Security and Deep Learning
Verifying the Authenticity of CT Images

Gerlofsma, M.

Award date:
2020

MASTER THESIS

Security and Deep Learning: Verifying the Authenticity of CT Images

Author
Markus Gerlofsma

Supervisors
Prof. Dr. Milan Petković
Ir. Peter van Liesdonk

Eindhoven University of Technology
Department of Mathematics and Computer Science

Philips Research

December 20, 2019

Abstract

Recent advancements in computational power and data availability have led to a new surge of autonomous Artificial Intelligence (AI) applications. Although the majority of these applications serve honest purposes, a recent attack has applied a generative adversarial network to tamper with computed tomography (CT) images within a hospital. The effectiveness of the attack demonstrated that many security controls remain absent or misconfigured in hospitals, and that sole reliance on the proficiency of medical staff is not sufficient to detect this type of image tampering attack. This thesis investigates a solution which is largely independent from the infrastructure of a hospital, yet promises to be an effective tool to detect a variety of attacks which seek to tamper with medical images. The solution harnesses distinct elements within CT scans which are inadvertently embedded during the image acquisition process by their respective scanner. These elements may consequently act as device fingerprints, which are sensitive to distortions. This thesis demonstrates that these device fingerprints facilitate accurate CT scanner classification, and furthermore enable the detection of CT images tampered by generative adversarial networks.

Preface

This thesis marks the end of my graduation project and the entirety of my Master's program. I am first of all grateful for the opportunity to complete this final phase at Philips Research. I have been granted great freedom while carrying out my graduation project, which consequently allowed me to explore and combine a lot of interesting fields of research. The exploratory and experimental character of this project furthermore rekindled my love for programming.

I would like to sincerely thank my supervisors Milan Petković and Peter van Liesdonk for their large time investment during this project. They continuously provided me with great ideas, feedback on my experiments, and ensured that I stay grounded throughout the entire project. Milan was always quick to provide me with all the necessary guidance and resources I required throughout my internship, and Peter has reviewed my work so often that he likely knows most of this thesis by heart. I would also like to thank Dimitrios Mavroedis and Vlado Menkovski for helpful and insightful discussions which incited me to consider challenges from a different angle. I am furthermore grateful to the data mining group of the university which granted me prolonged access to their compute cluster.

Finally, I would like to sincerely thank my family: Ingrid, Tony and Désirée. They supported and encouraged me throughout my entire Master's study, and are the primary reason that I started this study in the first place. I am grateful for their continuous support and would never have reached this goal without their encouragement.

Contents

Chapter 1

Introduction

The concept of Artificial Intelligence (AI) has existed for decades. Simple applications, such as a program for playing checkers, have been introduced as early as 1959 [84]. However, recent advancement in computational power, as well as the rise of practical implementations of AI algorithms, have led to a surge of novel and "smart" applications. Notably, advancements in deep learning have paved the way for an abundance of applications which aim to improve businesses processes, or simply relief humans in their daily lives. As an example within the healthcare sector, deep learning has been embraced to support medical staff in making diagnoses [56] and by predicting future complications of patients [16]. The growth in popularity and applicability of deep learning has also found its way into the domain of information security. Although deep learning and AI have opened up a new path to improve existing security tasks, the technology has also raised new concerns and challenges within information security itself. One recent example is a malicious application, capable of injecting and removing malign nodules from computed tomography (CT) scans by abusing a novel deep learning approach to alter the images autonomously [63]. This autonomy combined with the high quality of the modifications have made this attack challenging to detect.

This thesis surveys the applicability and impact of deep learning within the information security domain, and discusses the recent attack on CT images in more detail. This attack will furthermore serve as the main research topic of this thesis. In an effort to mitigate the threat of this attack, a potential solution is proposed which aims to negate this version of the attack as well as potential future variations. Finally, multiple experiments have been conducted to motivate and evaluate the performance of the proposed solution, as well as its limitations.

The remainder of this chapter is structured as follows. Section 1 first introduces the CT-GAN attack that is capable of real-time tampering of CT images. This section furthermore details the significance of the attack and motivates why further research into potential mitigations is meaningful. Section 2 is the final section of this chapter and describes the remaining structure of the thesis.

1.1 Tampered Images in the Medical Domain

Recently, researchers managed to enter a hospital and covertly deploy a microcontroller containing a malicious deep learning application [63]. This application, dubbed CT-GAN, intercepted CT images on the network of the hospital, and applied impactful modifications to these images. Namely, the

application selectively removed and injected malign cancer nodules from passing CT images. An example is presented in Figure 1.1, where the red rectangle highlights the injection of a malign nodule. Modifications such as the one presented in the figure went largely unnoticed and resulted in misdiagnoses by the medical staff, who assess these images to detect potential illness in patients. However, perhaps most notable is the manner in which the attack was executed. Specifically, CT-GAN employed a generative adversarial network (see section 2.1.3) to tamper with the CT images completely autonomously and in real-time. Paired with the high quality and success rate of the modifications it has made this attack particularly challenging to detect once it has been successfully deployed.

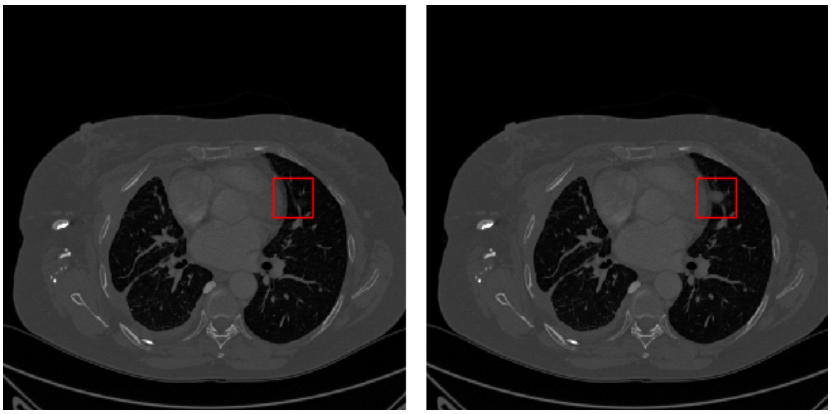


Figure 1.1: A successful attack by the CT-GAN. Left: the original CT image. Right: the tampered version with an injected nodule.

1.1.1 Significance and the Motivation for Countermeasures

The CT-GAN attack also highlighted several major security flaws that are often present in the ICT infrastructure of hospitals. Network communication shows to be often unencrypted, and medical images are rarely digitally signed to facilitate the painless exchange of imagery. Indeed, it is this lack of implemented security controls which enabled the attackers to deploy state-of-the-art technology on the hospital network, and tamper with the images.

Although tampering with (digital) imagery is by no means a novel concept by itself, it is, however, a threat that remains largely unrecognized in the medical domain. The apparent inability to ensure image integrity does not stem from a lack of solutions. Most if not all systems in hospitals have been offering cryptographic tools such as digital signatures for years, yet remain largely unused due to the complexity and risks involved during the implementation. Specifically, the implementation of these security controls remains challenging as the systems are vast, and prolonged downtime of certain critical systems may even risk the loss of human life.

However, security violations may also have significant consequences themselves. Next to the potential reputation damage [68], attacks such as the CT-GAN attack may also themselves affect the patients' well-being. Specifically, the deception may lead to a series of incorrect diagnoses and consequently delay crucial treatment or cause entirely misguided treatment. Medical staff is also likely to be less vigilant when assessing potentially tampered medical images as opposed to images in domains such as social media or internet forums where fabricated content is commonplace. Nevertheless, the small amount of incidents related to medical image tampering does not reduce the potential impact of a successful attack. Finally, given the autonomous manner in which the CT-GAN attack operates, the

attack may continue to go unnoticed for a prolonged amount of time. The current lack of awareness, combined with the potential impact of the CT-GAN attack makes it valuable to further investigate the threat and explore existing, as well as innovative, countermeasures.

1.2 Contributions

The previous section highlighted the current lack of implemented security controls within hospitals. Although highly effective countermeasures against image tampering (integrity) attacks exist, they are rarely put into practice due to their complex implementation. This thesis proposes a solution which harnesses distinct features, already embedded in all legitimate CT images, to detect image tampering performed by generative adversarial networks. By solely utilizing information already present within CT images, the dependency on existing infrastructure is largely eliminated, which consequently facilitates a painless implementation. Figure 1.2 illustrates the potential deployment of the solution within a hospital. As can be seen, the solution is activated during the diagnosis of a patient by a medical professional, and solely requires the CT images as its input. The solution subsequently extracts distinct features, the device fingerprint, from the images to discover potential tampering attempts. As the solution may be deployed locally on the workstation of the medical professional, an attacker will also unlikely be able to bypass or interfere with the detection.

However, the solution is largely dependent on the presence of distinct features embedded within legitimate CT images. As such, this thesis sets out to:

- Explore the existence of potential device fingerprints within CT images, embedded by the CT scanner which produced the image.
- Investigate the ability to distinguish CT scanner devices, based on CT images and the embedded fingerprint.
- Investigate if these device fingerprints may be harnessed to detect image tampering attacks.

This exploratory study was accomplished by performing and evaluating multiple experiments which lead to novel insights and contributions. This is, to the best of our knowledge, the first work to:

- Use a deep learning classifier to perform CT scanner classification based on CT images. (Chapter 6)
- Investigate the potential benefit of extracting a device fingerprint from CT images to perform classification of CT scanner devices. (Chapter 6)
- Propose and evaluate a solution to detect CT images tampered by a generative adversarial network. (Chapter 3 and 7)
- Investigate the potential benefit of harnessing device fingerprints to reliably detect tampered CT scans. (Chapter 7)

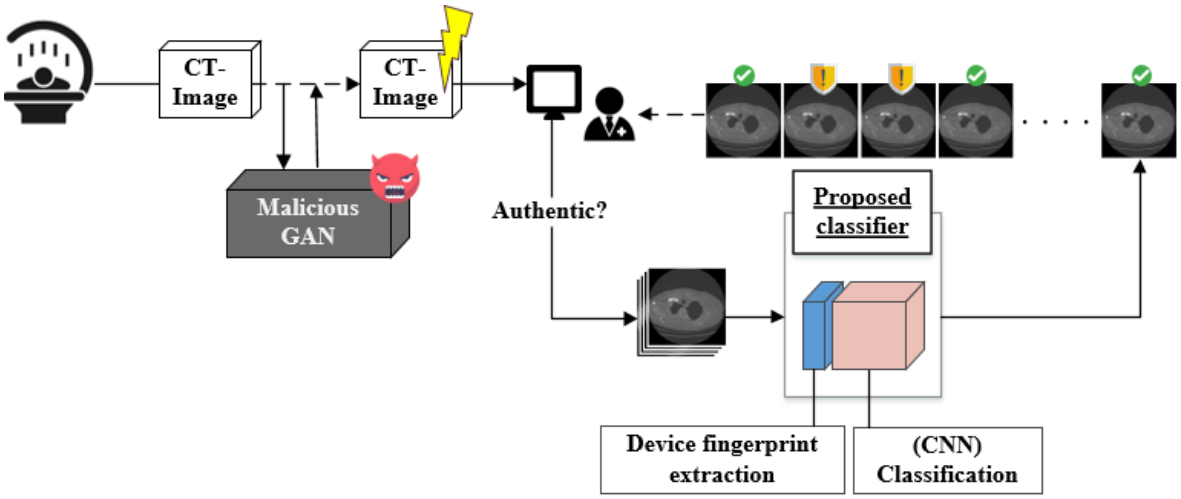


Figure 1.2: The proposed solution to support the detection of tampered images.

1.3 Remaining Structure of this Thesis

The remainder of this thesis is structured as follows. Chapter 2 first surveys the fields of security and deep learning, and provides several examples where these two domains interact with each other. The subsequent chapters then focus on the CT-GAN attack, and detail the research and experiments that have been conducted during this project. Correspondingly, chapter 3 formalizes the problem statement relevant to the CT-GAN attack and proposes a novel solution. The chapter furthermore discusses the related work relevant to the solution, and discusses a potential attacker model. Chapter 4 then introduces several research questions that will help to build and evaluate the proposed solution. Chapter 5 describes the experimental setup and considerations that apply throughout all experiments conducted during the project. Chapter 6 and 7 subsequently detail the experiments, their results, and provide answers to the research questions. Finally, chapter 8 discusses the overall results and concludes the thesis.

Chapter 2

Security and Deep Learning Survey

In recent years, deep learning has garnered a large amount of positive attention from researchers across various fields of expertise. However, in the domain of information security, deep learning has also raised unprecedented concerns. Although deep learning in this domain has also primarily been applied to improve existing tasks, deep learning models themselves have also been found vulnerable to abuse. This chapter surveys the fields of deep learning and security, and covers both the positive contributions as well as current vulnerabilities. Section 2.1 provides a brief and high-level introduction to deep learning, and covers topics that remain relevant throughout the remainder of the thesis. Section 2.2 then highlights several contributions of deep learning within the security domain. Finally, section 2.3 covers the most prominent vulnerabilities and abuse cases of deep learning models.

2.1 Deep Learning

This section will cover some of the very fundamentals of deep learning, on a high level. For a significantly more thorough read, which also provides vital information on learning rate, optimization and regularization, the reader is encouraged to read the work of Ian Goodfellow [30]. Furthermore, Jürgen Schmidhuber provides a primarily historic overview of all notable advancements within deep learning [85].

Deep learning is an approach which is part of the greater paradigm of machine learning. The machine learning methods perform a certain task, which based on given input x, produce some desired output y. One such task is classification, in which input, such as an image, is assigned a single label. For such a classification task, the model is trained on labelled samples, to then predict the label of unlabelled samples. This approach, in which the model is trained on labelled data, is called supervised learning. Conversely, clustering is a task on unlabelled data in which a machine learning model maps input samples which share certain features, closer together on a predetermined output space. Training on unlabelled data is an example of unsupervised learning.

The namesake of deep learning models is related to the learning approach, which consequently contributes to the often improved performance over traditional machine learning methods. Specifically, deep learning models do not compute the final output based on the immediate input, but instead possess additional, intermediate, hidden layers. These hidden layers, which collectively determine the depth of the complete model, extract increasingly abstract information before the model computes its final output. Most commonly, the output of a hidden layer serves as the input of its immediate

successive layer. This setup creates a chain of connected layers which may collectively form a fully connected, feed-forward network (Figure 2.1).

Each layer within such a feed-forward network consists of individual units, called neurons, which take the collective output of a previous layer and process it as its input. A non-linear activation function, such as the sigmoid or tanh function, is then subsequently applied on the processed input to determine the output value of the neuron. As such, the successive layers of neurons gradually increase the abstraction of the original input, and consequently cause the model to create its own feature representation from the original input data. This abstraction partially eliminates the necessity of manual feature-engineering that traditional machine learning models require.

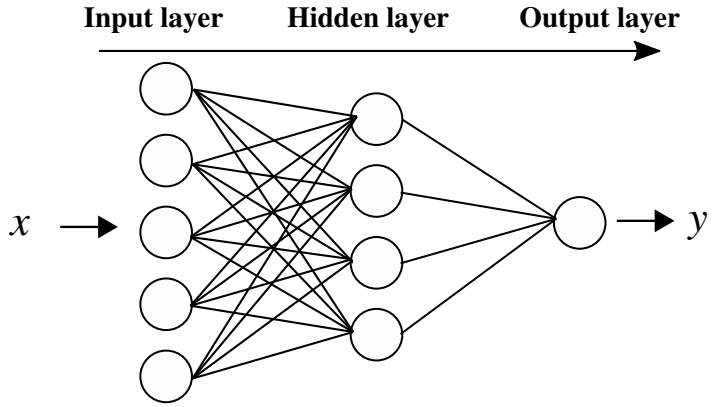


Figure 2.1: A Feed-forward neural network with a single hidden layer.

