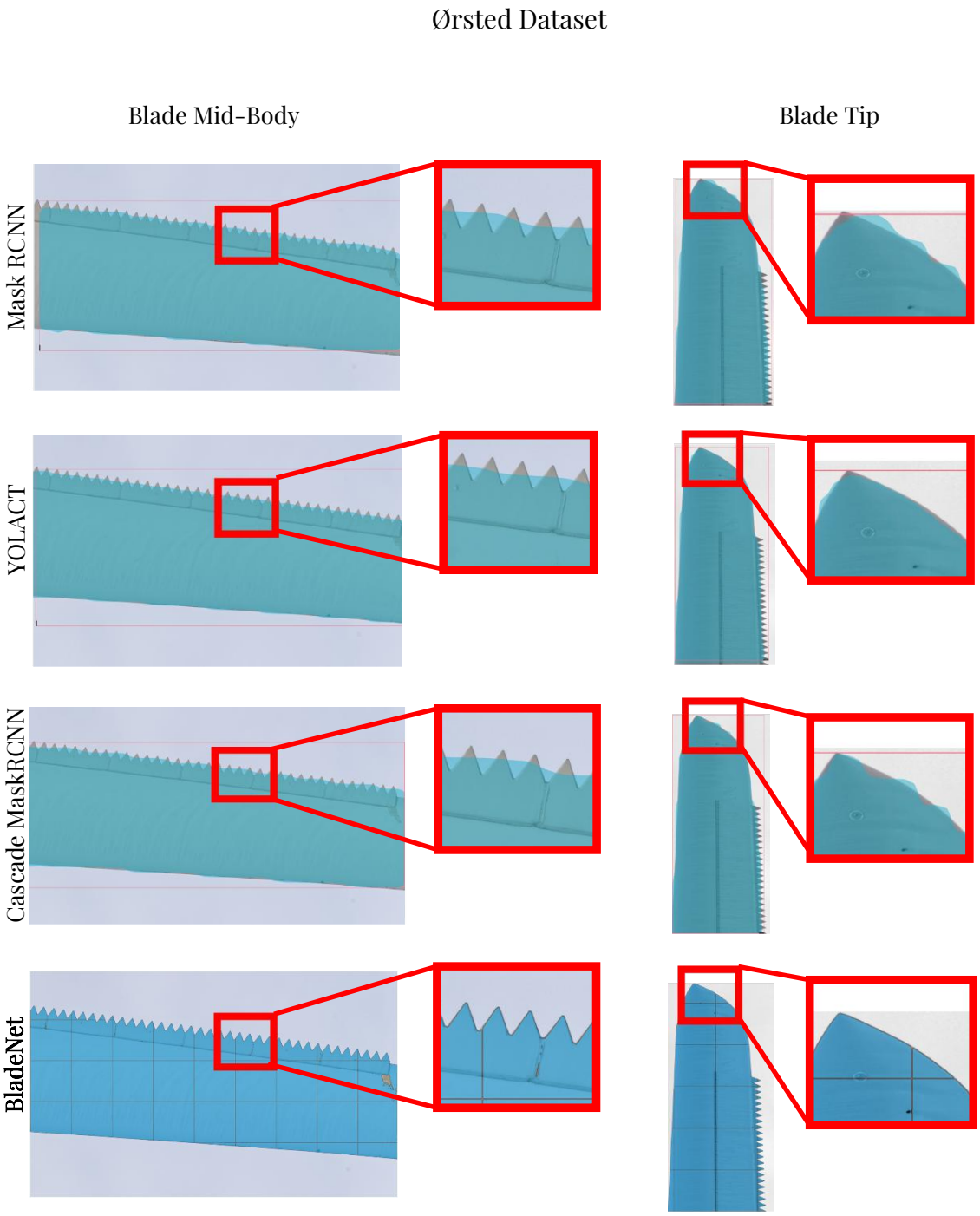

Figure 5.10: Instance segmentation mask quality comparison across the Ørsted Drone Inspection Dataset between Mask R-CNN [18], YOLACT [19], Cascade Mask R-CNN [20] and BladeNet.

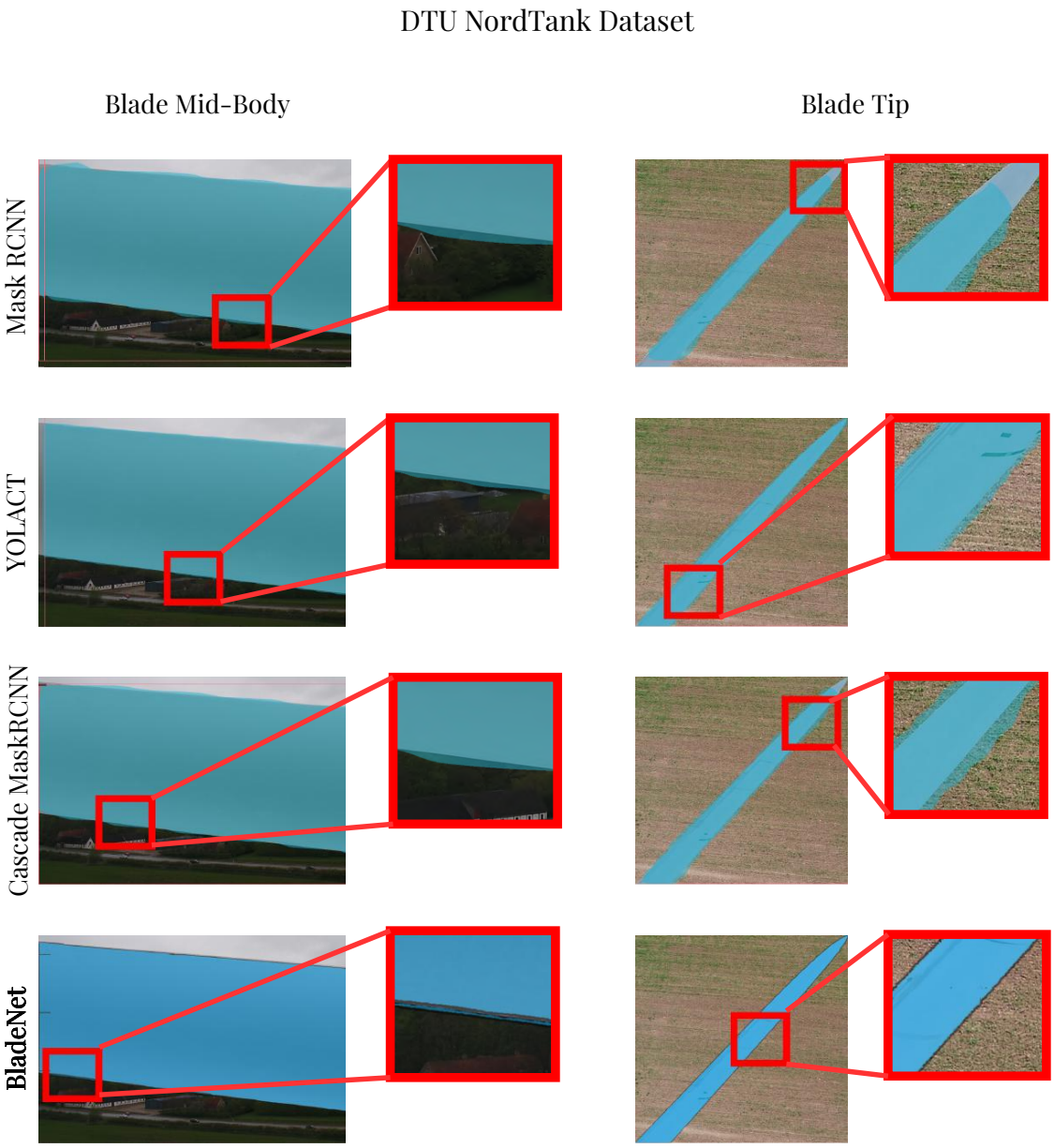

Figure 5.11: Instance segmentation mask quality comparison across the DTU Blade Inspection Dataset between Mask R-CNN [18], YOLACT [19], Cascade Mask R-CNN [20] and BladeNet.