2.1.1 Training a Neural Network

In order for the neural network to adequately perform on a given task, the network requires a function which reflects the overall performance of the model. This loss function computes a certain error term for the model, and minimization of this term should lead to desirable performance of the model. Binary cross-entropy and categorical cross-entropy are popular loss functions for binary and multiclass classification tasks, respectively.

In order to minimize the loss, the model adjusts the weight of each neuron during a procedure called back-propagation [82], in which the value of the loss is propagated backwards from the final layer of the network, to the first. Most commonly, the values of the weights themselves are then adjusted by algorithms based on the Stochastic Gradient Descent. This procedure is then repeated at certain intervals until the loss of the model ideally converges to a local or global minimum.

2.1.2 Convolutional Neural Networks

In addition to the traditional feed-forward network which consists of dense layers of fully connected neurons, alternative types of networks also exist. One such network is the Convolutional Neural Network (CNN) [51] which often incorporates fully connected layers, alongside additional convolutional and pooling layers. These networks perform well on data which is loosely spatially correlated, such as images. Specifically, the convolutional layers contain a rectangular kernel which is slid over each feature of the input. In the case of images, the kernel slides over each pixel of the image, and analyzes the pixel as well as its local neighbours to determine the activation value of the central pixel in

the kernel. After all features have been processed, the convolutional layer outputs an activation map which contains the activation value for each pixel. Pooling layers, such as max pooling, take small, local subsets of their input, such as the activation map of a convolutional layer, and pool this subset to a single value. The use of pooling layers subsequently reduces the dimensionality of its input. In a complete network, a common approach is to apply multiple convolutional layers, followed by a single pooling layer. This sequence of layers is then stacked multiple times, and finally supplemented with a fully connected layer to complete the network.

2.1.3 Generative Adversarial Networks

Although deep neural networks are perhaps predominantly known for making predictions on existing data, Goodfellow et al. [32] proposed an innovative network architecture capable of generating its owns data samples which show a striking similarity with samples from the original dataset. This Generative Adversarial Network (GAN) is composed of two models, a generator and discriminator, pitted against each other as illustrated in Figure 2.2. The generator transforms a (random) feature vector z and sends the modified sample to the discriminator. The discriminator trains on batches of samples which contain data from an original dataset, and samples produced by the generator. The discriminator is tasked to distinguish between the original, real samples, and the generated, fake samples produced by the generator. The result of the classification is propagated back to both networks to improve the discrimination, as well as the generative task.

A common analogy is to consider the generator as a counterfeiter whose desire is to produce fake currency undetected, while the discriminator acts as the police whose goal is to distinguish between real and fake currency. Both networks are adversaries of each other and trained simultaneously. During training, the generator will gradually improve the quality of its samples, while the discriminator gradually becomes more adept in distinguishing generated samples from samples belonging to the original dataset. Ideally, after the training process is completed, the generated samples will share great similarity with the original training data.

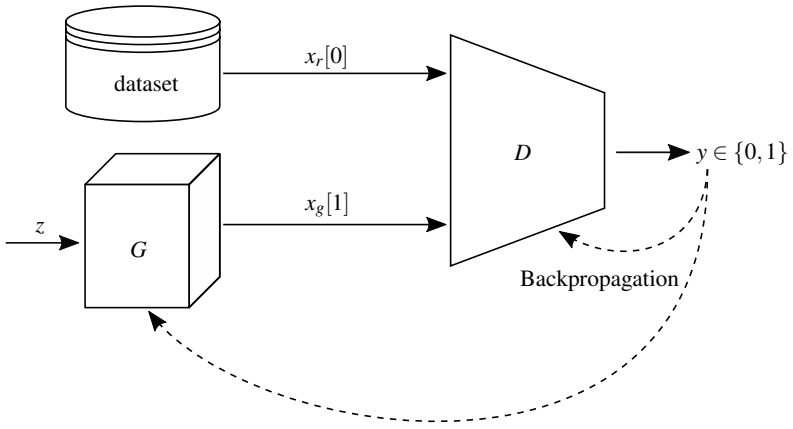


Figure 2.2: The setup of a Generative Adversarial Network.

Since the initial success of GANs, the field of generative models has developed substantially and numerous variations have been proposed. One notable example is the Conditional Generative Adversarial Net (CGAN) [64] in which additional auxiliary input is fed to the network. This auxiliary input

is used to condition the network in order to steer the generated output. This approach has been applied with great success for Image-to-Image translation tasks [105], including inpainting [39]. In the case of an inpainting task, the auxiliary input is an incomplete version of an image, and the network is tasked to fill in the missing section.

Generative models are most commonly deployed to augment training data in domains where data samples are often scarce, such as the medical domain. The augmentation with realistic samples leads to more training data and consequently improves the performance of existing task such as liver lesion classification [27] or lung nodule segmentation [41]. However, the rapidly improving quality of the generated samples have also raised concerns about the potential abuse of these generative models. For example, in the security domain, generative models have been employed to adapt malware in order to slip past Intrusion Prevention and Detection Systems [35, 77]. Furthermore, in the imaging and computer vision domain, Karras et al. [44] have managed to generate images of impressive realism, even garnering media attention [58, 96]. In addition to images, synthetic videos, often referred to as Deepfakes [69], have also garnered the attention of researchers and media alike. These videos often feature important political figures placed in disingenuous situations, or giving deceptive speeches [88, 2].

2.1.4 Deep Learning in Healthcare

The application of deep learning has also found widespread adoption within the medical domain, perhaps most commonly employed to offer support in various decision making. For instance, information from electronic health records may be fed into a stack of denoising autoencoders in order to learn representations which may then be used to predict future illness or diseases in patients [61]. Furthermore, numerous tasks for various medical imagery, which often utilize convolutional networks, have also been highly successful. For example, Esteva et al. [24] fine-tuned a pre-trained Google Inception V3 model [89] to classify various types of skin lesions in order to support the early detection of skin cancer. Besides these classification tasks, Ronneberger et al. [80] introduced a U-net architecture for convolutional networks to localize and segment neuronal structures in Electron Microscope (EM) images. Segmentation and localization tasks differ from classification tasks as the latter may merely conclude if a certain object is present (e.g. malign cancer nodules), while localization and segmentation tasks aim to highlight specific objects within a larger image.

Despite its overall success, the performance of a deep learning model is largely dependent on the amount and quality of its training data. Traditionally, small datasets are augmented by performing certain transformations such as rotation, mirroring and disposition on each samples in order to capture a larger variation of samples, but also increase the volume of the training set. However, the amount of effective transformations until the augmented samples are a near-duplicate of the original, is limited. Fortunately, the recent advancements in Generated Adversarial Networks [32] have introduced a new avenue to augment datasets. Generative networks have been applied to generate synthetic samples of medical images to boost the performance of e.g. a liver lesion classification task [27] or lung nodule segmentation [41].

The aforementioned examples provide merely a small glimpse into the vast application of deep learning within the medical domain. The reader is encouraged to look into the work of Thaler and Menkovski [91], and Miotto et al. [62] for more extensive surveys on the adoption of deep learning within healthcare.

2.2 Applying Deep Learning to Solve Security Tasks

The accomplishments achieved by machine- and deep learning models have, unsurprisingly, also gained notable attention within the domain of information security. In this domain, tasks commonly involve distinguishing malicious from acceptable, innocent behaviour. Separating phishing e-mails from legitimate ones, or identifying malware among ordinary binaries may all be viewed as (binary) classification tasks. Since the traditional decision-making within the information security domain is already predominantly based on digital data, employing deep learning to perhaps improve the existing tasks is a logical step. Although deep learning has indeed been applied with great success, the complexity of neural networks often makes it a challenge to rationalize the decision of the neural network. This lack of interpretability has even lead to a research field of its own [60], and is one of the reasons that some systems still favour knowledge-based solutions over a deep learning approach.

This section will consider a relatively small subset of applicable areas within information security. For a more extensive overview, the reader is encouraged to read the survey by Thaler et al. [92] which also approaches the applicable areas from a data-driven perspective.

2.2.1 Network Intrusion and Anomaly Detection

To determine whether a network has been compromised by an attacker, a common task is to analyze the traffic of the network, and identify any malicious data flows. The complexity of this task has greatly increased over the past years, as the diversity and sheer amount of network traffic has grown substantially. As a consequence, ensuring effective decision-making when identifying malicious activity often demands a vast amount of domain knowledge. The task primarily involves learning ordinary situations and behaviour, to later capture any divergence. This may include unexpected login times, or unusual data access by particular users. Although traditional machine learning has shown effectiveness, obtaining suitable features is not trivial. Deep learning improves this situation as it is able to learn suitable representations of the input data by itself, largely eliminating the necessity of manual feature engineering by domain experts.

Yin et al. [99] propose a Recurrent Neural Network (RNN) to identify various attacks such as DoS (Denial of Service) and Probing attacks. The complete dataset consists of a total of 5 classes of which the data samples contain 41 features. Their proposed classifier reaches an accuracy of up to 83.28%, outperforming traditional machine learning classifiers. Kang and Kang [43] proposed an IDS to improve the security of intra-vehicular networks, which often employ the CAN bus architecture to communicate messages across different components within a vehicle. The authors dissect a CAN packet into features to train a deep neural network to detect malicious activity on the network. The resulting classifier achieves an accuracy of up to 97.8% with false positive and false negative rates of 1.6% and 2.8%, respectively.

2.2.2 Content Classification

Next to the classification of (active) network traffic, neural networks have also been applied to classify content such as phishing e-mails or malware. To detect phishing e-mails, Basnet et al. [6] evaluate several machine learning approaches on a dataset containing $4,000$ e-mails, 973 of which are labelled as phishing e-mails. Among other sources, the authors combine information from the domain of the sender, and the HTML source of the e-mail to extract a total of 16 features. Both their proposed neural network and SVM achieved an accuracy of 97.99%,

To identify malicious Android apps available on the Google Play Store, Yuan et al. [102] combined a set of 250 apps known to be malicious, with the top 250 apps from the Google Play Store, assumed to be innocent, to acquire a total dataset containing 500 apps . Their proposed deep learning classifier achieves 96.5% accuracy and outperforms their second-best classifier, a SVM, by nearly 15%. In 2015, Microsoft also announced a Kaggle malware classification challenge, providing a dataset with over $20,000$ malware samples which still serves as a common benchmark [79].

2.2.3 Digital Forensics

In the field of media forensics, deep learning has been embraced for tasks related to steganalysis, watermarking and forgery localization. Qian et al. [75] were among the first to experiment how well a customized CNN is able to capture features that are caused by steganography traces. In their attempt, the authors were able to achieve comparable performance of state-of-the-art solutions which still applied extensive feature engineering. Bappy et al. [5] apply a recurrent neural network (RNN) to detect local image manipulation such as the copy-clone manipulation. Differently than local manipulation, Bayar and Stamm [7] proposed a CNN architecture to detect global manipulations such as JPEG compression or Guassian Blur. Their model manages to outperform traditional state-of-the-art, with an overall accuracy of 99.66%.

As the field of digital forensics continues to rapidly expand its application of deep learning techniques, the interested reader is encouraged to read the survey by Meng et al. [57], containing numerous examples of deep learning solutions for tasks related to digital watermarking and other information hiding techniques [104]. An extensive survey by Zheng et al. furthermore describes numerous manipulation techniques, detection techniques and datasets used in the realm of forensics.

2.3 Attacks Against Deep Learning Models

Despite their positive contribution in solving security tasks, deep learning models may also serve as an attack vector themselves. Training a neural network often demands considerable resources; research time, computational power, and even access to exclusive training data are only few among many cost factors. Certain models are therefore considered Intellectual Property (IP), often forming a vital part in certain business models such as Machine Learning as a Service (MLaaS). Consequently, it is often in a business' best interest to keep the inner workings of their models confidential, ensure correct behavior and integrity of the model, while also exposing it to a broader audience to benefit from its predictions (availability).

Besides protecting a model itself in the traditional sense of information security (Confidentiality, Integrity and Availability), it is often equally vital that a model does not leak critical information related to its training data and potentially jeopardize the privacy of its data subjects. Finally, deep learning may also be employed by malicious actors themselves, posing a powerful threat to honest parties.

This section explores the most common avenues in which deep learning models are attacked, or abused by attackers to aid in malicious activity. Even though most of these attacks apply to both deep learning and traditional machine learning, certain attacks are more effective, or practical, against a particular type of learning algorithm. For the sake of completeness, this survey will also consider attacks that predominantly threaten traditional machine learning models and have only found limited success against deep learning.

2.3.1 Attacker Models

When considering attacks against deep learning (or machine learning) models, an attacker's capabilities or knowledge of the model under attack, are often expressed in terms of white-box and black-box capability. In a strict black-box setting, an attacker may feed a model with their own data, but is merely able to observe the output of the model. Besides the accepted input of the model, its output, and an optional confidence value for prediction tasks, no information regarding the model itself is provided to the attacker. However, some researchers consider an attacker's knowledge of the underlying learning method of the model, e.g. neural net, random forest [8], or logistic regression, also part of the black-box attacker model.

White-box or gray-box models are subject to significantly more variance and are mostly dependent on the attacker's objective. Some attackers assume knowledge of a model's exact architecture, while other attackers also assume knowledge of hyperparameters or even the values of a neural network's entire set of weights. Differently, some attacks are primarily based on knowledge of the training procedure such as the dataset used during training, or the amount of data on which a model has been trained. Nevertheless, a white-box attacker model has always more knowledge and capability than the less powerful black-box attacker. However, an effective attack in a black-box setting poses a greater threat than an equally effective attack in a gray-box or white-box setting.

2.3.2 Model Extraction

In a model extraction attack (Figure 2.3), an attacker repeatedly queries a victim's machine learning model and exploits the model's output to produce a near-copy of the victim's model. The attack, sometimes referred to as 'model stealing', was first introduced by Tramèr et al. [95] in which the authors attack Machine Learning as a Service (MLaaS) services to mimic a high-performing model for their own gain, requiring less resources than training an entirely new model on their own. Once a model is successfully copied, the attacker is able to query her own model instead unrestrictedly, avoiding potential query costs or exploit the copy to craft further attacks such as an adversarial attack (section 2.3.4).

The initial attack by Tramèr et al. primarily executed the attack using equation-solving, which exploits models such as logistic regression that directly compute class probabilities based on the given input. When the underlying learning algorithm is known and the class probabilities are given as output after each query, the attack becomes almost trivial to execute. Since the attack relies on additional information, namely knowledge of the underlying algorithm and the class probabilities for the queried samples, it may not be considered a strict black-box attack. Nevertheless, as the required pieces of information are often provided by the query service, the attack still has practical application.

In the same study, the authors explore further application of model extraction techniques, devising attacks against decision tree models and even multilayer perceptrons. Although the attack was successful against neural networks, it required considerably more queries; for a near-copy of a small, multilayer perceptron ($2,225$ unknown parameters) it took around $11,000$ queries to extract, while the extraction of a logistic regression model for the same dataset was completed within $1,500$ queries. As the attack primarily relies on confidence values of the prediction, omitting this information from a model's output would significantly reduce the attack's effectiveness.

In a similar attack, Papernot et al. [72] aimed to train a substitute model with similar behaviour as their victim's model by adjusting the substitute model based on the victim's classifications of pur-

posefully crafted synthetic training samples. Although the main objective of the attack was to create highly effective adversarial examples, the proposed training process could be completed in as few as 2.000 queries to achieve similar behaviour than the victim's model. Although the substitute showed transferable behaviour regarding adversarial examples, its accuracy on ordinary samples remained significantly worse than its victim's accuracy.

More recently, Juuti et al. [42] presented an attack that is considerably more practical in comparison to the original model stealing attack [95]. In their proposed attack, taking inspiration from the attack by Papernot et al. [72], the authors were able to extract a deep neural network containing almost $500,000$ parameters, in just over $100,000$ queries, with a test-agreement of nearly 98%. However, the attack described by Juuti et al. requires significantly more knowledge of the victim's underlying model, specifically the entire architecture of the model.

Juuti et al. further propose a defensive mechanism to detect model extraction based on the distribution of the samples used in queries during the extraction process. Their detector was able to discover all previously discussed model extraction techniques [95, 72, 42] with little false-positive-rates (up to 0.6%). Finally, another mechanism to potentially mitigate the impact of a model extraction attack is to embed specific information or behaviour into the neural network, serving as a watermark. Proposed solutions include the modification of layer functions within the neural network [81] or training the network on carefully crafted samples [103]. To verify ownership of a potentially stolen model, both proposals involve querying the model in question on specific input and observing the model's responses and verify whether they match the unique responses given by the legitimate, original model. However, both proposals require at least black-box access to the investigated model, which may diminish the practicality of the solutions as an attacker may still use the model in secret without any hindrance.

In conclusion, the attack model extraction attack may certainly be abused in practice. However, in its current state, it may be predominantly abused as a stepping stone for a more advanced attack [72]. Specifically, extracting a near copy of a complex neural network still requires considerable knowledge of the target network. Nevertheless, traditional machine learning algorithms remain vulnerable to model extraction.

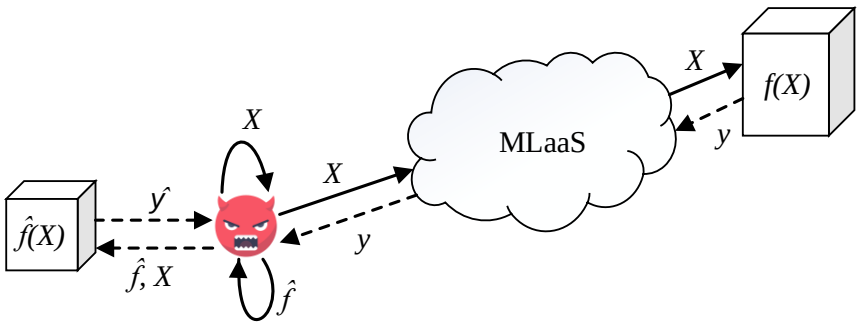


Figure 2.3: A simple example of a model extraction attack.

2.3.3 Model Inversion

Another threat to ML models is the leakage of information related to the training data used by the model. These attacks are often referred to as model inversion attacks, or membership inference attacks if the objective is to test whether a certain data sample was part of a model's training data [86, 26]. These attacks are especially threatening in situations where training data should remain confidential or contains sensitive information. In one attack, the authors were able to extract sensitive features from training data and could reasonably reconstruct faces from a facial recognition classifier [25].

Rounding, or completely omitting additional values such as confidence values from a model's output often are among the most common countermeasures to reduce the effectiveness of an attack. Differential privacy [23], an approach which adds a certain amount of noise to a data source, is also considered a strong candidate in thwarting these specific attacks. In an effort to apply differential privacy to the training data of deep neural networks, Abadi et al. [1] propose a modified, differential private version of the stochastic gradient descent algorithm (SGD). Additionally, they introduce an algorithm which is able to compute and track the privacy loss during training. Although the the addition of differential privacy comes at the cost of accuracy, their model trained on MNIST drops as little as 1.3% accuracy when differential privacy is applied.

2.3.4 Adversarial Examples

Adversarial attacks are perhaps the most well-studied form of attack and aim to undermine the integrity of deep learning models. The most common application of the attack is to target deep learning models which are tasked to classify given input samples. In the attack, the attacker picks an input sample from an arbitrary class A, which is correctly classified by the model to indeed stem from class A. The attacker then and adds small perturbations to the sample such that the model now classifies, often with high confidence, the sample as class B. This carefully added perturbation, leading to a so-called adversarial example, is often imperceptible by humans, adding to the curiosity of the attack. Figure 2.4 demonstrates how the imperceptible perturbation leads to drastic misclassification.

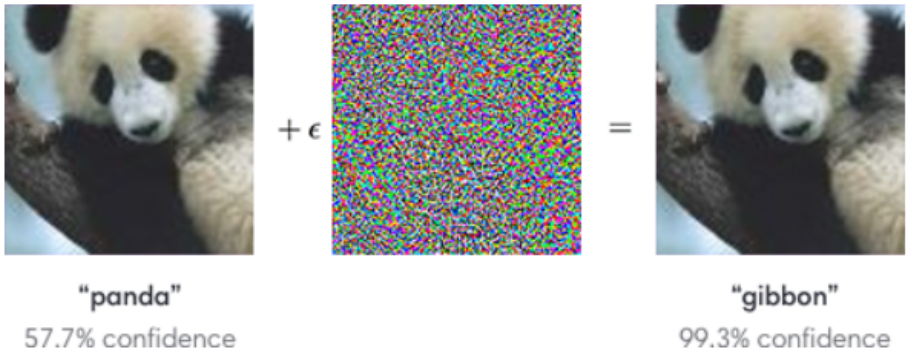


Figure 2.4: A popular example of an adversarial attack. Source: OpenAI [31]

.

Adversarial examples first gained notable attention after Szegedy et al. [90] found that deep neural networks contain certain blind spots that may induce misclassification, though the authors could not fully explain it at the time. Following the original discovery, Goodfellow et al. [29] attributed the vulnerability to an apparent linearity present even in non-linear models, such as neural networks.

More recent work by Ilyas et al. [38] suggests that the vulnerability to adversarial examples may be attributed to the existence of non-robust features within images.

Moreover, [29] introduced the "fast gradient sign method" to consistently and efficiently craft adversarial examples. Although the fast gradient sign method is considered to be one of least computationally expensive algorithms to craft adversarial examples, Papernot et al. [73] proposed a method exploiting a Jacobian Matrix, which requires more computations, yet adds fewer perturbations to the original sample. The fewer amount of perturbation make this enhanced attack more difficult to detect. More recently, Su et al. [87] proposed an algorithm in which merely a single pixel is modified to trigger a misclassification by an underlying model.

Besides the popular exploitation in computer vision, the presence of adversarial examples is not limited to image classification tasks, and in fact affects any domain employing deep learning models. Carlini et al. [11] managed to trigger specific commands by the voice assistant of Google ("OK Google") while the voice signal remained unrecognizable by humans. Kurakin et al. [50] furthermore showed that photographs of perturbed images also lead to misclassification, demonstrating the robustness of adversarial examples even when transferred to a different medium.

In addition to the inherit vulnerability to adversarial examples which all deep learning models posses, certain adversarial examples also transfer across entirely different models [94, 66]. I.e. examples crafted to fool model *A*, may also have similar effect on a different model *B*. In the seminal work of Papernot et al. [72], the authors demonstrate the effectiveness of adversarial attacks when merely a weak, black-box attacker model is considered. The authors first commence with a model extraction technique (section 2.3.2) to obtain a substitute for the victim's model. After obtaining a substitute which sufficiently mimics the original model of the victim, adversarial examples are crafted for the substitute model. The adversarial examples which are proven to be effective on the substitute are then consequently used on the original model. The attack proved to be highly effective, with misclassification rates of over 97% by the victim's original model.

Countermeasures

Given that the existence of adversarial examples is directly related to the core behaviour of neural networks, finding an effective countermeasure remains an open research problem. Initially proposed countermeasures such as improved gradient masking [93] or defensive distillation [74] have been surpassed by stronger adversarial attacks such as [12]. Thus far, the most consistent countermeasure to adversarial examples remains to engage in adversarial training in which samples from several attacks are used to train the network on these perturbations [29]. To facilitate the adversarial training, Papernot et al. [71] have created the library cleverhans[1] to train models on the most common attacks.

However, the field of adversarial attacks proceeds to evolve, with new attacks and possible mitigations being proposed regularly. In an effort to improve the quality of new defense proposals, Carlini et al. [10] have published a paper which may serve as a guideline to evaluate potential defensive techniques. Finally, given the substantial amount of literature available on this particular topic, the interested reader is encouraged to take a look at surveys which exclusively focus on the threat and current state-of-the-art of adversarial examples and potential countermeasures [3, 101].

[1] `https://www.cleverhans.io/`

2.3.5 Malicious Application of Generative Adversarial Networks

With a similarly deceptive goal as adversarial examples, generative adversarial networks (GAN) have also been applied to support in malicious activity. In order to bypass a malware detector, Hu and Tan [36] set up a GAN architecture and fit their discriminator to the detector under attack, starting out with an existing dataset of both benign and malware samples. Once the discriminator has become a suitable substitute for the detector under attack, the authors pit the generator and discriminator against each other to create undetectable malware samples. After sufficient training, the samples provided by the generator are then directly fed into the victim's detector. Their highly effective attack reduced the true-positive rate of the detector to nearly 0%, i.e. no malicious sample could be identified as such. Similarly, Rigaki and Garcia [78] have shown to dynamically modify network traffic to avoid detection by an Intrusion Prevention System (IPS). Although both applications consider a black-box attacker model, it is important to note that these applications still require direct feedback from the victim's detector as to whether or not they have been detected, which may be an unrealistic assumption in practice.

Within the setting of collaborative learning, a GAN has been employed to target a specific user and generate samples which greatly resemble the victim's private dataset [34]. Here the attackers abuse the shared, public model as its discriminator, while their constructed generator aims to produce samples resembling the victim's class A. The synthetic samples are then injected into the training process of the shared model, but under a false label B, causing the shared model to misclassify samples of class A as class B. As a consequence, when the victim attempts to improve the accuracy of the shared model, she unwittingly improves the acting discriminator of the attackers. This cycle repeats until the attacker's generator is able to produce samples which closely mimic the training data of the victim, potentially leaking sensitive information.

Recently, within the healthcare sector, Mirsky et al. [63] have constructed a CGAN capable of injecting and removing cancer nodules from Computed Tomography (CT) images. In their attack, the researchers managed to infiltrate a hospital and autonomously intercept and tamper with the contents of valid CT imagery. In the large majority of the cases, the malicious modifications achieved by the CGAN even managed to fool the medical staff of the hospital.

As the capability and significance of the attack was both demonstrated in practice, and thoroughly detailed in its publication, the attack provides an interesting topic of research. In addition, the CT-GAN attack has emerged only recently, and as chapter 3 will highlight, no realistic countermeasures have been proposed thus far. This is consequently the main motivation this specific attack has been chosen as the research topic for this thesis. As such, this attack will be discussed in great detail in the upcoming chapter 3 and furthermore forms the basis for the remainder of this thesis.

Chapter 3

Problem Exploration

This chapter elaborates further on the significance of the CT-GAN attack [63] and formulates the problem statement along with a potential solution. In addition, this chapter highlights the related work, current state-of-the-art and open challenges from the forensic and medical fields. This chapter is structured as follows. Section 3.1 reintroduces the CT-GAN attack and formulates the problem statement. Section 3.2 then focuses on the related work and highlights the relationship between the different fields of research. Section 3.3 subsequently applies the foundation of the related work to propose a novel solution to the problem. Finally, section 3.4 discusses the attacker model and elaborates on the potential deployment of the solution within the environment of a hospital.

3.1 Problem Statement

Section 2.1.3 of the previous chapter introduced the concept of Generative Adversarial Networks (GANs). Namely, a GAN consist of two individual networks that compete against each other. One network, the generator, produces samples resembling the data from an existing training set while the second network, the discriminator, aims to distinguish the generated data from the original samples. After training, the discriminator is often discarded while the generator is kept in order to benefit from its high quality image generation. The value of synthetic image generation has also been recognized in the medical field where GANs are often leveraged in numerous solutions which aim to augment the often scarce training data. The resulting, bigger dataset is then consequently applied to improve the performance of existing tasks such as lung nodule segmentation [41].

Although these solutions are developed with honest intentions, a recently emerged application of GANs seeks to tamper with existing medical images in order to mislead medical staff and deep learning classifiers alike. The attack pipeline introduced in [63] intercepts legitimate CT scans on the network of a hospital, and alters the image by selectively adding or removing malign lung nodules from the original images. These alterations were conducted completely autonomously and produced sufficiently realistic results such that it did not raise suspicion of the professional medical staff examining the tampered imagery.

The current infrastructure of most hospitals is ill-equipped to detect and mitigate any attacks that attempt to tamper with medical images during either transit or storage. Although numerous countermeasures, such as digital signatures, have already been standardized for years, the absence of actual

implementations indicates that the vulnerable state of hospitals is unlikely to change in the near future. In addition to the current lack of defences implemented in hospitals, the continuous evolution of GANs introduces an additional challenge. Any proposed countermeasures should remain effective as new type of GANs emerge and existing networks continue to improve, as an obligation to continuously update or replace existing countermeasures is infeasible within the fragile environment of a hospital. One potential avenue for such a solution is to explore hidden, distinct elements which are already present within all legitimate CT images. Since the primary purpose of a GAN is to produce realistic samples, we assume that the GAN will be unable to accurately recreate such elements, and inadvertently cause distortions during the tampering process. This assumption is furthermore motivated in section 3.4.3. Since the exact manifestation of the distortions caused by the GAN would be irrelevant for the detection, these hidden elements may not only exclusively serve as a warning system for a particular attack, but may instead be applied to detect a variety of different tampering attacks. The existence and potential of such hidden elements will be explored in section 3.2.2.

To summarize, an effective countermeasure against the CT-GAN attack should not be solely tailored for this particular attack, but instead should exhibit a certain resilience against different variations as well. Furthermore, the countermeasure should act independently from the existing hospital infrastructure. This thesis therefore aims to find reliable and isolated measures to detect any tampering attempts in CT scans that are conducted by exploiting generative adversarial networks.

3.2 Background and Related Work

Although the healthcare sector remains slow to adapt to new threats and attacks, numerous measures already exist to improve the ICT infrastructure security of hospitals. Nevertheless, detecting fabricated images produced by a GAN is still an active field of research with several open challenges. Only recently have studies related to image forensics begun to consider GANs as a real threat. This section presents the current state-of-the-art to detect generated imagery, and explores specific traditional forensic techniques that have shown the potential to deal with generated samples as well.

3.2.1 Detection of Generated Samples

The rapid improvement of achieved realism by GANs urges the necessity of reliable detection techniques in order to mitigate potential misuse. Although the field of image forensics contains substantial work on traditional image modification such as compression and copy-move manipulation [5, 7], the detection of GANs has only recently been gaining more attention. To detect the realistic images generated by [44], Mo et al. [65] proposed a highly accurate classifier aimed to distinguish real from generated images. Similarly, Jain et al. [40] split images into individual patches and combine a CNN with a SVM to evaluate the authenticity of individual patches in order to determine whether the entire image is authentic or fabricated.

Although these approaches achieve high accuracies of up to 99%, generalizing the detection of GAN-generated samples introduces additional challenges. Indeed, accurately detecting existing samples produced by generator A does not imply high accuracy when encountering newer samples produced by an improved version of A, or images produced by an entirely different generator B. The first challenge herein lies in the continuous improvement of the quality of generated samples, as the fundamental goal of the generator is to learn how to fool a discriminator. The second challenge emanates from the large variety of GANs, where alternative methodologies and network architectures may produce

similar realistic output, but leave behind different artefacts utilized to distinguish the samples. Thus, classification that is exclusively achieved on known samples has a particularly narrow scope and is generally not applicable in practice.

Li et al. [53] studied the effect of classifying samples from improved or completely unseen GANs than what was seen during the training phase of a classifier. The researchers measured the effect when a classifier is tested on samples from a single generator with varying training parameters. The results demonstrate that accuracies drop when a classifier is faced with a generator from a different epoch than the generator which produced the training data for the classifier. In addition, the researchers show that the classification accuracy may drop from 98% to 61% if the classifier is faced with an entirely different type of generator during test time (e.g. WGAN vs DCGAN). Furthermore, Xuan et al. [97] also acknowledge this difficulty and report a drop in accuracy from 95% to 68% when a classifier is suddenly faced with generated images from an unseen GAN architecture. These contributions further indicate that the effective scope of a high performing classifier may be narrow, impeding its longevity.