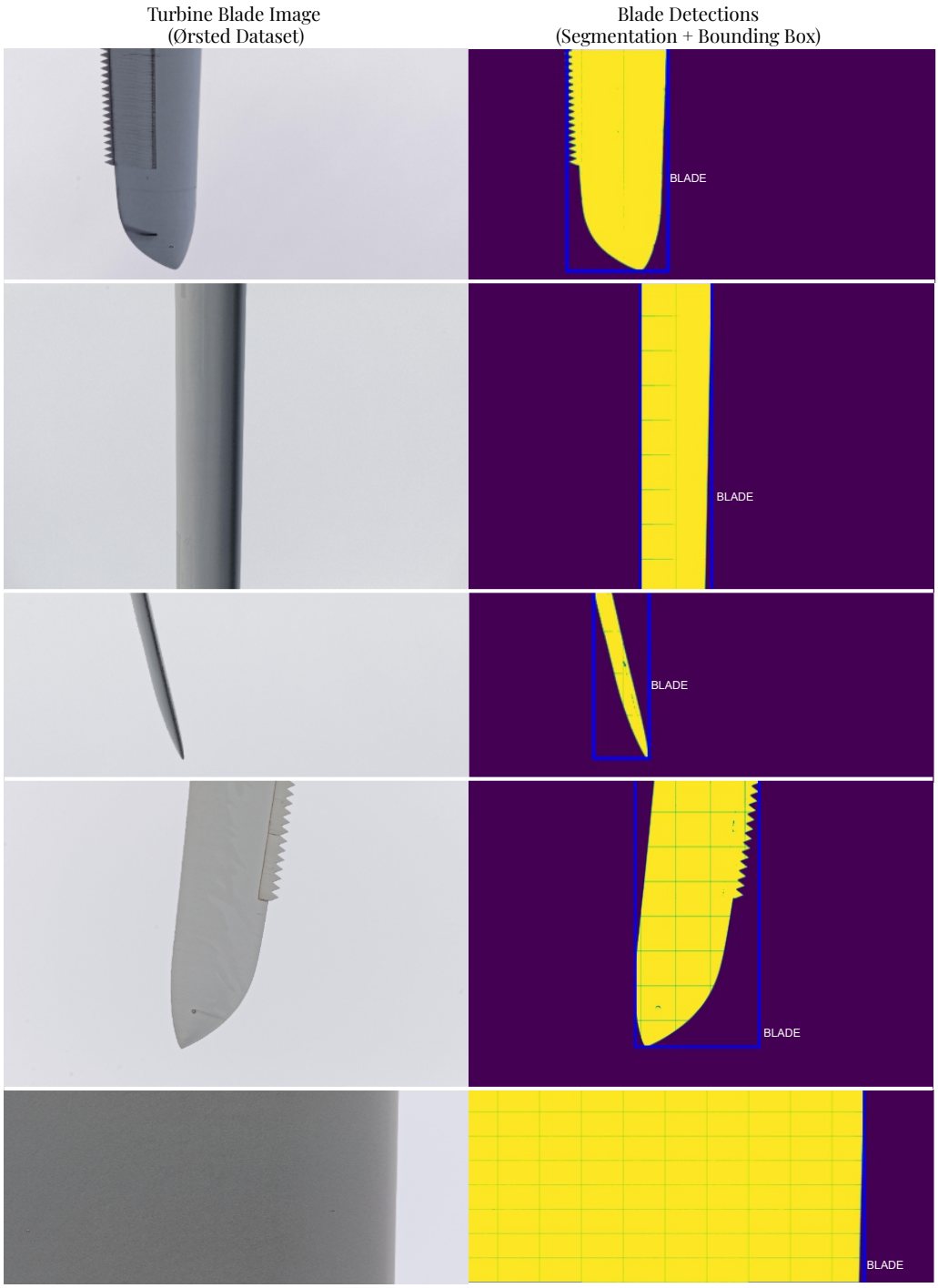

Figure 5.12: Examples of high accuracy instance segmentation and bounding box prediction of Ørsted turbine blades using BladeNet.

5.4.2 Anomaly Detection of Surface Defects

We include a quantitative study of Semi-Supervised anomaly detection approaches over the extracted SLIC super-pixel data of turbine blades. It can be seen in Table 5.2 that the U-GANomaly approach gains the highest Area Under Curve (AUC) value of 0.65 with a 95% Confidence Interval (CI) between 0.65 and 0.66. The performance of PANDA at AUC 0.64 is comparatively close to the performance of Skip-GANomaly which obtains 0.63. However, these models suffer from slower relative inference time compared to that of the Autoencoder which obtained 0.62, but only took 8.61 milliseconds compared with U-GANomaly at 96.34ms. AnoGAN exhibits sluggish inference speed of over 300ms for prediction and obtains the lowest AUC value of 0.61. However, the 95% CI is similar to that of the AE architecture.

The qualitative anomaly detection localisation results of U-GANomaly across the SLIC super pixels of the blade data can be seen in Figure 5.13. This shows that U-GANomaly can detect and segment surface faults in composite blade imagery with high accuracy even when such blade segments are small. Observing the results of this method, it is clear that the performance of the model would warrant it of little use in the real-world, however, we wish to show that we can benchmark these methods on this dataset as a building block for future research. We hope that future methods will increase this performance and result in a commercial product to be used in the real-world application.

Table 5.2: Area Under Curve (AUC) of ROC curve, inference time per image in Milliseconds (I/t(ms)) across semi-supervised anomaly detection methods.

Model	**AUC**	**95% CI (AUC)**	**I/t/(ms)**
AE	0.62	(0.61, 0.63)	**8.61**
AnoGAN	0.61	(0.61, 0.63)	302
GANomaly	0.63	(0.61, 0.63)	48.36
Skip-GANomaly	0.63	(0.62, 0.64)	97.21
PANDA	0.64	(0.63, 0.65)	50.3
U-GANomaly	**0.65**	**(0.65, 0.66)**	96.34

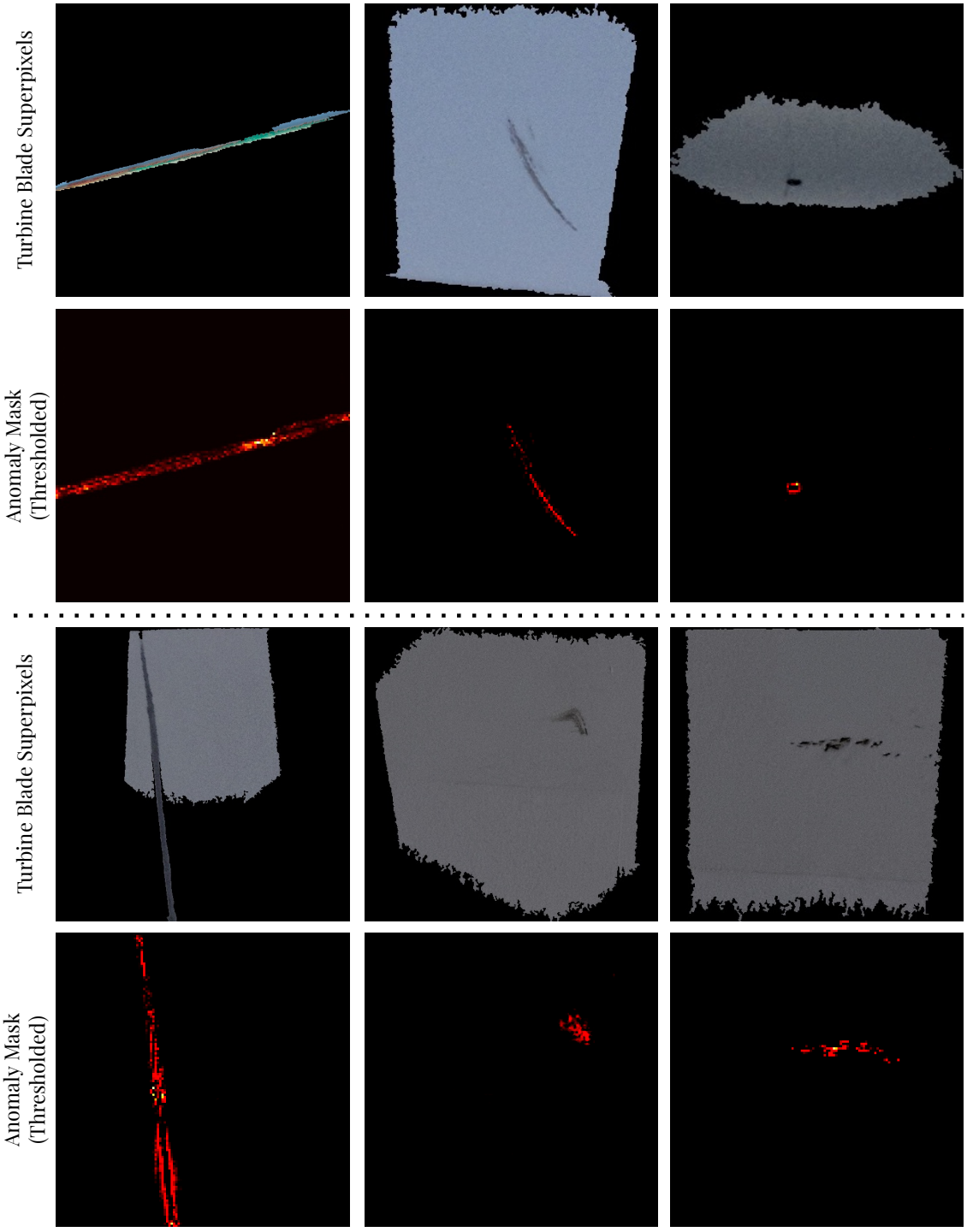

Figure 5.13: Turbine blade SLIC superpixel segmentations containing surface faults together with their corresponding anomaly masks produced by the U-GANomaly architecture.