In contrast, [19, 67] report promising results. In [67], co-occurence matrices based on the RGB channels of images are used to preprocess data before it is classified by a CNN. The authors report an accuracy of up to 99.49% when the classifier is trained on samples from a cycleGAN, but tested on a dataset generated by StarGAN. The solution presented in [19] demonstrates that an initial train-test mismatch between generated samples could be largely mitigated by retraining on just a few samples from the unseen architecture. For instance, the presented solution is able to improve its accuracy from 50% to 70% by retraining on merely 2 samples. However, completely new and unseen samples remain a challenge.

3.2.2 Noise Patterns for (Device) Fingerprints and Forgery Detection

To advance the field of forensics, Lukáš et al. [54] leveraged the Photo Response Non Uniformity noise (PRNU) present in the Sensor Pattern Noise (SPN) of photo cameras, in order to attribute images to specific camera models. To extract the noise component of an image, the approach uses a denoising filter based on the work of [59]. When the filter is applied to several images from the same, known camera model, the noise components can be averaged to obtain a reference pattern for a specific camera, which then serves as the fingerprint of the device. In order to attribute future images from an unknown source, the new images will be subject to the same denoising filter, and the resulting noise component compared to the previously computed reference pattern. This comparison is drawn based on the correlation between the two noise images, where a high correlation indicates the likelihood the image stems from the same camera.

Artificial Fingerprints

The seminal work of Lukáš et al. has furthermore sparked a surge of new advancements; next to a large-scale study to attribute a wide array of images to camera models, similar fingerprinting- and denoising techniques have been applied to detect forgeries within images [15, 20]. Notably, the original technique proposed in [54] has recently been applied to attribute generated images to specific GANs [55]. This recent study showed that even artificial images, not originating from a camera, leave behind consistent traces of their respective generator, granting the ability to attribute images to different GAN algorithms. In a similar manner, [100] employed an autoencoder to extract the noise pattern of images and found that even a divergence in the architecture of a GAN, or its training set, lead to distinctive fingerprints.

Device Fingerprints in the Medical Domain

The notion of device fingerprinting has also been applied in medical fields that employ the DICOM data format to store medical imagery. In an effort to minimize the reliance on the metadata of DICOM files, device fingerprints offer an alternative form of device identification in cases where the metadata is unavailable or may have been tampered with. Kharboutly et al. [46] applied the identification process from [54] to CT images and were able to reach an accuracy of up to 97% when tasked to identify three CT scanner models. In subsequent work, the researchers advanced the preprocessing steps, resulting in a perfect score when identification is based on just certain axes (X, Y, Z) of an image volume [47, 45]. In more recent work, Duan et al. [22] proposed new techniques for device identification based on Original Sensor Pattern Noise (OSPN) and the image reconstruction algorithm of a scanner. Their experiments aim to identify 15 distinct scanner models, originating from 4 different manufacturers. Their combined approach manages to achieve an accuracy of 96.65%.

3.2.3 Assessing Image Integrity

As highlighted in section 3.2.2, device fingerprints have also been applied to verify the integrity of images themselves as both local and global image modifications may leave behind visible traces in a fingerprint extracted from an image. However, In the medical domain, image forensics are less explored; Huang et al. [37] apply histogram statistics (HRBD and HRBT) along with a SVM [9] to identify global modifications (e.g. compression) within images and achieve an accuracy of up to 85% when analyzing CT images. Ghoneim et al. [28] have achieved an accuracy of 84.3% when classifying the PRNU pattern of mammograms to detect images tampered by copy-move operations. As of yet, no countermeasure considers medical images tampered by a GAN.

3.3 Proposed Solution

The successful attack on CT images [63] calls for reliable countermeasures which should aim to thwart the current attack, yet also stop potential future variations of the attack.

As presented in section 3.2, identifying generated (fake) samples from a known GAN has been done with great success [65]. However, detecting similar looking samples originating from an unknown network remains an ongoing challenge [53, 97]. Furthermore, the topic of image integrity within the medical domain is only explored to a limited extent [37]; generated fakes are not considered at all.

However, the SPN-extraction approach for camera identication demonstrated in [54] has also shown to be capable to detect forgeries within images [28]. Moreover, in the medical domain, the same extraction approach has been successfully applied to identify CT scanner devices as well [46, 22]. Finally, existing studies related to GAN attribution, i.e. determining the model behind generated samples, have also found that each model leaves behind distinct traces [55, 100]. These studies have yet again applied the same denoising technique that was initially applied to identify camera models based on PRNU. Yet, their study demonstrates that the same technique also seems effective for images which are completely synthetic and thus do not posses the physical association with a PRNU pattern.

As such, the large applicability of device fingerprints leads to the following hypothesises for a potential solution.

1. If CT scanner devices leave behind distinct fingerprints in the image generated by them, then a suitable classifier may also be able to distinguish different devices based on CT images.

2. If the device fingerprint of a CT scanner is indeed highly distinct, it may then act as a whitelist for legitimate scans. This would enable a classifier to develop strict boundaries for the fingerprint, and consequently detect any distortions. In addition, as the classifier does not pursue any specific distortions, it may be capable to detect various forgeries without any additional training.

3. GANs have also shown to embed distinct fingerprints, which are affected by the training data, in their generated samples. As such, if the GAN of an attacker is not exclusively trained on images from a single, particular scanner, the generated samples may contain a distorted fingerprint compared to the fingerprint in an image taken by a legitimate CT scanner. Moreover, it is unlikely that an attacker has sufficient data available to train a high-performing generative network to attack a single, arbitrary CT scanner, as this would require a sizeable amount of specific data. This hypothesis is further discussed and motivated in section 3.4.3.

Therefore, we propose a classifier that is deployed for a specific scanner device, and will be exclusively trained on device fingerprints of legitimate CT scans. Specifically, the classifier would be trained on CT slices from its associated device, and legitimate CT images from any other scanner device. As such, the classifier will learn a strict representation of the fingerprint of its device, while still being unaware of any potential attacks which aim to tamper with the images. Nevertheless, if a tampering attack indeed occurs after deployment, the classifier will be able to detect the distortion in the fingerprint and subsequently warn the medical professional (Figure 1.2).

3.4 The Role of the Solution Within a Hospital

The usability of the proposed classifier is not solely determined by its ability to detect a range of tampering attempts, but should also fit within the complex environment of a hospital. This section further elaborates on some of the ongoing challenges within hospitals, and discusses the objective of the solution from the perspective of this greater context. Finally, this section discusses a potential attacker's capability against the deployed solution.

3.4.1 The Ongoing Challenge of Implementing Security Controls in Hospitals

The CT-GAN attack, despite its novelty, was primarily successful due to the exploitation of known security flaws within hospitals. During the attack, the responsible researchers gained access to restricted areas, intercepted unencrypted network traffic and consequently exploited the picture archiving and communication system (PACS). Finally, the CT scans themselves did not possess any integrity controls, which enabled the final step of undetected image tampering.

If security controls would have been successfully implemented at any of theses stages, the attack would have been largely impeded or perhaps completely neutralized. However, the very existence of the attack exposes the absence of these controls. The current lack of successfully implemented security controls may be primarily attributed to the complexity and fragility of the ICT infrastructure of a hospital. Although countermeasures, such as digital signatures, have been standardized for years and are also supported by e.g. the DICOM file format[1], they remain largely unimplemented, as concrete integration into the complex environment of a hospital is far from trivial. In addition, any complications during the implementation may risk the malfunction of critical systems, which may then even

[1] `https://www.leadtools.com/help/leadtools/v20/dh/to/working-with-dicom-digital-signatures.html`

endanger the patients' lives. Therefore, hospitals will commonly accept the risks of a vulnerable system over the potential consequences which a failed implementation of a security control may carry. This dilemma consequently calls for alternative controls which mitigate certain attacks, yet remain sensible and provide a pragmatic implementation.

Besides the hospitals themselves, third parties such as device manufacturers, which provide medical devices and AI solutions to hospitals, also have a particular interest in ensuring that their technology does not get abused. For example, the malicious device which hosted the CT-GAN malware was disguised as a microcontroller from a legitimate manufacturer. Moreover, the realism of the injected nodules also mislead the AI solution that usually supports the medical staff in their diagnoses. As such, third parties were also directly affected by the security breach, which consequently has an effect on the reputation of the brand. Even though third parties such as the manufacturers of medical devices have certain influence on the practices and standardization within hospitals, the reach of actual governance is limited. Nevertheless, the third parties still carry the responsibility for their deployed services, and are committed to the security of their products. Finally, despite their limited reach, contributing an additional security control to ensure defence in depth may already be sufficient to largely mitigate attacks such as CT-GAN. Figure 3.1 illustrates the defence in depth paradigm and exemplifies how an attack may be prevented if multiple parties contribute to the overall security within an organization.

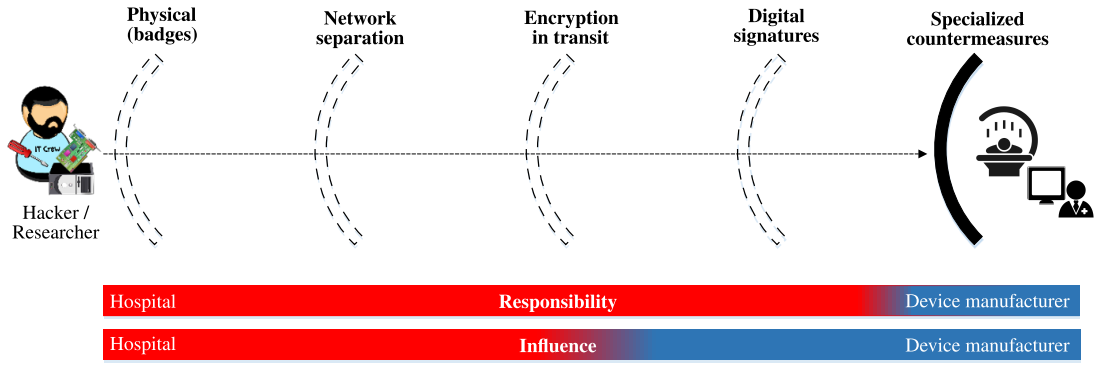


Figure 3.1: Defence in depth provided by multiple parties.

3.4.2 Deployment in Practice

The previous section elaborated on a few key factors which also shape the potential deployment of the proposed classifier. Specifically, the solution must not be disruptive and should only contain a realistic scope. As such, the classifier may be deployed in close association with the devices that are provided by their respective manufacturers. This ensures a large degree of independence from the hospital infrastructure, while also providing third parties with governance over their own security control. Furthermore, the classifier would operate as an additional layer in the defence in depth paradigm, specialized to identify tampered images. The classifier should therefore be unobtrusive and never obstruct any of the existing business processes of a hospital.

In practice, the classifier may be deployed within a workstation and activate whenever a medical professional chooses to analyze a medical image for potential diseases. The classifier would raise a warning whenever a CT scan contains evidence of tampering, encouraging the medical professional to assess the image more carefully. This procedure is illustrated in Figure 1.2. A high or consecutive amount of warnings by the classifier could indicate a persistent attack, which is likely to prompt

further investigation and lead to an early detection of the underlying security breach.

3.4.3 Attacker Model

In order to evaluate the proposed solution as a whole, it is important to define an attacker model. Since the CT-GAN attack already showed its efficacy in practice, it is sensible to assume a similar attacker model. As such, it is assumed that the attacker has control over the hospital network and has furthermore infiltrated the picture archiving and communication system (PACS), which enables the attacker to intercept and modify CT scans either during transit or while stored in a database. The attacker furthermore has access to a GAN and is able to tamper with the image data (i.e. pixels) as well as the DICOM metadata attached to the image. However, the attacker will be unable to further train the GAN once it is deployed. Furthermore, the workstation of the medical professional that analyzes the images is not compromised. As such, during the assessment, the image as well as the workstation are out of reach for the attacker. Although the workstation is not accessible, the attacker is aware that the image will be scrutinized for any modifications. Finally, the attacker knows that the corresponding CT device is taken into account when the authenticity of the image is assessed.

Data Disparity

As the attacker exploits a deep learning network, i.e. a GAN, to tamper with the medical images, it is also important to consider the data driven aspect of this instrument. Namely, the amount of data available to both the defender and the adversary is an important factor in determining which party will be successful. When the attacker does not have sufficient training data available, the malicious GAN may produce fabricated samples of poor quality which consequently leads to an easy detection. Likewise, when the defender does not possess the required data to establish adequate decision boundaries to identify authentic images, the defender may raise a large amount of false alarms or inadvertently lets valid attacks slip by.

In practice, it is highly likely that an attacker has merely access to a limited amount of publicized datasets, while defenders such as device manufacturers have an abundance of additional data available. In addition, if an attacker would instead primarily attempt to trick the defending classifier and disregard the realism of the modifications, the attack may slip by the defending party, yet still raise the suspicion of the medical staff that assesses the images as the image may now depict anomalies that do not match the human anatomy. Moreover, the device fingerprint used in this thesis, which will be further discussed in section 5.4, is by no means optimized for CT scanner devices. As such, a more advanced and fragile fingerprinting technique would also add to the attacker's difficulty in executing a successful attack. Finally, since this thesis is primarily intended as exploratory work, the experiments in this thesis consider the scenario in which the attacker and defender have access to the same dataset.

Chapter 4

Research Questions

The proposed solution introduced in section 3.3 aims to detect tampered CT images based on the device fingerprints that are left behind by CT scanner devices. The hypothesis behind the solution is that the tampering process distorts the underlying fingerprint and therefore inadvertently facilitates the detection of the attack. In this chapter of the thesis, a set of research questions is formulated which aim to validate the hypothesis and advance the current state-of-the art. These research questions will be adequately answered by evaluating multiple experiments which have also been conducted as part of this thesis. The detailed experiments, evaluation and answer to each research question are described in their respective chapters (chapters 6 and 7).

4.1 Research Questions

In order to gain understanding and sensibly evaluate the possible mitigations against tampered medical images, this explorative study sets out to answer the following research questions.

R1: Can a deep learning approach distinguish between various CT scanner devices based on CT images? Previous works which aim to determine the device origin of CT images rely on either traditional metrics, such as correlation [45, 46, 47], or apply various preprocessing and feature engineering steps before letting a SVM [9] make the final classification [22, 52]. Deep learning approaches, such as convolutional neural networks (CNN) [51], have not yet been applied for this task. However, these methods have shown exceptional performance on other image classification tasks. As such, it is valuable to investigate how a neural network fares when classifying unprocessed CT images. This question therefore aims to establish a baseline for future classification tasks on CT images. This question will be further motivated and answered in chapter 6.

R2: How do extracted device fingerprints influence CT scanner classification performed by deep learning? The device fingerprinting concept introduced in [54] has led to improved results for device identification, and a variation of this technique has been used as a preprocessing step for an SVM in the medical domain [22]. However, deep learning approaches have the added benefit of shaping their own feature representation, often eliminating the necessity of preprocessing the input data. As such, this question seeks to investigate the performance of a neural network when classifying the extracted device fingerprint of an image, compared to the classification on the original, unprocessed

images itself. As the device fingerprint will ideally play a key role in detecting generated samples, answering this research question will indicate any potential conflict between the preprocessing step and the own learnability of the neural network. This question will also be further motivated and answered in chapter 6.

R3: Can device fingerprints be leveraged to support the detection of artificially generated CT imagery? Previous work has shown that artificially generated images leave behind distinct traces which enable fine-grained attribution to identify the source generator of the generated samples [55, 100]. However, the goal in this thesis is to determine whether medical images, inpainted by a GAN, leave sufficiently large traces in the tampered images such that they cause distinguishable distortions. This research question aims to determine whether, and to what extent, the distortion created by a GAN affects the device fingerprint. Will the fingerprint further highlight the distortion, or counterproductively conceal the distortion? This question will be answered in chapter 7.

R4: Is the detection affected when a wider range of generative adversarial networks are evaluated? The work from [55] has shown that GANs have distinct fingerprints; even a divergence in the architecture affects the fingerprint [100]. This research question therefore aims to investigate how the performance of the solution is affected when multiple GANs are evaluated. Will the classifier detect all fingerprints equally well, or are some harder to detect than others? This question will be answered in chapter 7.

Chapter 5

Experimental Setup

To adequately answer the research questions, multiple experiments were conducted. This chapter details the considerations that persist throughout all of the experiments. These considerations are related to the processing of the dataset as well as assumptions on the data itself. Section 5.1 first describes the content, format and size of the dataset used in the experiments. Section 5.2 subsequently details the necessary steps to incorporate the data into the experimental environment. Section 5.3 then highlights the potential relationships that exist between the data samples. Section 5.4 details the specific algorithm that is used to extract the noise patterns which make up the device fingerprints introduced in section 3.2.2 and 3.3. Finally, section 5.5 describes the experimental environment as well as the metrics used to evaluate the experiments.

5.1 Dataset

In order to effectively conduct the experiments in this study, a collection of existing CT scans is required. CT scans are generally stored in the 'Digital Imaging and Communications in Medicine' (DICOM) file format[1] which, besides the image data, contains useful metadata related to the image scanning process itself. The DICOM metadata also provides information about the scanner. Specifically, the manufacturer, the specific scanner model and even the software version are given, although the latter is only supplied sparingly. Within a single CT session (or study) of a patient, a large amount (often over a hundred) of 2D images is taken. Each individual 2D image is called a *slice*, and the accumulation of these slices from a single session form a *volume*.

An anonymized and publicly accessible collection of CT imagery is provided by the 'Lung Image Database Consortium image collection' (LIDC-IDRI) [4] which contains a total of 1018 CT image series, amounting to a total of 244527 2D image slices. Each individual slice has a size of 512×512 pixels. As the experiments within this thesis are largely based on the presence of device fingerprints of CT scanners, it is valuable to also explore the diversity and distribution of the device types present within this dataset. Table A.1 highlights the 4 CT scanner manufacturers which together account for a total of 17 scanner models represented in the LIDC-IDRI dataset. Although the dataset principally contains a sufficient amount of samples, the table also illustrates the high imbalance of the classes. Especially devices from General Electric are overrepresented. However, due to the sensitive (medical)

[1] `https://www.dicomlibrary.com/dicom/`

content of the data, not many other datasets are publicly available. The LIDC-IDRI is, to our knowledge, the largest publicly available dataset. Furthermore, this is the same dataset which was used for training by the (existing) GANs described in chapter 7. As such, the LIDC-IDRI is still considered a viable dataset, despite its shortcomings related to the class imbalance.

5.2 Data Processing

The original LIDC-IDRI dataset is structured in a hierarchy of directories and is supplemented with an additional *.csv* file containing metainfo related to each of the studies. This metainfo file is a digest of the full DICOM metadata that is embedded in each file.

In order to structure the data in a manner that is more suitable for the experiments at hand, i.e. data organized by CT device manufacturer and model, all images were processed and collectively stored into a .hdf5[2] file. This particular format provides the benefit of solely storing the original image data along with a small subset of the largely redundant DICOM metadata. In addition to the relatively lightweight storage of this format, consulting metadata ('attributes' in hdf5 terms) does not load the largely-sized images into memory. Instead, the hdf5 format provides the ability to further filter and organize the data up until the point where the actual image is needed, which makes additional filtering substantially faster.

5.3 Assumptions on the Data

The DICOM format, in which each CT slice is originally stored, contains an abundance of metadata related to the scan. In addition to the device information briefly discussed in the previous sections, the metadata furthermore contains data such as the beam strength during the scan, and additional patient information (anonymized in LIDC-IDRI). The metadata also provides coordinates (within a XYZ-plane) that indicate the relative position of a slice within the complete volume. However, these properties may also introduce unforeseen relationships among different CT slices.

As the majority of the experiments involve splitting the data into separate and independent training and test sets, it is crucial to consider potential relationships among individual slices to ensure a fair and unbiased split in the data. These considerations lead to the following assumptions.

1. **Slices are affected by CT Scanner model and manufacturer**. This assumption may even be considered the basis of the experiments. It is assumed that scanner devices leave behind consistent fingerprint in their images, and by extracting a distinct noise component from a specific slice it is possible to attribute each slice to a specific device.

2. **Slices carry relationships with other slices from the same volume**. Besides originating from the same scanner device model, all slices stemming from the same volume are bound to have additional properties in common. Namely, every slice is likely influenced by the specific conditions and settings of the scanning device during a particular session. Most importantly, all slices within the same volume depict an image from the same patient at a particular point in time, potentially connecting slices by unique features present in the image.

3. **Slices which have x, y, z coordinates in common may have greater similarity compared with those who do not**. It is assumed that certain slices taken at a particular angle, share attributes

[2] `https://www.hdfgroup.org/`

with slices taken from the same angle. This effect may be amplified when the type of image content is the same. For example, a slice which depicts the top of a skull may show greater similarity with other slices showing the top than with slices depicting the side of the skull.

4. **Slices are affected by the image content of the scan**. Finally, the type of content shown in the image also affects the relationship between two slices. A volume presenting a chest scan is likely more related to another volume of a chest scan than to a volume which presents a CT scan of a skull.

Due to the limited amount of data available, it is infeasible to account for all of these assumptions and split the data in such a way that training and test data are completely independent and unrelated. However, some of these relations may be diminished by the noise extraction (section 5.4) performed throughout the experiments. As the experiments are primarily conducted on the extracted noise of a slice, it is assumed that little information from the original image leaks into the noise image. As a result, the similarities mentioned in point *3)* in the above list may, at least, no longer rely on the depicted image (e.g "side of a skull"). However, the specific angle may still be observable in the final noise image. The last point (*4)*), regarding the *type* of image content, may only minimally affect the experiments as the dataset only consists of chest CT scans. As such, all images depict similar content.

As the experiments actively aim to study the presence and capability of device fingerprints in CT scans as noted in point *1)*, the only relationship likely to be still in tact even after the noise extraction will be point *2)*. In an effort to eliminate this factor as well, the data split for training and test data is predominantly done on a volume-based level. Moreover, this ensures that a classifier exclusively faces unseen volumes during its test phase, which also mimics the setting for a real-world deployment of the classifier. Figures 5.1 and 5.2 illustrate the manner in which the data may be split. As presented in Figure 5.1, the data spit based on individual slices cause a large number of neighbouring slices to get separated among the training and test sets.

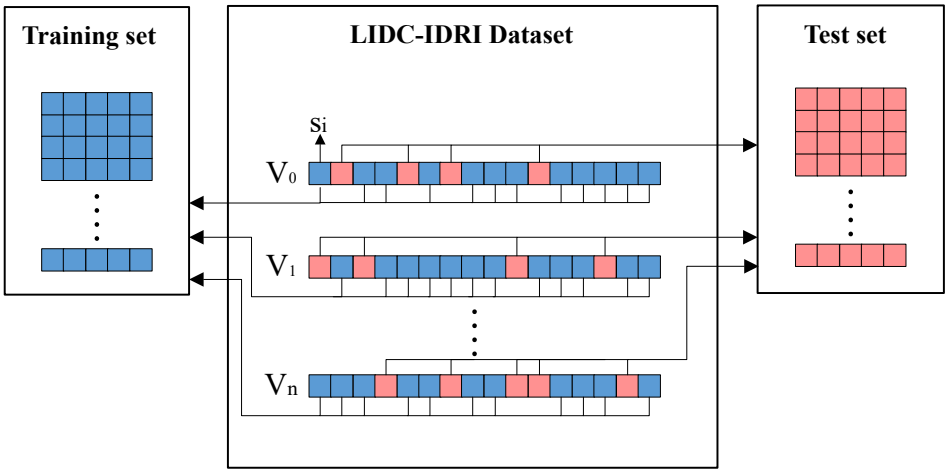


Figure 5.1: A split based on individual CT slices. Neighbouring slices may end in either split.

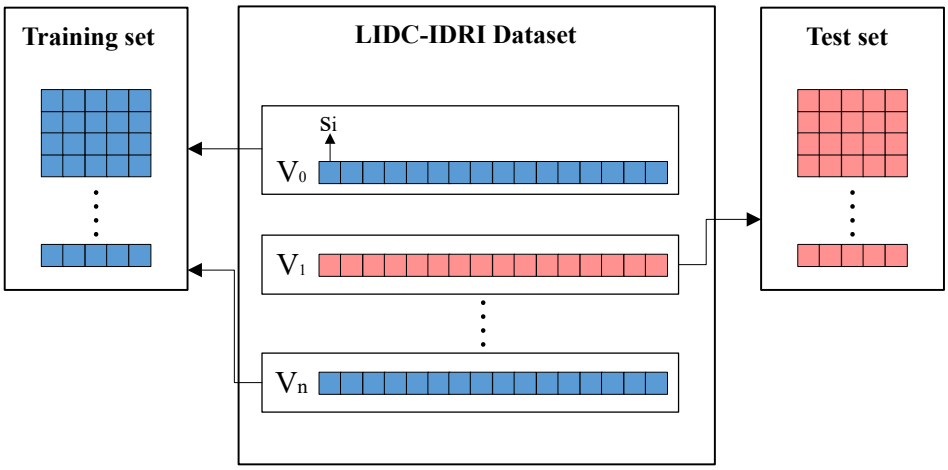


Figure 5.2: A split based on CT Volumes. This type of split will keep several potential relationships among slices within a single side of the split.

5.4 Acquiring Fingerprints of CT Scanners

Sections 3.2.2 and 3.3 introduced the related work which detailed the extraction of specific noise patterns, and their potential to capture distinct device fingerprints. The key motivation behind a "CT scanner fingerprint" is to extract a fragile, yet static set of features from a CT image. These extracted features should be static and nearly identical for all images taken from a distinctive type of CT scanner model, yet should be sufficiently fragile such that the same set of features are distinguishable from other CT scanner models. This approach has first been applied with great success for regular camera models [54], but has also found application in the medical domain [46, 22].

As this set of features should be nearly identical for each device of a specific CT scanner model, even a small modification would be noticeable. This assumed property is the key motivation behind the detection of generated samples by a GAN. It brings forth the hypothesis that a GAN, even on an inpainting task, will produce samples that noticeably divert from the expected fingerprint, which in turn leads to accurate detection of the generated samples. In addition, the type of GAN which produces the samples should not have any influence on the detection as the detector will merely pay attention to a broken fingerprint, not the realism of an image. Finally, as GANs remain an open challenge to train, adding the complexity of the desired fingerprint to one of the training goals will surely affect the image quality. The degraded quality would consequently deteriorate the primary objective of the malicious GAN: producing realistic images capable of deceiving medical staff.

In the experiments of this thesis, the fingerprints of the CT scanner models are acquired by extracting a specific noise component from each slice within the LIDC-IDRI dataset. The noise extraction algorithm used throughout the experiments is based on the approach applied by Lukáš et al. [54] which uses the filtering technique proposed in [59].

To denoise a slice s from a volume V, the denoising filter F is applied. The denoised image is subtracted from the original slice to obtain the desired noise component w:

$$w = s - F(s), \quad s \in V$$

The denoising filter [59, 54] is constructed in the wavelet domain in two stages. First, the local image

variance is estimated, then, a Wiener filter is applied to obtain the denoised image. These stages are performed using the steps described below:

1. The fourth-level wavelet decomposition is calculated for the slice.
2. For each level, the three high frequency-bands are used for further processing. These are the vertical, horizontal and diagonal sub-bands.
3. For each wavelet coefficient in the sub-bands, the local variance $\hat{\sigma}^2(i,j)$ is estimated within a $W \times W$-square neighborhood, for $W \in \{3,5,7,9\}$. The minimum of these 4 variances is chosen as the final estimate for the image variance:

$$\hat{\sigma}_W^2(i,j) = \max\left(0, \frac{1}{W^2}\sum_{(i,j)\in N} h^2(i,j) - \hat{\sigma}_0^2\right), (i,j) \in J$$

$$\hat{\sigma}^2(i,j) = \min(\hat{\sigma}_3^2(i,j), \hat{\sigma}_5^2(i,j), \hat{\sigma}_7^2(i,j), \hat{\sigma}_9^2(i,j)), (i,j) \in J$$

   J denotes the decomposition level. $\hat{\sigma}_0$ is a manually set parameter, and has been set to $\hat{\sigma}_0 = 5$ to follow the original work [54].

4. After the variance estimation, the Wiener filter is applied to obtain the denoised wavelet coefficients:

$$X_{den}(i,j) = X(i,j)\frac{\hat{\sigma}^2(i,j)}{\hat{\sigma}^2(i,j) + \hat{\sigma}_0^2}$$

5. The above steps $2-4$ are repeated for each wavelet sub-band, on each level of the decomposition.
6. The finally denoised image $F(s)$ is then obtained by applying the inverse wavelet transform.

An exact implementation of this algorithm, written in Matlab, has been kindly published by the original authors[3]. To minimize potential mistakes and, facilitate reproducibility, this is also the implementation used throughout the experiments conducted in this thesis.

5.5 (Model) Training and Classification

5.5.1 Environment

With the exception of the noise extraction algorithm, all experiments and data processing steps have been implemented in Python 3.7. For the noise extraction, the Matlab implementation by [15] was used. Due to the discrepancy in environments and to accelerate the preprocessing, the noise extraction was done on the entire dataset before any experiment and stored as a separate dataset.

The noise extraction was exclusively conducted on a HP Zbook studio G3 with an Intel Core i7-6700HQ, 8GB of RAM, and a Nvidia Quadro M1000M. Deep learning models were primarily trained and evaluated on a separate cluster kindly provided by the Eindhoven university data mining group. The machine used throughout most experiments had an Intel Xeon Broadwell-EP 2683v4, 1024Gb of RAM, and a Geforce GTX 1080 Ti. Training the GAN has been accomplished on a machine with an Intel Xeon E5-2698v4, 256GB of RAM, and a Nvidia Tesla V100.

[3] `http://dde.binghamton.edu/download/camera_fingerprint/`

5.5.2 Metrics

All experiments conducted in this thesis involved classification tasks; given a data sample X, having a single label y_{true}, a model seeks to predict the correct label y_{pred}. The classification is a success if $y_{pred} = y_{true}$. The performance, or success rate of a model during the experiments is measured using accuracy and precision. Accuracy measures the overall performance of the model by calculating:

$$\text{Accuracy} = \frac{\text{true positives + true negatives}}{\text{total number of predictions}}$$

In addition, particularly for the detection of the forged samples, it is important to inspect the number of times that the model (mis)classified authentic samples as being fake. Considering a real world example where the deployed model will give a warning whenever it detects a tampered sample, a large number of false positives may cause the warnings to be consistently ignored by the responsible staff, even in cases where the warning may be justified. The rate of justified warnings is indicated by precision:

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives + false positives}}$$

GAN Evaluation

The quality of samples produced by generative adversarial networks are often assessed using the Inception Score [83]. This score uses the popular inception model, such as [89], to indicate whether the generated samples are classified correctly and furthermore measures the diversity of the samples.

The original CT-GAN model [63] has not been tested in such a way, instead, it was deployed in practice and evaluated by the rate in which it managed to mislead professional medical staff. As the experiments in this thesis predominantly focus on the underlying noise pattern of images, the achieved realism of the generated samples of the GAN is merely a secondary concern, and as such has not been rigorously evaluated.