5.5 Conclusion

Within this chapter we propose a method for accurately detecting and segmenting visual surface faults present in Glass Fibre Reinforced Plastic (GFRP) turbine blade structures. We use official visual turbine blade inspection data collected from the onboard camera of an Unmanned Aerial Vehicle (UAV) across both onshore (DTU Nord Tank [12]) and offshore (Ørsted [29]) wind turbines. We propose BladeNet, a dual-stage architecture to first detect and extract turbine blade parts from given imagery and then to categorise these blade parts as normal or otherwise anomalous while requiring minimal manual annotation in the training process.

We make use of the opening morphological operator to produce pseudo groundtruth for the Ørsted dataset. This process gives accurate pixel-perfect annotations blade detections for a vast amount of turbine blade images for free, but is very fragile as the contrast between the detections in Figure 5.3 and Figure 5.4 illustrates. Due to this instability, we train a U-Net [30] architecture across this pseudo groundtruth together with negative samples (Illustrated in Figure 5.5) which then manages to cleanly segment the failure cases of the operator. We chose to use this U-Net semantic segmentation approach as it gives very accurate, pixel-perfect instance segmentation masks for blade edges while conventional Object Detection methods [18–20] give oscillating edges (as shown in Figures 5.10 and 5.11) on their predicted masks which can fail to capture the leading edge or blade tip which are regions which are most prone to damage [12]. The quantitative results outlined in Table 5.1 show that our U-Net detection method gains vastly superior detection performance, obtaining an Average Precision of 0.995 across the Ørsted dataset and 0.223 when transferred across to the DTU Nordtank dataset while utilising fewer (17.3 million) parameters. One limitation is slow throughput of 3439.21ms per image, however, which could render the real-time, in-situ deployment of this method infeasible. Another limitation is that the turbine blade images in our datasets have singular instances hence allowing us to use a semantic segmentation model for object detection; In the eventuality that the task contains multi-instance objects per image, our method will

not be able to distinguish between the different instances and as such will have to be paired with a detection method capable of multi-instance detection to give the individual categorisation of blades.

The accurate blade extractions are then split into patches by the SLIC method (Section 5.2.2) checked for damage by a suite well established architectures [2,4,6,13, 14] for semi-supervised anomaly detection, the best performing of which, PANDA [13] obtains an AUC of 0.64. We also introduce U-GANomaly (Section 5.8), a fully residual modification of Skip-GANomaly where we replace the DCGAN [125] discriminator module with a U-Net discriminator [17] which obtains a state-of-the-art AUC value of 0.65. We also demonstrate the capability of U-GANomaly to accurately segment the anomalous regions of the turbine blade subsections in Figure 5.13.

CHAPTER 6

Conclusion

Visual anomaly detection is an inherently difficult task even for humans to perform effectively. The open-set nature of anomalous samples means that they may present in any style or shape and as such, effectively representing the anomalous class is difficult and often are rare occurrences in real-world tasks. An example is the task of detecting threat items in x-ray baggage scans; Anomalies may present very rarely, but the security operators will still have to accurately detect them. This can lead to severe class imbalance, or no anomalous class entirely in anomaly detection tasks. This is why it is preferable to train solely across the normal data (Semi-supervised) to essentially detect how deviant from normality a given sample is during inference using the knowledge of normality established during training.

Work in semi-supervised anomaly detection which assumes no access to anomalous data during training can all be broadly categorised into three paradigms: probabilistic, classification-based and reconstruction-based. Although probabilistic and classification-based approaches gain superior results in anomaly detection capability, they often struggle with explainability during inference as to why a given sample is anomalous or otherwise normal. The later, classification-based approaches are also

prone to adversarial examples, which some anomalies in visual tasks with a lesser degree of uniformity may present as normal to these models. As such, the work in this thesis primarily focuses on reconstruction-based approaches which have obtained significant traction lately. As the reconstruction error is a pixel-wise direct score of abnormality, it can be used to explain, to some degree, the regions which are anomalous in a given image which is illustrated at multiple times throughout this thesis. Of particular interest, however, are samples which deviate only slightly from normality and as such, pose a significant challenge to anomaly detection methods.

To address this problem with accurately detecting subtle anomalies, this thesis introduces a method to perform fine-grained anomaly detection via a number of bespoke components. This method is proven through rigorous evaluation across many challenging datasets, to be superior to other such anomaly detection approaches in the literature. Further to this, the thesis then applies this model and others to the task of detecting faults in Glass Fibre Reinforced Plastic (GFRP) wind turbine blades by using a two-stage process consisting of blade detection, followed by anomaly detection. The detection approach not only detects turbine blades with more accuracy, but also segments the edges of the turbine blades with more precision whereas other methods exhibit oscillating detections. The following section outlines the contributions contained in this thesis in more detail.

6.1 Contributions

The work within this thesis begins by outlining, in Chapter 2, a comprehensive review of literature into previously proposed methods in anomaly detection with a direct focus on state-of-the-art methods utilising reconstruction-based approaches.

The Perceptually-Aware Neural Detection of Anomalies (PANDA) method is introduced in Chapter 3 and is a model proposed to better detect fine-grained (visually subtle) anomalies within visual data. The PANDA method applies an adversarially trained autoencoder which is inspired from the VQ-VAE architecture [177] together with a unique and bespoke fine-grained discriminator module. Through the exhaus-

tive qualitative and quantitative evaluation of this method outlined in this chapter, it is evident that the PANDA approach out-performs prior state-of-the-art methods [2, 4–6] and manages to detect subtle, as well as severe anomalies within visual samples presented at inference as illustrated in Figures 3.6 and 3.8.

While autoencoder methods are inherently more stable than GAN-based architectures during training for reconstruction-based anomaly detection, they are still prone to unintentionally fitting to the identity function [214, 215]. One way to circumvent this is to use a denoising approach [105] as this forces the autoencoder to learn the reconstruction of a reference pixel solely using the information of the surrounding pixels when the reference pixel is masked. Prior methods of denoising utilise manually defined noise such as Gaussian noise [216, 217], image masking [218], or a combination of noise [201, 219] and have shown improved results, especially in the task of reconstruction-based anomaly detection. Methods implementing learned noise have been proposed in the task of reconstruction-based anomaly detection [31, 32, 206]. However, they suffer from downfalls which are outlined in the chapter. Such weaknesses include:

- Sampling pre-defined adversarial examples prior to training which maximise latent distance, but minimise perceptual distance [31]. This does not take into account the dynamic nature of the latent representation during training; if it did, it would be very computationally expensive and would increase training latency significantly.

- Singular valued masking [32] which does not tailor the noise to the particular task being trained on, instead opting for a 'one-size-fits-all' approach to masking.

The work presented in Chapter 4 proposes an approach that tackles these limitations within prior works by introducing adversarial training between a noise generator module and a denoising autoencoder. The noise generator is trained to maximise the loss of the reconstruction while the denoising autoencoder is conversely trained to reduce it. This results in optimally difficult, bespoke and continuous noise being

applied to the images and thus leads to more a more robust denoising autoencoder during inference. The results outlined in this chapter suggest strongly that this method performs better than these prior methods.

Lastly, Chapter 5 outlines an approach to detecting flaws in the surface of GFRP wind turbine blades using non-annotated visual drone imagery data. The pipeline initiates by accurately extracting turbine blades using a semantic segmentation U-Net [30] approach trained on pseudo-ground truth masks generated from morphology operators to separate the sky from the turbine blade in some cases. Some of the blade regions extracted in this way would not be accurate, so the best images are manually selected and used for training of the U-Net together with negative samples containing images of ground and sky. This creates a more robust segmentation model that can rectify the annotation of sub-optimal blade extractions to be more accurate, as demonstrated in Figures 5.4 and 5.3. These accurately extracted blade parts are then processed using the SLIC [16] algorithm to generate super-pixel sub-regions of the larger blade. Each superpixel is then processed using a suite of anomaly detection approaches [2, 4, 6] including the method from Chapter 3 [13] as well as a new architecture which we propose in this chapter, U-GANomaly, which utilises a U-Net generator, and then takes inspiration from [17] to utilise a U-Net discriminator which gives pixel-level feedback (rather than a classification loss feedback) to the generator module during training. It is shown in our quantitative experiments that this approach is competitive to all prior methods of anomaly detection featured.