Chapter 6

CT Scanner Classification Based on CT Images

This chapter sets out to answer research questions R1 and R2 introduced in section 4.1. Specifically, the experiments conducted in this chapter aim to determine to what extent a deep learning approach is able to classify CT scanner devices, based on the images which they produce. In addition, a pre-processing step is evaluated which aims to extract persistent device fingerprints to further improve the classification performance. The primary purpose of these experiments is to investigate the effect of the fingerprint extraction technique on the classification performance. Although the fingerprints are likely to extract helpful features, it may conflict with the ability of a neural network to select its own range of features from the unprocessed image.

The chapter is structured as follows. Section 6.1 introduces and discusses the related work. Section 6.2 reproduces the experiments from the related work and is meant to verify the experimental setup. As the experiments from the related work still have a fairly limited scope, section 6.3 extends the objective, and introduces a new set of extended experiments which feature the approach of the related work, as well as the use of a neural network to perform the experiments. Section 6.4 subsequently presents the results of the experiments. Finally, section 6.5 and 6.6 conclude the chapter with a discussion and conclusion.

6.1 Related Work

Classifying the device origin of CT images is a topic which is only scarcely explored [52, 22, 46, 45, 47]. Previous works which aim to identify the devices rely on image correlation, sometimes supplemented with an SVM classifier to make the final classification [22]. Furthermore, related works often leverage a variation of the sensor pattern noise (SPN) extraction, popularized by Lukáš et al. [54]. This process, illustrated in Figure 6.1, extracts the distinct noise from a chosen number of images and subsequently averages the resulting noise patterns to create a reference pattern, which will then serve as the common device fingerprint. The commonly used $80 - 256$ number of images to create the reference pattern are all previously known to stem from a certain device.

The subsequent classification process itself is illustrated by Figure 6.2; an unknown image is compared to all of the previously computed reference patterns (Figure 6.1). The resulting correlation with a

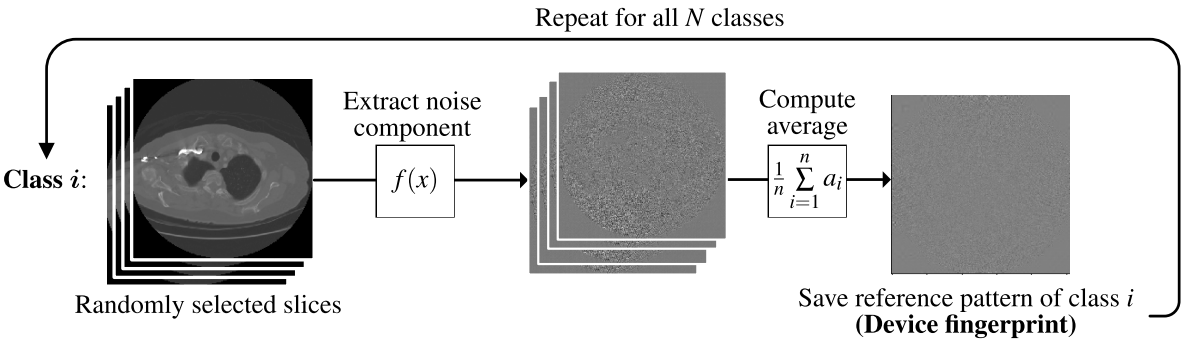


Figure 6.1: Creation of reference patterns for a certain amount of devices, given a set of slices from known devices.

reference pattern indicates the likelihood that the image originates from the same device. This may be accomplished by either establishing a certain threshold for the correlation value, or let the decision be based on the pattern that exhibits the highest correlation.

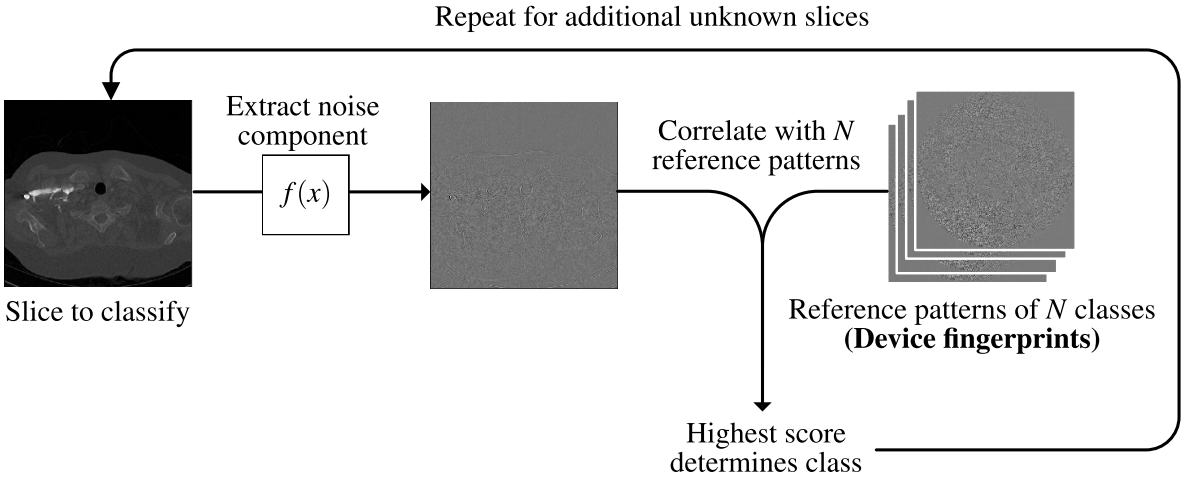


Figure 6.2: The process to determine the class of unknown slices, given a precomputed set of reference pattern.

Kharboutly et al. [46] applied this approach on CT images to distinguish three CT scanner models and achieved a classification accuracy of 97%. As CT scanners capture images differently compared to ordinary cameras, Duan et al. [22] did not directly apply the denoising technique on the CT images, but instead applied additional preprocessing on the images to extract the Original Sensor Pattern Noise (OSPN) of the scanners as well as the image reconstruction algorithm. Their best approach achieved an accuracy of 96.65% when classifying 15 different CT scanner models. Differently, Lee at al. [52] did not apply any preprocessing, and instead perform a quantitative analysis based on the density distribution of Hounsfield Unit (HU) values within CT images. The authors fit a SVM on the density distributions from three CT scanner models, and obtain an accuracy of 91.1% when classifying the three models.

6.1.1 Potential Shortcomings for Generalizability

Although the related work demonstrates promising results, it is important to address several choices in the experiments and chosen datasets which may limit the applicability of the related work in practice. The experiments by Kharboutly et al. [46] classify a total of three specific CT scanners of undisclosed models. In addition, in the original work, the slices which are used to compute the reference patterns make up for 45% of the entire dataset, and no volume is excluded during the computation of the reference patterns. Although subsequent work by the authors has used a larger set of images [45, 47], none of the volumes are expressly excluded from the computation of the reference patterns. This may lead to favourable results as the test is likely affected by the undesired relationships between slices previously discussed in section 5.3. Especially considerations 2 and 4 are not taken into account, which detail the potential relationship of neighbouring slices within a volume, and the content depicted by the image volume itself.

Similarly, the experiments by Duan et al. seem to consider only a single volume per device, with varying anatomical objects for different volumes. Although their experiments consider 15 devices as opposed to three, the use of a single volume per device is likely to keep several relationships in tact (again, point 2 and 4 of section 5.3).

6.2 Reproduction of Related Work

Although the denoising approach (section 5.4) used in this thesis is similar to the related work, the dataset as well as the exact implementation of the algorithms differ. As such, it is important to first reproduce several experiments from the related work and compare the results. Major discrepancy in the results may indicate misconceptions which may be detrimental in subsequent experiments.

To perform this first set of experiments, the setup detailed in sections 5.1, 5.2, 5.4 and 5.5.1 of the previous chapter has been applied. Specifically, the procedures illustrated in Figures 6.1 and 6.2 have been implemented using Python 3.7. For the extraction of the noise patterns, an existing Matlab implementation has been used (section 5.4). All data samples come from the publicly available LIDC-IDRI dataset [4] which is further detailed in section 5.1.

6.2.1 Simple Scanner Device Classification

The experiments by Kharboutly et al. [46] consider a total of three devices, manufactured by either Siemens or General Electric. Their original dataset contained a total of 8 volumes with 100 slices contained in each volume. The Siemens devices were each represented by 3 volumes, while the remaining 2 volumes contained images from the GE device. For each device, 120 slices were used to compute the reference pattern as demonstrated by Figure 6.1, while the remaining $80 - 180$ slices from each device served as the test set.

Since the LIDC-IDRI dataset used in our experiments did not have volumes with exactly 100 slices, volumes of similar size were prioritized instead. As the exact device models used in the original experiment are unknown, we opted to select the devices at random for the reproduction. The result is presented in Figure 6.3. As can be seen, the reproduction led to an overall accuracy of 99.7%, while the original work reported an accuracy of 95.8%. This small discrepancy may be due to the random selection of devices as well as the potentially modified denoising technique (section 5.4). Nevertheless, the great similarity in the results affirm that the implementation of the experiments

match closely.

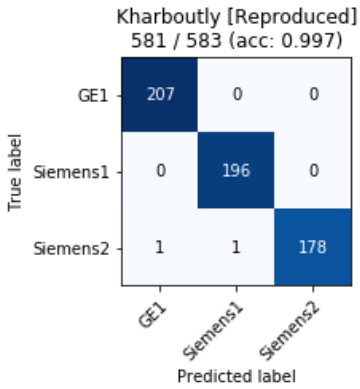


Figure 6.3: Result from reproducing Kharboutly et al. [46] with 3.9% improved accuracy.

6.2.2 CT Scanner Model Classification

The second related work which applied the extraction of noise patterns, conducted by Duan et al. [22], applied a more advanced approach in order to classify a total of 15 CT scanner models produced by 4 different manufacturers. Their denoising approach, based on OSPN (Original Sensor Pattern Noise), is considerably more complex and requires additional domain knowledge. However, since this thesis serves an exploratory purpose, their approach has not been replicated for this thesis. As such, the reproduction of their experiments is not entirely accurate. In addition, their experiment features a slightly different set of models, and furthermore combines CT volumes from a variety of unknown sources which further hampers an exact replication of the experimental setup. Nevertheless, the researchers present an interesting task and it is valuable to investigate the relationships among specific scanner models as well as determine how fine-grained the device fingerprint can distinguish a wider range of classes.

To simulate the original experiment as closely as possible, a single volume per scanner model was selected at random. Then, for each volume, 30 slices are picked at random to create the reference patterns, while 100 slices from the same volume are selected for the test phase.

The confusion matrix in figure B.1 within appendix B shows that, with the exception of General Electric devices, scanner models are classified perfectly. Furthermore, the misclassified General Electric devices get predominantly classified as the Lightspeed Ultra device. A potential cause for this phenomenon may be that this device lends exclusive components to the other devices within the General Electric family. However, one could expect that the component sharing would lead to a wider misclassification distribution within General Electric, which is clearly not the case. As this isolated incident is not alarming for the experiment, it is not further investigated. Finally, as was in line with expectations, the overall classification accuracy is lower than in the original experiment, most likely due to the discrepancy in the preprocessing step (OSPN vs SPN).

6.3 Expanding the Experiments for Greater Applicability

The original experiments from the related work either consider a limited set of devices, or feature only as a small amount of CT volumes per device. In addition, neither of the existing works based on noise patterns consider volumes that are intentionally left out of the reference pattern. As a result, the reference pattern is never tested against slices from completely new volumes, which would be the prevalent case in practice. The current setup of the experiments may therefore considerably favour the classification performance, and provides only limited insight into the feasibility of device identification in practice. This section therefore introduces a new set of experiments which consider stricter, yet more pragmatic conditions.

6.3.1 Objectives

Although the current pieces of related work all aim to classify or identify CT scanner devices, the experimental setups are not directly comparable. Kharboutly et al. [46] consider 3 specific devices from 2 manufacturers, Duan et al. [22] classify 15 scanner models from 4 manufacturers, and Lee et al. [52] classify a total of 3 scanner models from 3 manufacturers. The expanded set of experiments in this chapter define two clear objectives. The first objective is the classification of manufacturers of CT scanner devices, while the second objective is to classify specific scanner models. Each objective considers classification based on the respective image data of CT scans. The totality of the classes is presented by Table 6.1 which features the 4 manufacturers and 15 scanner models used throughout the expanded set of experiments.

1. **Objective 1**: Classification of 4 CT scanner manufacturers.
2. **Objective 2**: Classification of 15 CT scanner models.

6.3.2 Approach and Experimental Setup

In addition to the definition of two clear objectives, the experiments in this thesis uses a reproducible experimental setup which uses the publicly available LIDC-IDRI dataset. This section further details the experimental setup, and describes the approaches which have been applied to perform the new set of experiments.

Baselines for CT Scanner Classification.

The related works which consider additional preprocessing steps do not establish a baseline for their results [22, 45, 46, 47]. Specifically, these works rely on noise patterns for the classification, but exclusively perform the experiments on the processed images, without considering the performance on original CT images. As such, it is not possible to validate whether the extraction of noise patterns, i.e. device fingerprints, is a beneficial preprocessing step. To remedy this situation, the approaches mentioned in this section also conduct each experiment on the original, unprocessed CT slices.

Reference Pattern and Correlation

The first approach is similar to the related work of Kharboutly et al. [46] which computes a reference pattern per device and consequently applies correlation to obtain the classification. To evaluate this approach on the new objectives (section 6.3.1) and the larger LIDC-IDRI dataset, the approach has been left unchanged. Aside from a potentially different implementation for the denoising technique

Manufacturer	Model
Philips	Brilliance 16 Brilliance 16P Brilliance 40 Brilliance 64
GE Medical Systems	Lightspeed plus Lightspeed power Lightspeed Pro 16 Lightspeed QX/i Lightspeed Ultra Lightspeed VCT Lightspeed 16
Siemens	Emotion 6 Sensation 16 Sensation 64
Toshiba	Aquilon

Table 6.1: The classes for the CT scanner classification experiments.

(section 5.4), the approach as illustrated in Figures 6.1 and 6.2 remains the same. Specifically, in the first phase, a reference pattern for each class is established by selecting slices from each class at random. Then, the noise from each of the selected slices is extracted, and subsequently averaged. This averaged noise component then serves as the device fingerprint for a particular class. The experiments which establish the baseline described in the previous section 6.3.2, skip the noise extraction step and instead will average the unprocessed images.

During classification, each slice that is to be classified will be compared to each of the reference patterns by correlation. The highest correlation value determines the class, which slightly differs from the related work of Kharboutly et al. [46] which applied a threshold instead. Nevertheless, similar to the work of [46], the computed reference patterns for both the manufacturer classification and model classification objective are computed using the average of 120 slices per class. However, the slices from volumes embedded in the reference pattern are entirely separated from the volumes that are classified during the test phase. This simulates the use case of classifying new, unseen volumes.

Classification with Deep Learning

The related works which did not apply direct correlation, have only applied traditional machine learning, specifically SVMs, to aid in the CT scanner classification tasks [22, 52]. However, deep convolutional neural networks may also be suitable classifiers for this task as they have consistently demonstrated excellent results on image classification. As such, this section details an approach which uses a CNN to perform the classification tasks.

[1]Note that the features within the patterns are subtle and may become indistinguishable on print-out versions of this thesis.

Work	Input	Preprocessing	Classifier
Kharboutly et al. [46, 45, 47]	CT slices	Noise pattern [59]	Reference Pattern + Correlation SVM
Duan et al. [22]	CT slices	OSPN	Reference Pattern + Correlation SVM
Lee et al. [52]	Density function	—	SVM
This work	CT slices	Original image Noise Pattern[15]	Reference Pattern + Correlation CNN

Table 6.2: Classifier setup of experiments.

Work	Volumes	Classes	Data split
Kharboutly et al. [45, 46, 47]	$8-40$	3	Slice
Duan et al. [22]	15	15	Slice
Lee et al. [52]	326	3	**Volume**
This work	**1200**	$\mathbf{4,15}$	**Slice and volume**

Table 6.3: Comparison of objectives and datasets for CT scanner classification.

Approach and Setup. The neural networks are tasked with the same set of experiments as in the previous section. The first experiment aims to label CT slices by their respective manufacturer, while the second aims for a more fine-grained classification by distinguishing scanner device *models*. The LIDC-IDRI dataset is used for training and testing, and is split 80% and 20% for the training and test set, respectively. In addition, the data is split as illustrated by Figure 5.2. Specifically, the split is performed on a volume level, which prevents the separation of neighbouring slices into different sides of the train/test split. However, to evaluate the assumptions detailed in sections 6.1.1 and 5.3, the neural networks are also evaluated *once* on the alternative setup illustrated in Figure 5.1. For this splitting technique, each individual slice is put in either the training set or the test set at random, without any further consideration. Although this setup may trigger favourable classification results, it is important to verify and highlight the importance of a sensible data split.

Finally, to also establish a baseline for the classification by neural networks, the performance between the noise patterns and unprocessed CT slices as input samples is compared (section 6.3.2).

Neural Network Architecture. As the purpose of the experiments is to first and foremost explore the potential of device fingerprints, an existing architecture is chosen over the alternative of designing and building a new architecture for this specific task. A common approach in this regard is to employ transfer learning, in which a successful model from one domain is fine-tuned to perform a similar task in a different domain [70]. This approach has also been applied successfully in the medical domain [56].

Initially, transfer learning had also been attempted for the experiments in this thesis. For the base architectures, the pre-trained networks from Inception V3 [89] and ResNet50 [33] models were selected. The networks were then fine-tuned for the CT device experiments using the LIDC-IDRI dataset. However, both of these network quickly overfitted on the training data and performed unsatisfactory on the validation set. One potential cause may have been the relatively small number of samples within the LIDC-IDRI dataset. Another contributing factor to the underperformance may have been the content

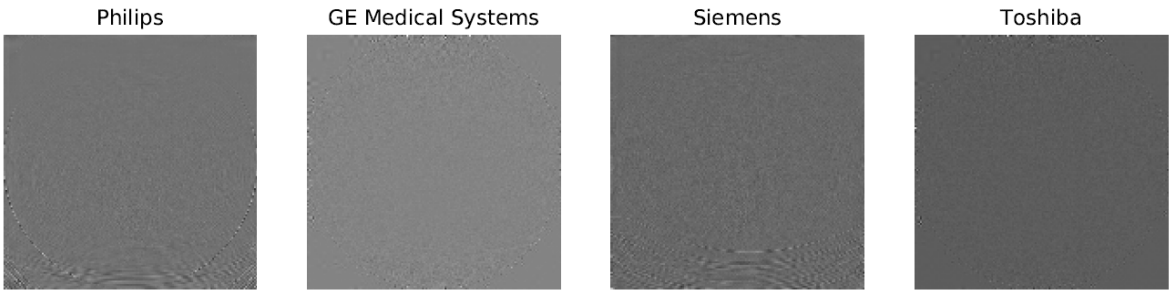


Figure 6.4: Examples of computed reference patterns for the different CT scanner manufacturers.[1]

of the data itself. Specifically, the original networks are trained on the ImageNet [21] dataset which contains multi-channel (RGB) images of ordinary content, while the LIDC-IDRI dataset contains single-channel CT images.

Consequently, a new architecture was chosen which only contained the first few layers of the Inception V3 model. This architecture contained the first three convolutional layers, but with a reduced number of filters and an additional dropout layer to avoid overfitting. Furthermore, the new network was not pre-trained, and instead directly trained on the LIDC-IDRI dataset. Finally, to reduce the training speed, complexity and to stand by the original IV3 architecture, the image dimensions have been reduced to 299×299. Although in earlier iterations of the experiments, the original dimensions (512×512) have also been applied, no significant impact on the performance was observed.

An overview of the architecture is presented in Figure 6.5. This new architecture performed well on all tasks, and has been applied for all deep learning experiments conducted in the remainder of this chapter. As such, it is important to reiterate the mainly exploratory purpose of this thesis. Although the chosen architecture performed well on all of the given tasks and provides valuable insights, an architecture optimized for each individual task would undoubtedly be able to outclass the current network, and is left as future work.

Although the general architecture has been constant throughout the experiments, the training parameters were adapted for each task. For the manufacturer classification task, the networks were trained for 10 epochs. For the larger scanner model classification task, the models were trained for 15 epochs. In addition, the top layer of the network has been adapted throughout the experiments to accommodate for the different number for classes (4 and 15).

Furthermore, all networks were trained on batches of size 128. Although different setups have been evaluated to accommodate for the different types of (un)processed input images, both types of input demonstrated the best performance with the architecture detailed in Figure 6.5. As such, the architecture and setup for classification on noise patterns and unprocessed slices is identical.

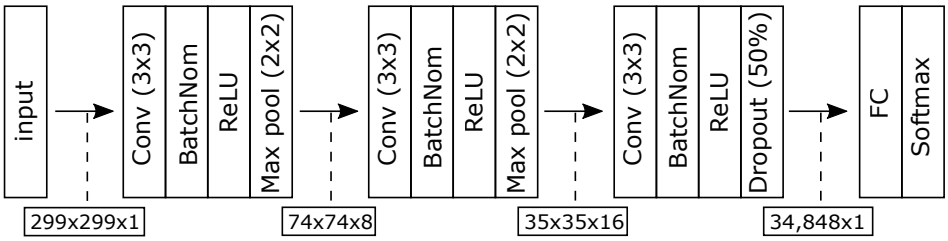


Figure 6.5: The CNN architecture used for the classification of CT scanners.

6.4 Results

This section presents the results of all experiments described in the previous sections. The experiments had a total of two objectives, with 4 and 15 classes respectively. Two approaches have been considered, based on correlation with a reference pattern, and the training of a neural network. In addition, each instance of the experiment considered two types of input images, being the unprocessed CT slices and the extracted noise patterns. Finally, the neural networks have also been evaluated on alternative data splits (Figure 5.2 and 5.1).

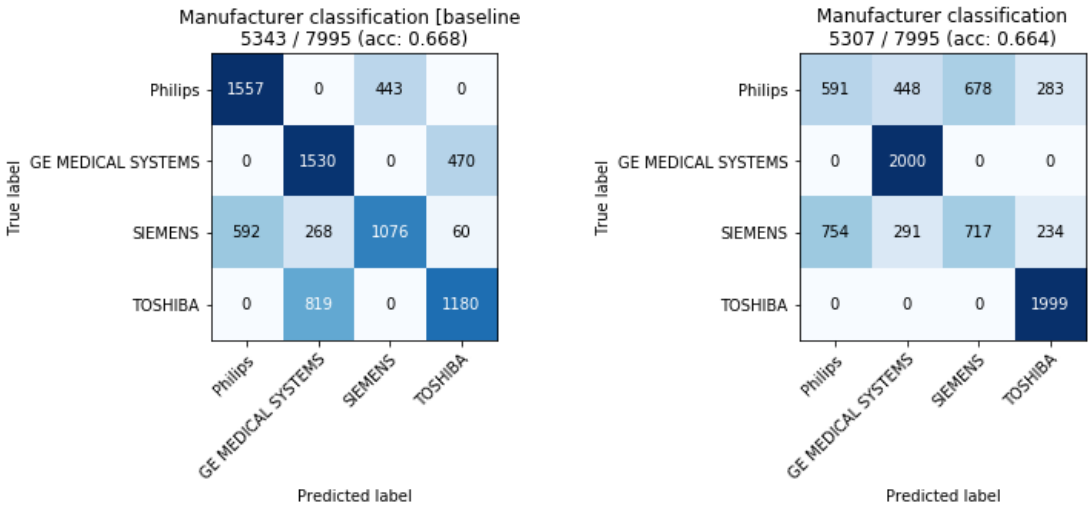
6.4.1 CT Scanner Device Manufacturer Classification

Reference Pattern and Correlation

As presented in Figure 6.6b, the extended objective and dataset have caused the accuracy to drop significantly compared to the initial reproduction related work illustrated in Figure 6.3. The new result, based on noise patterns, achieves an accuracy of 66.4%. The figure shows that slices from Toshiba and General Electric are classified perfectly, while CT slices from both Siemens and Philips are classified significantly worse. It is also peculiar that Philips and Siemens are most often misclassified amongst each other, which may indicate that these devices share certain software or hardware components which consequently affect the device fingerprint. In addition, similarities in the acquisition process may also cause a higher correlation between the images of these devices. However, we do not possess sufficient reference information or domain knowledge to further investigate this hypothesis.

For the classification on the original, unprocessed slices, the result presented in Figure 6.6a shows that, although the overall accuracy of 66.8% is similar to the previous result of 66.4% displayed in Figure 6.6b, predictions are significantly more distributed. Specifically, predictions for General Electric and Toshiba devices have become split amongst each other and are classified significantly worse. However, Philips and Siemens devices are predicted more accurately on the original images. Moreover, prediction mistakes for a certain class often fall into just one single, other class. This occurrence once more highlights the connection which seemingly exists between Philips and Siemens devices.

Nevertheless, the relatively low accuracy for either approach, being the classification on device fingerprints or original images, indicates that classification based on image correlation with a reference pattern is insufficient to make accurate predictions.



(a) Classification on unprocessed images.

(b) Classification on noise patterns.

Figure 6.6: Manufacturer classification by correlation.

Classification with Deep Learning

The reproduction of the related work in section 6.2 already alluded to the importance of a sensible train-test split of the data. Figure 6.7 shows that this hypothesis holds true for deep learning classification as well. When slices are naively divided into a training and test set (Figure 5.1), the classifier reaches an accuracy of 99.8% for the manufacturer classification task. This result even outperforms the much smaller setting presented in [46].

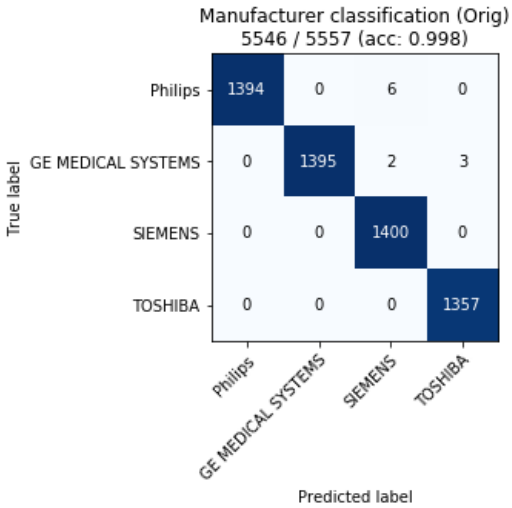
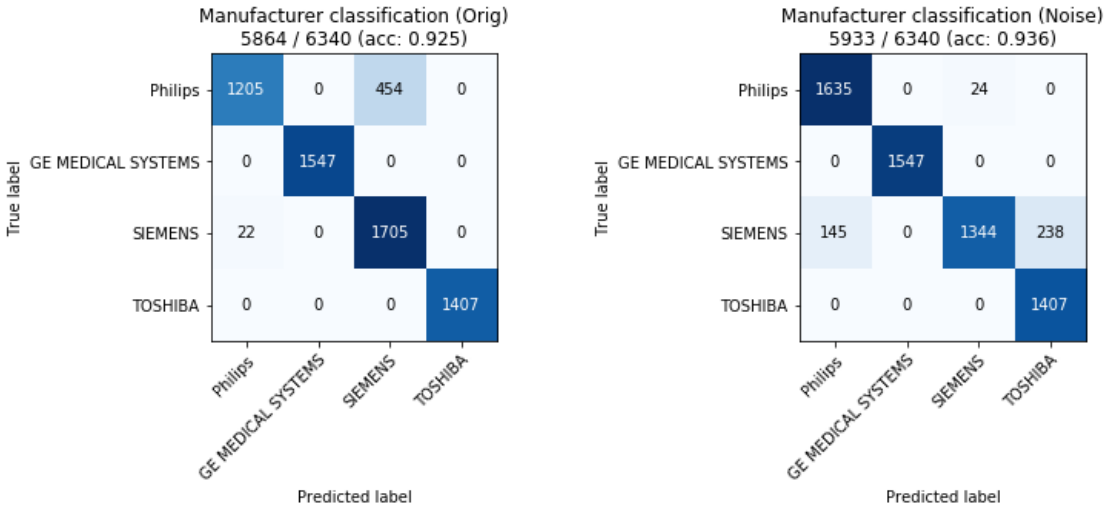


Figure 6.7: Manufacturer classification when volumes are disregarded during the data split.

Nevertheless, the neural network largely retains its performance even when the data is indeed split in a sensible manner. The confusion matrices presented in Figure 6.8 show that the classifier achieves up to 93.5% accuracy when all slices within a volume are exclusively selected for either the training or

test set. This is a significant improvement over the correlation-based experiments presented in Figure 6.6 which achieved a maximum accuracy of 66.8%. Furthermore, the deep learning classifier also performed slightly better when classifying noise patterns as opposed to the original CT images. This result suggests that the extraction of device fingerprints does not interfere with the capability of the neural network to learn its own features. Although an optimized neural network with suitable depth and layers should learn the device fingerprint using its own representations, the provided noise patterns nevertheless provide a positive contribution to the performance of the current model. Interestingly, the misclassifications shown in Figure 6.8a also continue to support the apparent similarity between Siemens and Philips devices already highlighted in section 6.2.



(a) Classification on original images.

(b) Classification on noise patterns.

Figure 6.8: Manufacturer classification using a neural network.

6.4.2 CT Scanner Model Classification

Reference Pattern and Correlation

Similarly as section 6.4.1, the result presented in Figure B.2 shows that accuracies drop to as low as 34% when volumes are strictly separated and the dataset greatly expanded. The unusual behaviour of the General Electric devices already seen in figure B.1 also seems to persist in the larger dataset. However, in this experiment the majority of General Electric devices get classified as the Lightspeed QX model, while in the previous experiment the Lightspeed Ultra model was predominantly predicted. Moreover, the result further highlights the peculiar relationship between Philips and Siemens devices which was already noticeable in the manufacturer classification experiment of section 6.4.1, visualized in Figure 6.6.

To conclude the experiments based on reference pattern and correlation, the extended setup from the previous sections is again used to classify CT slices based on the original imagery, i.e. without extracting the device fingerprint. In addition to the low accuracy of merely 27.8%, the result presented in Figure B.3 shows the greater spread of misclassification that also appeared in the comparable experiment for manufacturer classification presented in Figure 6.6a. Moreover, both results indicate similar relationships between the devices as Philips and Siemens devices remain most commonly misclassi-

Objective	Classes	Input	Classifier	Accuracy
Manufacturer	4	**Noise pattern**		
			CNN	**93.6%**
			Reference pattern + correlation	66.4%
		Original slice		
			CNN	92.5%
			Reference pattern + correlation	66.8%
Model	15	**Noise pattern**		
			CNN	**59.4%**
			Reference pattern + correlation	34%
		Original slice		
			CNN	47.3%
			Reference pattern + correlation	27.8%

Table 6.4: Complete list of results from experiment which consider a volume-based split.

fied among each other, while General Electric devices predominantly get predicted as devices from the same manufacturer.

Classification with Deep Learning

The results presented in Figure B.4 and B.5 in appendix B show that this task is significantly more challenging than mere manufacturer classification. Still, the neural network reaches substantially higher accuracy than the results obtained by usage of a reference pattern and correlation calculation presented in Figure B.2 and B.3. With a maximum accuracy of 59.4%, the neural network outperforms the usage of the reference pattern by 25.4%.

Furthermore, the classification on noise patterns outperforms the classification on original images by 12.1%, which further supports the hypothesis that CT scanner devices have certain device fingerprints. Interestingly, the type of misclassifications largely persist throughout the experiments, regardless of the underlying classifier. As can be seen by the coloured overlayed rectangles in Figures B.4 and B.5, misclassified samples predominantly get labelled as classes of the same manufacturer. In addition, certain Siemens device share similarities with Philips devices, a recurrent characteristic which is also notable in Figures B.2, B.3 and 6.8a.