6.2 Limitations and Future Work

This section will discuss the limitations of the work presented in this thesis as well as any potential for future work to be built from the approaches presented. Although much of the work achieves substantial improvement over prior work, there are certainly limitations of such work. This enables the research process to continue in exploring these limitations and further improving methods to obtain even better results than the ones presented within this thesis.

6.2.1 Limitations of Approaches

The goal of the PANDA method in Chapter 3 was to produce an approach which would accurately detect subtle anomalies while reducing the number of false positives. Although the PANDA approach obtains superior results at detecting anomalies, it comes at the cost of added complexity of the model. This is particularly evident in the fine-grained classifier discriminator module which features intricate parts, most of which are inspired from prior work in the task of Fine-Grained Image Categorisation (FGVC) [185, 186]. This complexity can make it difficult to debug when trying alterations to the architecture especially during experimentation. This chapter is the first to propose a fine-grained classifier discriminator to the task of reconstruction-based anomaly detection. This discriminator approach improves performance of our method as demonstrated in Table 3.2 and enables a lower reconstruction error than other prior work compared with that of the DCGAN discriminator [125] as used in [2, 6]. It will be interesting to observe if any future work will adopt this discriminator paradigm to other such areas of computer vision such as image generation or semantic segmentation. It is interesting within our study that we show that Perceptual Loss [175] does not make a significant difference to the performance of our reconstruction-based approach and even damages performance in some tests which we performed experimentally, even though it achieves highly promising results in style transfer tasks.

The adversarial noising approach (ALCN) featured in Chapter 4, although per-

forming well at increasing performance of a novel autoencoder denoising network, there are improvements which could help to further boost performance of the model. The first would be to implement a scheme similar to the work by Adey *et al.* [201] in which the full image is not reconstructed, but rather a mask which when applied to the noised input, creates the original input. This has shown to improve performance, however, in [201] the authors utilise user-defined noise and so the hope is that adding the ALCN adversarial noising technique could enable better performance than each method separately. Another change could be to guide the noising via attention-based methods which would corrupt the maximally important regions rather than globally across the image. Although the ALCN method shows that it is possible to increase performance with adversarially learned noise, it also has a large parameter overhead, so optimisations may be required to reduce the size of the model.

Chapter 5 presents an approach utilising a two-stage process to detect faults in Glass-Fibre Reinforced Plastic (GFRP) materials. As mentioned in the Conclusion of this chapter, the U-Net segmentation model does not utilise instance segmentation, and rather performs semantic segmentation. As the turbine blades are typically inspected close-up and on a one-by-one basis by the unmanned aerial vehicle, a semantic segmentation method can be used in this given use-case. However, if there are multiple blades in a given image, then each blade will be categorised as the same blade and faults on blades may be mixed up if the model is focusing on another blade while running the inspection on a different blade. The other such object detection methods utilised in this work [18–20] succeeded at object detection of each blade. However, the segmentation masks were sub-optimal (as illustrated in Figure 5.10). To this end, perhaps a solution implementing an efficient detection and segmentation (hybrid approach) could be effective for multi-blade detection and segmentation in super high-resolution imagery. Further to this, the act of splitting each blade into super-pixel sub-regions and processing each of them separately is also time-consuming. It would be better to run an anomaly detection algorithm on the full-scale blades, or fixed-shape sliding window patches so as to avoid the

anomalous misclassification of the edges of the superpixels as evidenced in Figure 5.13.

6.2.2 Limitations of Data

On the whole, the datasets used for training and evaluating the presented methods within this thesis offer a rigorous and controlled way in which to benchmark and compare the performance to other such methods. This is with the minor exception of the Ørsted Turbine Blade dataset featured in Chapter 5. This dataset came from engineers at Ørsted who annotated flaws with the turbine blades in-situ during inspection. Annotations contained within the Ørsted dataset are shown visually in Figure 6.1. The annotations are severely limited in that: Neither bounding box nor polygon segmentations of blade outlines were included in the data, annotations of the location of blade damage are rarely accurate such that they contain singular coordinates for large damage regions (Figure 6.1C) or miss the blade entirely (Figure 6.1D).

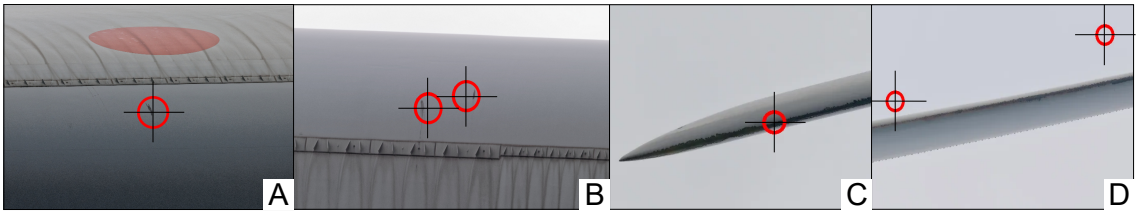

Figure 6.1: Visualisations of blade damage location annotations within the Ørsted Turbine Blade dataset. The centre of each red circle outlines a coordinate annotation of blade damage supplied in the dataset. A and B show the annotations close to the blade damage. C shows that a singular coordinate point is supplied for damage across a large region. D shows the annotations laying off the turbine blade all-together.

The dataset by Shihavuddin [12] was also missing sufficient annotation. Although the authors of the work [12] used the segmentation of damaged parts, they did not publicly release such annotations. As a result we annotated the 1170 images of this dataset with polygon segmentations outlining where the blades are in given images. This allowed the method presented in Chapter 5 to be adequately measured on out-of-distribution examples. We could not, however, annotate the damaged region of the blades due to the technical skill and knowledge required to perform this evaluation; As such, the performance of the anomaly detection stage of the

approach is limited across this dataset. In order to properly evaluate and benchmark our method, accurate and detailed annotations of defective blade parts are required across this dataset. This may be costly, however, due to the aforementioned technical knowledge required to carry this out and the time-consuming nature of such a task.

Bibliography

[1] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," *ArXiv*, vol. abs/1511.08060, 2015. (document), 1, 1.2, 1.1, 2.5, 2.5.1, 2.5.4, 2.5.4, 2.2, 3.1, 3.2.4, 3.3, 3.3.1, 3.2, 3.4, 3.4, 3.5

[2] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly : semi-supervised anomaly detection via adversarial training.," in *14th Asian Conference on Computer Vision*, no. 11363 in Lecture notes in computer science, pp. 622–637, 2019. (document), 1.3, 2.1, 2.4.1, 2.4.3, 2.4.3, 2.2, 2.5, 2.5.1, 2.5.5, 2.8, 2.5.5, 3.1, 3.1, 3.2, 3.2.1, 3.2.2, 3.3, 3.3.1, 3.1, 3.2, 3.3, 3.4, 3.4, 3.4.1, 3.9, 3.10, 4.1, 4.3, 4.4.1, 4.4.2, 4.4.2, 4.5, 5.2.1, 5.2.3, 5.2.4, 5.3, 5.5, 6.1, 6.2.1

[3] N. Bhowmik, Y. Gaus, S. Akcay, J. W. Barker, and T. P. Breckon, "On the impact of object and sub-component level segmentation strategies for supervised anomaly detection within x-ray security imagery.," in *18th IEEE International Conference on Machine Learning and Applications*, 2019. (document), 1.1, 1.3, 2.1, 2.3, 2.4.3, 2.3, 2.5.1, 2.9, 2.5.5, 3.1, 3.1, 3.2.3, 3.3, 3.3.1, 3.2, 3.4, 3.4, 3.5, 4.1

[4] T. Schlegl, P. Seeböck, W. Philipp, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Information Processing in Medical Imaging*, pp. 146–157, 03 2017. (document), 2.4.3, 2.4.3, 2.2, 3.1, 3.1, 3.2, 3.2.2, 3.3, 3.1, 3.2, 3.3, 3.4, 4.1, 4.3, 4.4.1, 4.4.2, 4.4.2, 4.4.3, 4.5, 5.2.3, 5.3, 5.5, 6.1

[5] H. Zenati, C. Foo, B. Lecouat, G. Manek, and V. Chandrasekhar, "Efficient gan-based anomaly detection," *arXiv*, vol. abs/1802.06222, 2018. (document), 2.2, 2.4.3, 2.5.1, 2.5.1, 3.2, 3.3, 3.3.1, 3.1, 3.2, 3.4, 4.1, 4.3, 4.4.1, 4.4.2, 4.4.2, 4.4.3, 4.5, 6.1

[6] S. Akcay, A. Atapour-Abarghouei, and T. Breckon, "Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection," *Proceedings of the International Joint Conference on Neural Networks*, 2019. (document), 1.2, 2.4.1, 2.4.3, 2.4.3, 2.2, 2.5, 2.5.1, 2.5.5, 3.1, 3.1, 3.2.2, 3.3, 3.3.1, 3.1, 3.3, 3.4, 3.4, 3.4.1, 3.9, 3.10, 4.1, 4.3, 4.4.2, 4.4.2, 4.5, 5.2.1, 5.2.3, 5.8, 5.3, 5.5, 6.1, 6.2.1

[7] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs*, vol. 2, 2010. (document), 1.1, 2.5, 2.5.1, 2.5.1, 2.4, 3.3, 3.1, 3.5, 4.3, 4.4, 4.4.1, 4.1, 4.2, 4.4.3, 4.4.3

[8] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009. (document), 1.1, 2.5, 2.5.1, 2.5.1, 2.4, 3.3, 3.1, 3.5, 4.3, 4.4, 4.4.1, 4.1, 4.2, 4.4.3

[9] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection," in *Conference on Computer Vision and Pattern Recognition*, pp. 9584–9592, 2019. (document), 1.1, 2.1, 2.2, 2.5, 2.5.1, 2.5, 2.1, 3.1, 3.3, 3.3.1, 3.3, 3.5, 3.4, 3.5, 3.8, 3.9, 3.10, 4.3, 4.3, 4.4, 4.4.2, 4.3

[10] V. Mahadevan, W. X. LI, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Conference on Computer Vision and Pattern Recognition*, pp. 1975–1981, 2010. (document), 1.1, 2.1, 2.5, 2.5.1, 2.5.3, 2.6, 3.3, 3.2, 3.4, 3.5

[11] D. P. Hughes and M. Salath'e , "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," *CoRR*, vol. abs/1511.08060, 2015. (document), 2.7, 3.6, 4.3, 4.4, 4.4.2, 4.4.2, 4.4, 4.4.3, 4.7

[12] A. Shihavuddin, X. Chen, V. Fedorov, A. Nymark Christensen, N. Andre Brogaard Riis, K. Branner, A. Bjorholm Dahl, and R. Reinhold Paulsen, "Wind turbine surface damage detection by deep learning aided drone inspection analysis," *Energies*, vol. 12, no. 4, 2019. (document), 2.10, 2.6.1, 2.6.2, 5.1, 5.3, 5.5, 6.2.2

[13] J. W. Barker and T. P. Breckon, "Panda: Perceptually aware neural detection of anomalies," in *International Joint Conference on Neural Networks*, 2021. (document), 2.10, 4.3, 5.2.1, 5.2.3, 5.3, 5.5, 6.1

[14] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations*, 2014. (document), 3.3, 3.2, 3.4, 3.5, 3.4, 4.1, 4.4.1, 4.4.2, 4.4.2, 5.3, 5.5

[15] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," *Advances in neural information processing systems*, vol. 26, 2013. (document), 4.1, 4.1

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012. (document), 5.2, 5.2.2, 5.6, 6.1

[17] E. Schonfeld, B. Schiele, and A. Khoreva, "A u-net based discriminator for generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8207–8216, 2020. (document), 1.2, 5.2.4, 5.8, 5.3, 5.5, 6.1

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017. (document), 1.2, 2.6.3, 5.1, 5.2.1, 5.3, 5.10, 5.11, 5.5, 6.2.1

[19] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *International Conference on Computer Vision*, 2019. (document), 1.2, 5.1, 5.2.1, 5.3, 5.10, 5.11, 5.5, 6.2.1

[20] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018. (document), 1.2, 5.1, 5.2.1, 5.3, 5.10, 5.11, 5.5, 6.2.1

[21] H. Shaonian, H. Dongjun, and Z. Xinmin, "Learning multimodal deep representations for crowd anomaly event detection," *Mathematical Problems in Engineering*, pp. 1–13, 2018. (document), 3.3, 3.2

[22] S. M. Steiner-Koller, A. Bolfing, and A. Schwaninger, "Assessment of x-ray image interpretation competency of aviation security screeners," in *43rd Annual 2009 International Carnahan Conference on Security Technology*, pp. 20–27, 2009. 1.1

[23] E. C. A. Conference, "Ecac policy statement in the field of civil aviation facilitation, doc 20, part i," *ECAC Strategy for the Future*, vol. 12, 2018. 1.1

[24] I. C. A. Organization, "Annex 17 - aviation security," *International Standards and Recommended Practices*, vol. 12, 2022. 1.1

[25] E. Parliament, "Regulation no 300/2008," *common rules in the field of civil aviation security and repealing Regulation (EC) No 2320/2002*, vol. 300, no. 10, 2008. 1.1

[26] I. Molotsky, "20% of mock weapons slip by in test of security at airports," *The New York Times*, p. 1, 1987. 1.1

[27] L. Vries, "Airport security fails the test," Jul 2002. 1.1

[28] N. A. Andriyanov, A. K. Volkov, A. K. Volkov, A. A. Gladkikh, and S. D. Danilov, "Automatic x-ray image analysis for aviation security within limited computing resources," *Conference Series: Materials Science and Engineering*, vol. 862, no. 5, 2020. 1.1

[29] J. Barker, N. Bhowmik, and T. Breckon, "Semi-supervised surface anomaly detection of composite wind turbine blades from drone imagery," in *International Conference on Computer Vision Theory and Applications*, 2022. 1.1, 2.6.2, 5.5

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," vol. 9351, pp. 234–241, 2015. 1.2, 2.4.3, 2.4.3, 5.1, 5.2.1, 5.2.4, 5.3, 5.5, 6.1

[31] M. Salehi, A. Arya, B. Pajoum, M. Otoofi, A. Shaeiri, M. H. Rohban, and H. R. Rabiee, "Arae: Adversarially robust training of autoencoders improves novelty detection," *Neural Networks*, vol. 144, pp. 726–736, 2021. 1.2, 1.4, 2.4.2, 2.5.1, 4.1, 4.3, 4.4.1, 4.4.1, 4.5, 6.1

[32] J. T. Jewell, V. Reza Khazaie, and Y. Mohsenzadeh, "One-class learned encoder-decoder network with adversarial context masking for novelty detection," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 2856–2866, 2022. 1.2, 1.4, 2.4.1, 2.4.2, 2.5.1, 4.1, 4.3, 4.4.1, 4.4.1, 4.5, 4.6, 6.1

[33] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Conference on Computer Vision and Pattern Recognition*, pp. 481–490, IEEE Computer Society, 2019. 1.2, 4.1, 4.3, 4.4.1, 4.4.1, 4.5

[34] P. Perera, R. Nallapati, and B. Xiang, "Ocgan: One-class novelty detection using gans with constrained latent representations," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2893–2901, 2019. 1.2, 2.4.3, 4.1, 4.3, 4.4.1, 4.4.1, 4.5

[35] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402, 2018. 1.2, 4.3, 4.4.1, 4.4.1, 4.5

[36] H. S. Vu, D. Ueta, K. Hashimoto, K. Maeno, S. Pranata, and S. Shen, "Anomaly detection with adversarial dual autoencoders," 2019. 1.4, 2.4.3, 4.4.1, 4.5

[37] Y. F. A. Gaus, N. Bhowmik, S. Akçay, P. M. Guillén-Garcia, J. W. Barker, and T. P. Breckon, "Evaluation of a dual convolutional neural network architecture for object-wise anomaly detection in cluttered x-ray security imagery," in *International Joint Conference on Neural Networks*, pp. 1–8, 2019. 2.1, 2.3, 3.1, 3.1, 4.1

[38] N.-C. Ristea, N. Madan, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "Self-supervised predictive convolutional attentive block

for anomaly detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2.1, 2.4, 2.4.3

[39] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *International Conference on Learning Representations*, 2020. 2.1, 2.2, 2.3

[40] B. Antić and B. Ommer, "Video parsing for abnormality detection," in *International Conference on Computer Vision*, pp. 2415–2422, 2011. 2.2

[41] T. Ehret, A. Davy, J.-M. Morel, and M. Delbracio, "Image anomalies: A review and synthesis of detection methods," *Journal of Mathematical Imaging and Vision*, vol. 61, pp. 710–743, 2019. 2.2

[42] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014. 2.2

[43] L. Wang, D. Zhang, J. Guo, and Y. Han, "Image anomaly detection using normal data only by latent space resampling," *Applied Sciences*, vol. 10, no. 23, 2020. 2.2

[44] X. Xie and M. Mirmehdi, "Texems: Texture exemplars for defect detection on random textured surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1454–1464, 2007. 2.2

[45] J. Kim and C. D. Scott, "Robust kernel density estimation," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2529–2565, 2012. 2.2

[46] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *Machine Learning and Data Mining in Pattern Recognition*, pp. 61–75, 2007. 2.2

[47] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962. 2.2

[48] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018. 2.2

[49] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, 2009. 2.2

[50] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "VT-ADL: A vision transformer network for image anomaly detection and localization," in *30th IEEE International Symposium on Industrial Electronics*, 2021. 2.2

[51] B. Kim, K. Kwon, C. Oh, and H. Park, "Unsupervised anomaly detection in mr images using multicontrast information," *Medical Physics*, vol. 48, no. 11, pp. 7346–7359, 2021. 2.2

[52] P. Seeböck, S. M. Waldstein, S. Klimscha, H. Bogunovic, T. Schlegl, B. S. Gerendas, R. Donner, U. Schmidt-Erfurth, and G. Langs, "Unsupervised identification of disease marker candidates in retinal oct imaging data," *IEEE Transactions on Medical Imaging*, vol. 38, no. 4, pp. 1037–1047, 2019. 2.2

[53] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, p. 3861–3870, 2017. 2.2

[54] J. Yang, R. Xu, Z. Qi, and Y. Shi, "Visual anomaly detection for images: A systematic survey," *Procedia Computer Science*, vol. 199, pp. 471–478, 2022. The 8th International Conference on Information Technology and Quantitative Management: Developing Global Digital Economy after COVID-19. 2.2

[55] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," *arXiv preprint arXiv:2005.02357*, 2020. 2.2, 2.3

[56] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "Padim: A patch distribution modeling framework for anomaly detection and localization," in *Pattern Recognition. International Workshops and Challenges* (A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, eds.), pp. 475–489, 2021. 2.2

[57] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *IEEE Computer Vision and Pattern Recognition Conference*, 2022. 2.2

[58] S. Lee, S. Lee, and B. C. Song, "Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization," *IEEE Access*, vol. 10, pp. 78446–78454, 2022. 2.2

[59] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems* (S. Solla, T. Leen, and K. Müller, eds.), vol. 12, MIT Press, 1999. 2.3, 4.1

[60] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computing.*, vol. 13, no. 7, p. 1443–1471, 2001. 2.3

[61] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Data Warehousing and Knowledge Discovery*, pp. 170–180, 2002. 2.3

[62] D. Tax and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, pp. 45–66, 2004. 2.3

[63] F. Sohrab, J. Raitoharju, M. Gabbouj, and A. Iosifidis, "Subspace support vector data description," 2018. 2.3

[64] F. Nunnari, H. M. T. Alam, and D. Sonntag, "Anomaly detection for skin lesion images using replicator neural networks," in *Machine Learning and Knowledge Extraction*, pp. 225–240, 2021. 2.3

[65] A. S. Hashmi and T. Ahmad, "Gp-elm-rnn: Garson-pruned extreme learning machine based replicator neural network for anomaly detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1768–1774, 2022. 2.3

[66] L. Tóth and G. Gosztolya, "Replicator neural networks for outlier modeling in segmental speech recognition," in *Advances in Neural Networks* (F.-L. Yin, J. Wang, and C. Guo, eds.), pp. 996–1001, 2004. 2.3

[67] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, 2004. 2.3

[68] R. Dwivedi, T. Dutta, and Y.-C. Hu, "A leaf disease detection mechanism based on l1-norm minimization extreme learning machine," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022. 2.3

[69] H. Dai, J. Cao, T. Wang, M. Deng, and Z. Yang, "Multilayer one-class extreme learning machine," *Neural Networks*, vol. 115, pp. 11–22, 2019. 2.3

[70] J. Cao, K. Zhang, M. Luo, C. Yin, and X. Lai, "Extreme learning machine and adaptive sparse representation for image classification," *Neural Networks*, vol. 81, pp. 91–102, 2016. 2.3

[71] J. Cao, K. Zhang, H. Yong, X. Lai, B. Chen, and Z. Lin, "Extreme learning machine with affine transformation inputs in an activation function," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2093–2107, 2019. 2.3

[72] Y. Yang and Q. M. J. Wu, "Multilayer extreme learning machine with subnetwork nodes for representation learning," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2570–2583, 2016. 2.3

[73] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su, "One-class classification with extreme learning machine," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–11. 2.3

[74] L. Kasun, H. Zhou, G.-B. Huang, and C.-M. Vong, "Representational learning with elms for big data," *IEEE Intelligent Systems*, vol. 28, pp. 31–34, 2013. 2.3

[75] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, 2015. 2.3

[76] T. Wang, J. Cao, X. Lai, and B. Chen, "Deep weighted extreme learning machine," *Cognitive Computation*, vol. 10, 2018. 2.3

[77] C. Wong, C.-M. Vong, P.-K. Wong, and J. Cao, "Kernel-based multilayer extreme learning machines for representation learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–6, 2016. 2.3

[78] V. Vapnik, *Statistical learning theory*. Wiley, 1998. 2.3

[79] Y. Bengio and Y. Lecun, *Scaling learning algorithms towards AI*. MIT Press, 2007. 2.3

[80] R. Chalapathy, A. Krishna Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv*, 2018. 2.3, 2.4

[81] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," KDD, p. 665–674, 2017. 2.3

[82] P. Oza and V. M. Patel, "One-class convolutional neural network," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 277–281, 2018. 2.3

[83] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," pp. 3379–3388, 2018. 2.3, 2.4.3

[84] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018. 2.3

[85] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," *IEEE International Conference on Image Processing*, pp. 1577–1581, 2017. 2.3, 2.4.3, 2.4.3, 3.3, 3.2

[86] 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*, pp. 4393–4402, 2018. 2.3

[87] P. Perera and V. Patel, "Learning deep features for one-class classification," *IEEE Transactions on Image Processing*, vol. PP, 2018. 2.3

[88] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems*, vol. 31, 2018. 2.3, 3.1, 3.4

[89] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations.," in *6th International Conference on Learning Representations*, 2018. 2.3, 2.4.3

[90] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, "Learning and evaluating representations for deep one-class classification," in *International Conference on Learning Representations*, 2021. 2.3

[91] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015. 2.3

[92] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), pp. 649–666, 2016. 2.3

[93] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9659–9669, 2021. 2.3, 2.4.3

[94] S. Schneider, D. Antensteiner, D. Soukup, and M. Scheutz, "Autoencoders - a comparative analysis in the realm of anomaly detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1986–1992, 2022. 2.4

[95] F. Dong, Y. Zhang, and X. Nie, "Dual discriminator generative adversarial network for video anomaly detection," *IEEE Access*, vol. 8, pp. 88170–88176, 2020. 2.4

[96] M. I. Georgescu, R. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "A background-agnostic framework with adversarial training for abnormal event detection in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2021. 2.4

[97] D. E. Rumelhart and J. L. McClelland, *Learning Internal Representations by Error Propagation*, pp. 318–362. 1987. 2.4.1

[98] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of International Conference on Machine Learning Workshop on Unsupervised and Transfer Learning*, vol. 27 of *Proceedings of Machine Learning Research*, pp. 37–49, 2012. 2.4.1

[99] P. Gallinari, Y. Lecun, S. Thiria, and F. Fogelman Soulie, "Memoires associatives distribuees: Une comparaison (distributed associative memories: A comparison)," in *Proceedings of COGNITIVA*, 1987. 2.4.1

[100] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning – ICANN 2011*, pp. 52–59, 2011. 2.4.1

[101] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *VISIGRAPP*, 2019. 2.4.1

[102] 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*, p. 665–674, 2017. 2.4.1

[103] P. A. Adey, S. Akçay, M. Bordewich, and T. Breckon, "Autoencoders without reconstruction for textural anomaly detection," *2021 International Joint Conference on Neural Networks*, pp. 1–8, 2021. 2.4.1, 2.4.2, 2.4.3, 4.6

[104] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–742, 2016. 2.4.1

[105] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising autoencoders as generative models," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, p. 899–907, 2013. 2.4.1, 2.4.2, 6.1

[106] V. Zavrtanik, M. Kristan, and D. Skočaj, "Reconstruction by inpainting for visual anomaly detection," *Pattern Recognition*, vol. 112, p. 107706, 2021. 2.4.1, 2.4.2, 4.1

[107] H. Steck, "Autoencoders that don't overfit towards the identity," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 19598–19608, 2020. 2.4.1, 2.4.2

[108] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, p. 1096–1103, 2008. 2.4.1

[109] S. Wager, S. Wang, and P. Liang, "Dropout training as adaptive regularization," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, p. 351–359, 2013. 2.4.1, 2.4.2

[110] X. Ma, Y. Gao, Z. Hu, Y. Yu, Y. Deng, and E. Hovy, "Dropout with expectation-linear regularization," 2016. 2.4.1, 2.4.2

[111] D. Kunin, J. Bloom, A. Goeva, and C. Seed, "Loss landscapes of regularized linear autoencoders," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 3560–3569, 2019. 2.4.1

[112] S. Wang and C. Manning, "Fast dropout training," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28 of *Proceedings of Machine Learning Research*, pp. 118–126, 2013. 2.4.1

[113] D. P. Helmbold and P. M. Long, "On the inductive bias of dropout," *The Journal of Machine Learning Research*, vol. 16, no. 1, p. 3403–3454, 2015. 2.4.1

[114] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Computation*, vol. 7, no. 1, pp. 108–116, 1995. 2.4.1

[115] S. Mohamed, R. Ejbali, and M. Zaied, "Denoising autoencoder with dropout based network anomaly detection," *International Conference on Software Engineering Advances*, p. 110, 2019. 2.4.1

[116] G. An, "The effects of adding noise during backpropagation training on a generalization performance," *Neural Computing*, vol. 8, no. 3, p. 643–674, 1996. 2.4.1

[117] D. Xu, Y. Yan, E. Ricci, and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Computer Vision and Image Understanding*, vol. 156, pp. 117–127, 2017. 2.4.2

[118] A. Creswell, A. Pouplin, and A. A. Bharath, "Denoising adversarial autoencoders: classifying skin lesions using limited labelled training data," *IET Computer Vision*, vol. 12, no. 8, pp. 1105–1111, 2018. 2.4.2

[119] B. Poole, J. N. Sohl-Dickstein, and S. Ganguli, "Analyzing noise in autoencoders and deep networks," *ArXiv*, vol. abs/1406.1831, 2014. 2.4.2

[120] A. Kascenas, N. Pugeault, and A. Q. O'Neil, "Denoising autoencoders for unsupervised anomaly detection in brain mri," in *Medical Imaging with Deep Learning*, 2022. 2.4.2, 4.1

[121] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014. 2.4.3, 3.2.1

[122] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 3481–3490, 2018. 2.4.3

[123] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *34th International Conference on Machine Learning - Volume 70*, p. 214–223, 2017. 2.4.3

[124] T. Schlegl, P. Seeböck, S. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, vol. 54, pp. 30–44, 2019. 2.4.3, 2.4.3, 3.1, 3.1, 3.2, 3.2.2, 3.3, 3.2, 3.4, 4.1, 4.3, 5.2.1

[125] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations*, 2016. 2.4.3, 2.4.3, 3.4.1, 5.2.4, 5.5, 6.2.1

[126] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in gans," in *international joint conference on neural networks*, pp. 1–10, 2020. 2.4.3

[127] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *International Conference on Learning Representations*, 2017. 2.4.3

[128] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, 2015. 2.4.3

[129] K. Fukushima, "Cognitron: a self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3-4, p. 121—136, 1975. 2.4.3

[130] Y. Sterchi, N. Hättenschwiler, S. Michel, and A. Schwaninger, "Relevance of visual inspection strategy and knowledge about everyday objects for x-ray baggage screening," *2017 International Carnahan Conference on Security Technology*, pp. 1–6, 2017. 2.4.3

[131] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," *2019 IEEE Winter Conference on Applications of Computer Vision*, pp. 1896–1904, 2019. 2.4.3

[132] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5967–5976, 2017. 2.4.3

[133] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision*, 2004. 2.4.3

[134] J. Donahue, T. Darrell, and K. Philipp, "Adversarial feature learning," in *5th International Conference on Learning Representations*, International Conference on Learning Representations, 2019. 2.4.3

[135] F. D. Mattia, P. Galeone, M. D. Simoni, and E. Ghelfi, "A survey on gans for anomaly detection," *ArXiv*, vol. abs/1906.11632, 2019. 2.4.3

[136] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International Conference on Data Mining*, pp. 727–736, 2018. 2.4.3

[137] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin, "Alice: Towards understanding adversarial learning for joint distribution matching," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, 2017. 2.4.3

[138] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *Advances in Neural Information Processing Systems*, vol. 31, 2018. 2.4.3

[139] I. Haloui, J. S. Gupta, and V. Feuillard, "Anomaly detection with wasserstein gan," *ArXiv*, vol. abs/1812.02463, 2018. 2.4.3

[140] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 214–223, 2017. 2.4.3

[141] K. Lei, M. Mardani, J. M. Pauly, and S. S. Vasanawala, "Wasserstein gans for mr imaging: From paired to unpaired training," *IEEE Transactions on Medical Imaging*, vol. 40, pp. 105–115, 2019. 2.4.3

[142] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 2.4.3, 3.2.2

[143] K. Zhou, S. Gao, J. Cheng, Z. Gu, H. Fu, Z. Tu, J. Yang, Y. Zhao, and J. Liu, "Sparse-gan: Sparsity-constrained generative adversarial network for anomaly detection in retinal oct image," *IEEE 17th International Symposium on Biomedical Imaging*, pp. 1227–1231, 2020. 2.4.3

[144] C. Chen, P. Chen, H. Song, Y. Tao, Y. Xie, S. Ding, and L. Ma, "Anomaly detection by one class latent regularized networks," *arXiv: Computer Vision and Pattern Recognition*, 2020. 2.4.3

[145] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision*, pp. 618–626, 2017. 2.4.3

[146] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *European Conference on Computer Vision* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), pp. 485–503, 2020. 2.4.3

[147] D. Kimura, S. Chaudhury, M. Narita, A. Munawar, and R. Tachibana, "Adversarial discriminative attention for robust anomaly detection," in *2020 IEEE Winter Conference on Applications of Computer Vision*, pp. 2161–2170, 2020. 2.4.3

[148] M. Z. Zaheer, J. Lee, M. Astrid, and S. Lee, "Old is gold: Redefining the adversarially learned one-class classifier training paradigm," in *2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14171–14181, 2020. 2.4.3

[149] P. Perera, V. I. Morariu, R. Jain, V. Manjunatha, C. Wigington, V. Ordonez, and V. M. Patel, "Generative-discriminative feature representations for open-set recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2.4.3

[150] X. Xia, X. Pan, X. He, J. Zhang, N. Ding, and L. Ma, "Discriminative-generative representation learning for one-class anomaly detection," 2021. 2.4.3

[151] P. Oza, H. V. Nguyen, and V. M. Patel, "Multiple class novelty detection under data distribution shift," in *European Conference on Computer Vision*, pp. 432–449, 2020. 2.4.3

[152] J. W. Song, K. Kong, Y. I. Park, S. Kim, and S.-J. Kang, "Anoseg: Anomaly segmentation network using self-supervised learning," *ArXiv*, vol. abs/2110.03396, 2021. 2.4.3

[153] V. Zavrtanik, M. Kristan, and D. Skocaj, "Draem - a discriminatively trained reconstruction embedding for surface anomaly detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8330–8339, 2021. 2.4.3

[154] K. Perlin, "An image synthesizer," *SIGGRAPH Computer Graphics*, vol. 19, no. 3, p. 287–296, 1985. 2.4.3

[155] M. Salehi, A. Eftekhar, N. Sadjadi, M. H. Rohban, and H. R. Rabiee, "Puzzle-ae: Novelty detection in images through solving puzzles," 2020. 2.4.3

[156] V. Narayan and B. Shaju, *Malware and Anomaly Detection Using Machine Learning and Deep Learning Methods*, pp. 104–131. 01 2020. 2.5.1

[157] X. Hoque and S. Sharma, *Ensembled Deep Learning Approach for Maritime Anomaly Detection System*, pp. 862–869. 09 2019. 2.5.1

[158] X.Fang, W. Guo, Q. Li, J. Zhu, Z. Chen, J. Yu, B. Zhou, and H. Yang, "Sewer pipeline fault identification using anomaly detection algorithms on video sequences," *IEEE Access*, vol. 8, pp. 39574–39586, 2020. 2.5.1

[159] D. of Economic and D. Social Affairs, "Growing at a slower pace, world population is expected to reach 9.7 billion in 2050 and could peak at nearly 11 billion around 2100," 2019. 2.5.4

[160] J. Ranganathan, R. Waite, T. Searchinger, and C. Hanson, "How to sustainably feed 10 billion people by 2050, in 21 charts," 2018. 2.5.4

[161] A. Ficke, C. Cowger, G. Bergstrom, and G. Brodal, "Understanding yield loss and pathogen biology to improve disease management: Septoria nodorum blotch - a case study in wheat," *Plant Disease*, vol. 102, no. 4, pp. 696–707, 2018. 2.5.4

[162] C. Civil Aviation Authority, "2022 quarter one flight data," 2022. 2.5.5

[163] T. Rogers, N. Jaccard, E. Morton, and L. Griffin, "Automated x-ray image analysis for cargo security: Critical review and future promise," *Journal of X-Ray Science and Technology*, vol. 25, 2016. 2.5.5

[164] F. P. García Márquez and A. M. Peco Chacón, "A review of non-destructive testing on wind turbines blades," *Renewable Energy*, vol. 161, 2020. 2.6

[165] M. Shafiee, Z. Zhou, L. Mei, F. Dinmohammadi, J. Karama, and D. Flynn, "Unmanned aerial drones for inspection of offshore wind turbines: A mission-critical failure analysis," *Robotics*, vol. 10, no. 1, 2021. 2.6

[166] L. Wang and Z. Zhang, "Automatic detection of wind turbine blade surface cracks based on uav-taken images," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7293–7303, 2017. 2.6

[167] R. Girshick, "Fast r-cnn," *CoRR*, vol. abs/1504.08083, 2015. 2.6.3

[168] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks.," in *NIPS*, pp. 91–99, 2015. 2.6.3

[169] J. Redmon, S. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016. 2.6.3

[170] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517–6525, 2017. 2.6.3

[171] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015. 3.1

[172] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," *arXiv preprint arXiv:1802.03041*, 2018. 3.1

[173] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11383, pp. 161–169, 2018. 3.1

[174] H. S. Vu, D. Ueta, K. Hashimoto, K. Maeno, S. Pranata, and S. Shen, "Anomaly detection with adversarial dual autoencoders," *ArXiv*, vol. abs/1902.06924, 2019. 3.1, 3.1, 3.4, 4.4.1

[175] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016. 3.1, 3.2, 3.2.3, 6.2.1

[176] Q. Wu, C. Fan, Y. Li, Y. Li, and J. Hu, "A novel perceptual loss function for single image super-resolution," *Multimedia Tools and Applications.*, vol. 79, no. 29–30, p. 21265–21278, 2020. 3.2

[177] A. Razavi, A. van den Oord, and O. Vinyals, *Generating Diverse High-Fidelity Images with VQ-VAE-2*. 2019. 3.2.1, 6.1

[178] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *31st International Conference on Neural Information Processing Systems*, p. 6309–6318, 2017. 3.2.1

[179] A. Majumdar and A. Tripathi, "Asymmetric stacked autoencoder," in *International Joint Conference on Neural Networks*, pp. 911–918, 2017. 3.2.1

[180] A. Siddiqua and G. Fan, "Asymmetric supervised deep autoencoder for depth image based 3d model retrieval," in *IEEE Visual Communications and Image Processing*, pp. 1–4, 2019. 3.2.1

[181] J.-H. Kim, J.-H. Choi, J. Chang, and J.-S. Lee, "Efficient deep learning-based lossy image compression via asymmetric autoencoder and pruning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2063–2067, 2020. 3.2.1

[182] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015. 3.2.1, 3.2.3, 3.4

[183] A. Razavi, A. Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 14866–14876, 2019. 3.2.2

[184] Y. Wang and Z. Wang, "A survey of recent work on fine-grained image classification techniques," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 210 – 214, 2019. 3.2.2

[185] T. Hu, H. Qi, Q. Huang, and Y. Lu, "See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification," 2019. 3.2.2, 6.2.1

[186] Y. Wang, V. Morariu, and L. Davis, "Learning a discriminative filter bank within a cnn for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4148–4157, 2018. 3.2.2, 6.2.1

[187] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fe, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 3.2.3

[188] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 935–942, 2009. 3.3, 3.2

[189] J. Kim and K. Grauman, "Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 3.3, 3.2

[190] L. Weixin, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, p. 18–32, 2014. 3.3, 3.2

[191] C. Yang, Y. Junsong, and L. Ji, "Abnormal event detection in crowded scenes using sparse representation," *Pattern Recognition*, vol. 46, no. 7, pp. 1851–1864, 2013. 3.3, 3.2

[192] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," in *The British Machine Vision Conference*, 2015. 3.3, 3.2

[193] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations* (Y. Bengio and Y. LeCun, eds.), 2015. 3.3.1

[194] D. Smolyak, K. Gray, S. Badirli, and G. Mohler, "Coupled igmm-gans with applications to anomaly detection in human mobility data," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 4, 2020. 3.1

[195] J. Tack, S. Mo, J. Jeong, and J. Shin, "Csi: Novelty detection via contrastive learning on distributionally shifted instances," in *Advances in Neural Information Processing Systems*, 2020. 3.1, 3.4

[196] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 15663–15674, 2019. 3.1, 3.4

[197] T. W. Tang, W. H. Kuo, J. H. Lan, C. F. Ding, H. Hsu, and H. T. Young, "Anomaly detection neural network with dual auto-encoders gan and its industrial inspection applications," *Sensors*, vol. 20, p. 3336, 2020. 3.3

[198] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351 of *LNCS*, pp. 234–241, 2015. 3.3

[199] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds.), vol. 27, 2014. 4.1

[200] Z. Zhang, M. Li, and J. Yu, "On the convergence and mode collapse of gan," in *SIGGRAPH Asia 2018 Technical Briefs*, 2018. 4.1

[201] P. A. Adey, S. Akçay, M. J. Bordewich, and T. P. Breckon, "Autoencoders without reconstruction for textural anomaly detection," in *2021 International Joint Conference on Neural Networks*, pp. 1–8, 2021. 4.1, 4.2, 6.1, 6.2.1

[202] C. Baur, S. Denner, B. Wiestler, N. Navab, and S. Albarqouni, "Autoencoders for unsupervised anomaly segmentation in brain mr images: A comparative study," *Medical Image Analysis*, vol. 69, p. 101952, 2021. 4.1

[203] Q. Xiang and X. Pang, "Improved denoising auto-encoders for image denoising," in *11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–9, 2018. 4.1

[204] A. Villar-Corrales, F. Schirrmacher, and C. Riess, "Deep learning architectural designs for super-resolution of noisy images," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1635–1639, 2021. 4.1

[205] L. Yasenko, Y. Klyatchenko, and O. Tarasenko-Klyatchenko, "Image noise reduction by denoising autoencoder," in *IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pp. 351–355, 2020. 4.1

[206] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016. 4.1, 6.1

[207] Y. Y. Yuhji Matsuo, Akira Yanagisawa, "A global energy outlook to 2035 with strategic considerations for asiaand middle east energy supply and demand interdependencies," *The Institute of Energy Economics*, vol. 13, no. 4, pp. 79–91, 2013. 5.1

[208] GWEC, "Global wind energy outlook 2008," tech. rep., Global Wind Energy Council, Brussels, Belgium, 2008. 5.1

[209] C. L. Archer and M. Z. Jacobson, "Evaluation of global wind power," *Journal of Geophysical Research: Atmospheres*, vol. 110, no. 12, 2005. 5.1

[210] G. Sinden, "Characteristics of the uk wind resource: Long-term patterns and relationship to electricity demand," in *Energy Policy*, vol. 35, pp. 112–127, 2007. 5.1

[211] L. Mishnaevsky, K. Branner, H. N. Petersen, J. Beauson, M. McGugan, and B. F. Sørensen, "Materials for wind turbine blades: An overview," in *Materials*, vol. 10, 2017. 5.1

[212] H. Meng, F.-S. Lien, E. Yee, and J. Shen, "Modelling of anisotropic beam for rotating composite wind turbine blade by using finite-difference time-domain (fdtd) method," *Renewable Energy*, vol. 162, pp. 2361–2379, 2020. 5.1

[213] C. U. G. Anne Juengert, "Inspection techniques for wind turbine blades using ultrasound and sound waves," in *Non-Destructive Testing in Civil Engineering*, 2009. 5.1

[214] H. Steck, "Autoencoders that don't overfit towards the identity," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 19598–19608, 2020. 6.1

[215] A. Grover and S. Ermon, "Uncertainty autoencoders: Learning compressed representations via variational information maximization," in *22nd International Conference on Artificial Intelligence and Statistics*, vol. 89 of *Proceedings of Machine Learning Research*, pp. 2514–2524, 2019. 6.1

[216] P. Liang, W. Shi, and X. Zhang, "Remote sensing image classification based on stacked denoising autoencoder," *Remote Sensing*, vol. 10, no. 1, 2018. 6.1

[217] J. A. Rodríguez-Rodríguez, M. A. Molina-Cabello, R. Benítez-Rochel, and E. López-Rubio, "Test time augmentation by regular shifting for deep denoising autoencoder networks," in *International Joint Conference on Neural Networks*, pp. 1–7, 2021. 6.1

[218] Q. Wu, Y. G. Hang Ye, L. W. Huishuai Zhang, and D. He, "Denoising masked autoencoders are certifiable robust vision learners," *arXiv:2210.06983*, 2022. 6.1

[219] N. M. Tun, A. I. Gavrilov, and N. L. Tun, "Facial image denoising using convolutional autoencoder network," in *International Conference on Industrial Engineering, Applications and Manufacturing*, pp. 1–5, 2020. 6.1