6.5 Discussion

The results presented in the previous sections revealed some interesting points which warrant further discussion. Firstly, it is important to discuss the presence of device fingerprints for CT scanners. Moreover, the impact of a sensible data split, and the large gap in accuracy between device manufacturer and device model classification also warrants further discussion. Finally, it is valuable to briefly discuss the current limitations and further potential of the employed deep learning classifiers.

Potential of device fingerprints of CT scanners. The results obtained from the experiments indicate that the extraction of SPN-based device fingerprints has merit. Although the experiments which

use the reference pattern technique do not show a direct improvement in accuracy, predictions on fingerprints are less spread among devices, and maintain consistent prediction patterns. In contrast to the traditional classification technique, the deep learning classifiers yielded consistently higher performance when trained and tested on the noise patterns of devices. Nevertheless, the deep learning classifiers also seem capable of learning their own representations, as the accuracy between original images and noise patterns does not always yield significant improvements in performance (Figure 6.8).

Importance of a sensible data split. The experiments consistently show that the hypothesis introduced in point 2 of section 5.3 largely holds true: slices contained within the same volume hold persistent relationships with each other, resulting in optimistic classification when these related slices are unrestrictedly distributed among the training and test set. Although the computation of a reference pattern applied in the correlation experiments (section 6.2) is a relatively simple procedure to fit the data, it nevertheless influences the subsequent test phase. A clear separation in volumes selected to compute the reference pattern, and volumes used for later testing show significant effect on accuracies, as can be seen in the large gap of accuracy between e.g. Figure B.1 and B.2.

In the deep learning experiments of section 6.3.2, the separation of volumes becomes even more crucial. Given the common ratio of a train/test split, e.g 80/20, it is likely that neighbouring slices of a slice selected for testing, end up in the training set. This consequently causes two nearly identical slices to be used for both training and testing. Therefore, it is unsurprising that tolerating these separations, positively affects the accuracy. This is also highlighted by the near-perfect accuracy presented in Figure 6.7. As a consequence, the reported results from the related work which did not account for these separations, may be inaccurate and not directly apply in practice.

Model versus manufacturer classification. The third point of discussion is the substantially worse performance of device model classification in comparison to the classification of device manufacturers. There are several explanations which likely contribute to this discrepancy. As the difficulty of the task increases from 4 to 15 classes, so is the same amount of data distributed across more classes. As a result, there is less training data available per class which hinders the generalizability of the classifiers. Another factor which may contribute to the performance gap is the design of the devices itself paired with the corresponding device fingerprint. Devices are likely to share software as well as hardware components with models from the same manufacturer. As such, a different model may not necessarily have a unique device fingerprint, or the fingerprint may be too similar to be accurately captured by the current noise extraction technique. Although this hypothesis is supported by B.4 and B.5 which clearly shows that misclassified devices are most commonly labelled as a model from the same manufacturer, this does not fully explain the worse performing classification on the original, unprocessed images. As such, the general underperformance is likely brought about by the lack of data, and insufficient optimization of the neural network. As the LIDC-IDRI dataset is not entirely balanced for this type of task, certain classes only have three distinct volumes at their disposal during the training phase, which is likely insufficient to obtain accurate device fingerprints for specific device models.

Further potential of deep learning classifiers. As the final point of discussion, it is important to address the performance of the deep learning classifiers. Although the results show major progress compared to the correlation-based approach, further improvements are undoubtedly possible. The

current network architectures are based on existing networks for general-purpose image classification such as MNIST[2] and ImageNet [21]. Sensible optimization and fine-tuning of the networks is a promising step to achieve even higher performance. Likewise, augmenting the current, publicly available dataset with new data will likely lead to drastic improvement of the device model classifiers.

6.6 Conclusion

The previous sections discussed an array of experiments aimed to reproduce the related work, and expand the work under stricter, yet more pragmatic conditions. In addition, experiments were conducted on original, unprocessed CT slices in order to provide a direct comparison with the SPN-based device fingerprint. Table 6.3 shows a comparison between the related works and the extended experiments from this chapter. The table signifies the larger dataset used in this work, as well as the different manner in which the data may be split (section 5.3).

This chapter reproduced the experiments of the related work and highlighted the effect of a naive train/test split. Namely, CT slices that originate from the same volume share great similarities with each other, resulting in misguided accuracy when these related slices do not end up on the same side of the train/test split. As a consequence, the original experiments performed in the related work (section 6.1) indicated the potential of (O)SPN-based device fingerprints, yet do not form a sensible baseline for future results. To address this shortcoming, the experiments have been reproduced under stricter conditions and on a larger dataset. With maximum accuracies of up to 66.8% for device manufacturer classification and a mere 34% for device model classification, the new set of experiments showed that classification solely based on correlation is insufficient for any practical deployment.

Sections 6.4.1 and 6.4.2 also evaluated the performance of convolutional neural networks on the aforementioned experiments. As illustrated by Table 6.4, accuracies increased to 93.6% for device manufacturer classification and 59.4% for device model classification, with architectures that are primarily suited for general-purpose image recognition. The lack of a custom network further highlights the potential capability of a deep learning approach for this task. The still relatively low performance on device model classification has furthermore been discussed in previous section 6.5, and may be attributed to the design of the devices as well as the potentially limitation of the SPN-based fingerprint.

Nevertheless, throughout the experiments using the deep learning models, the preprocessing step of the SPN-based fingerprint consistently outperformed the classification on the original, unprocessed slices. This furthermore cements the assumption that CT scanner devices do indeed possess a consistent fingerprint even though the image acquisition process is entirely different from ordinary cameras. In addition, the generally improved performance highlights that the device fingerprints provide a positive contribution to the neural networks used in the current experiments.

6.6.1 Answer to Research Questions

Finally, This chapter set out to answer the first two research questions introduced in section 4.1. With the thorough experiments that have been conducted, these can now be confidently answered.

R1: Can a deep learning approach distinguish between various CT scanner devices based on CT images? The results presented in section 6.3.2 showed that a deep learning network can at least

[2] `http://yann.lecun.com/exdb/mnist/`

achieve 92.5% accuracy for device manufacturer classification and 47.3% when classifying specific models based on original, unprocessed CT slices. This is an improvement of $19.5\% - 25.7\%$ compared to the correlation-based approach initially investigated in the related work. Furthermore, these results should primarily be interpreted as a lower bound for the actual capability of a deep learning classifier as the employed network architectures have not been extensively optimized for the given tasks. Moreover, merely a single, publicly available dataset has been used during the experiments. As such, one could expect significantly higher accuracies for more optimized architectures, trained on larger datasets. In conclusion, the reported results support the hypothesis that convolutional neural network are well-suited to distinguish CT scanner devices based on their produced images, with high accuracy.

R2: How do extracted device fingerprints influence CT scanner classification performed by deep learning? Both experiments from section 6.3.2 have shown that device classification based on the SPN-based fingerprint consistently outperforms classification on original CT slices. During the manufacturer classification experiment, the network trained and tested on noise patterns slightly outperformed the original slices by 1.1%. This gap grows to 12.1% for the device model classification task. Nevertheless, a fully optimized network with sufficient data at its disposal should be capable to learn its own approximation of the device fingerprint, without the provision of the noise patterns as input. However, the results of the experiments show that the currently extracted fingerprint certainly captures useful features. Therefore, it can be concluded that the extraction of SPN-based device fingerprints provide a positive contribution to device classification by deep neural networks.

Chapter 7

Detection of Tampered CT Scans

The previous chapter highlighted the potential benefit of extracting specific noise patterns from CT images in order to determine the corresponding manufacturer and model of a CT scanner. The excellent results of the experiments furthermore confirmed the existence of implicit device fingerprints. This chapter will define the final set of experiments to discuss and answer the final set of research questions R3 and R4 from chapter 4. Specifically, this chapter will evaluate whether the device fingerprints are an effective instrument to detect image tampering performed by a GAN. Furthermore, the chapter will evaluate how the detection rate is affected when multiple types of GANs are evaluated.

The remainder of this chapter is structured as follows. Section 7.1 discusses the related work concerned with the detection of images tampered or produced by GANs. Section 7.2 will detail the experiment setup of the final set of experiments. This section will discuss the setup, as well as the generative networks and classifiers used for the experiments. Section 7.3 presents the results of the experiments. Finally, section 7.4 and 7.5 will discuss the results and conclude the chapter.

7.1 Related work

The detection of samples generated by GANs is an extensively explored topic. However, although GANs are applicable to a large amount of domains, the studies that aim to reliably detect fabricated content are primarily focused on a small subset of these domains. The domain of Deep fakes is perhaps the most well-studied domain related to the detection of AI-generated content. These fakes, most commonly featured in video format, encompass media that has been altered with high quality, and often depict celebrities in deceiving context [88]. To distinguish these videos, existing detection techniques primarily focus on specific properties, such as head-movement [98] or specific high-profile individuals [2]. Potential general-purpose detection techniques are still actively being studied.

Moreover, studies which focus on generated images, primarily consider colour images. These studies often involve datasets containing the realistic celebrity images generated by [44], or images from StarGAN [17], which delivers high quality face manipulations. Mo et al. [65] proposed a highly accurate classifier aimed to distinguish real from generated faces. On a similar dataset, Jain et al. [40] split images into individual patches and combine a CNN with an SVM to evaluate the authenticity of individual patches in order to determine whether the entire image is authentic or fabricated. However, these solutions do not evaluate their networks on samples from unknown or updated GANs. Instead, their solutions are exclusively trained and evaluated on samples taken during a single, static training

state of a (known) GAN. Indeed, finding a general solution which detects samples from GANs that are not contained in the training set, has proven difficult even in the specific domain of fabricated facial images [48, 97, 53]. Although several studies have also found success on such a generalized setup [19, 67], these works still solely consider coloured, three-channel, RGB images.

Furthermore, existing work on tamper detection and localization techniques which are based on PRNU or other noise patterns, have not yet been evaluated on GANs [49, 20]. Similarly, Ghoneim et al. [28] have found success by utilizing noise patterns within medical images to detect forgeries. However, the authors have exclusively evaluated copy-move and image-splice forgeries. As such, this thesis is the first work that aims to detect GAN manipulations within medical images, while also evaluating the solution on multiple types of GANs.

7.2 Experimental setup

The experiments of this chapter feature two binary classification networks which are trained and evaluated on specific portions of the noise patterns extracted from CT slices. The networks classify their respective input as either legitimate or fake. The classifier of the first experiment will process the entire, uncropped noise pattern of a slice as input, while the classifier of the second experiment will scrutinize the extracted noise patterns of individual patches which previously depicted nodules. Figure 7.1 provides an example of the noise patches which are fed as input to the second classifier.

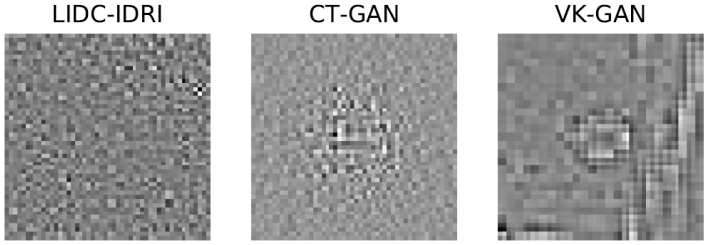


Figure 7.1: Examples of the extracted noise component of a nodule from the original LIDC-IDRI dataset (1), the CT-GAN (2) and VK-GAN (3).

The goal of the experiments is twofold. Firstly, the noise patterns, which are highly correlated to the device fingerprints, are evaluated on their capability to facilitate the detection of tampered images. Secondly, since the proposed solution should be GAN-agnostic, multiple GANs are evaluated to determine whether the detection is affected by the type of GAN.

Both experiments will again use the LIDC-IDRI dataset [4], which will be used by the classifiers to acquire a strict representation of legitimate data. Since the proposed solution (section 3.3) will make decisions based on the fingerprint of a single related CT scanner device, only slices from a specific manufacturer are considered legitimate. For the upcoming experiments, slices from *Philips* will be considered legitimate, while all other slices must be rejected. By labelling all non-Philips slices as illegitimate, the underlying detector should learn a stricter representation of the desired Philips fingerprint, without training on any samples from the GANs. Figure 7.2 presents the experimental setup for the detector. As can be seen, all samples, except LIDC-IDRI samples from Philips devices, are assigned label '1' which denotes these samples as illegitimate. After training, both detectors are evaluated on samples from two independently developed GANs. These networks, CT-GAN and VK-GAN, will be detailed in section 7.2.1 below.

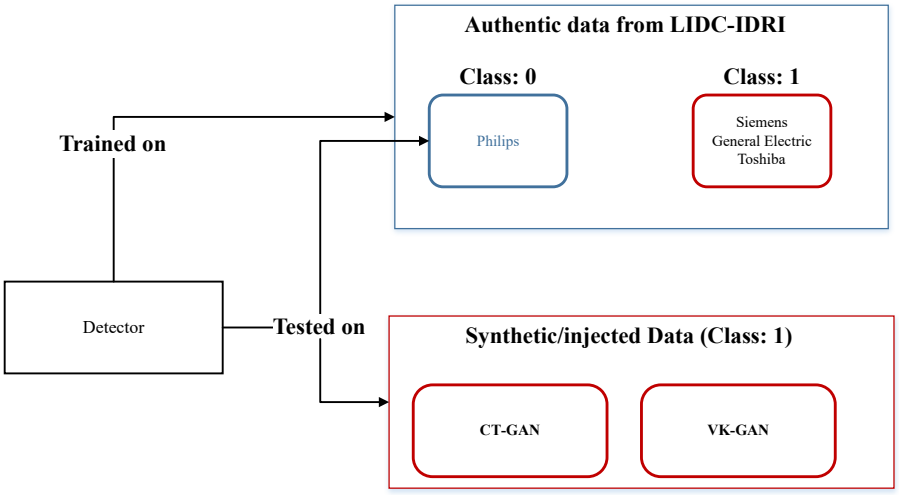


Figure 7.2: Training setup for the detector; only images from the original LIDC-IDRI dataset may be used for training.

Since the experiments in this thesis also aim to simulate the data discrepancy between attack and defender, previously discussed in section 3.4.3, the attacker (GANs) and defender (detector) are trained on different subsets of the data. As the defender aims to acquire a device fingerprint from Philips CT scanner devices, the attacker networks will only have relatively limited access to these samples. This disadvantage for the attacker should indicate if the defender's knowledge of the fingerprint indeed benefits the detection. Since the CT-GAN modifies existing slices, the setup will be as follows. During the training phase, the CT-GAN does not train on slices from Philips devices, while the injections themselves will still be exclusively performed on slices from Philips. Consequently, the tampered slice will primarily carry the Philips fingerprint, while the injected portion does not. Figure 7.3 illustrates this setup.

7.2.1 Generative Adversarial Networks

The experiments involve two independently developed GANs: a retrained version of the CT-GAN which was used in the original attack by Mirsky et al. [63], and a Conditional GAN (CGAN) that was initially developed for dataset augmentation to aid in nodule segmentation tasks. Although both GANs have been independently developed and trained for different purposes, both networks have been trained on the LIDC-IDRI dataset, and thus directly applicable to our experiments.

CT-GAN

The first GAN used within the experiments is a retrained CT-GAN which closely resembles the model from the original attack [63]. As the authors kindly provided their code on Github[1], it was possible to replicate the exact network architecture as well as the training procedure. However, as the experiments in this thesis also aim to simulate the data discrepancy between attacker and defender (discussed in section 3.4.3), the training data and attack pipeline have been modified. Specifically, the GAN is not trained on any samples from Philips devices. This approach should amplify the distortion of

[1] https://github.com/ymirsky/CT-GAN

the fingerprint as the trained GAN is tasked to inject nodules into slices of which the fingerprint is unknown (Figure 7.3).

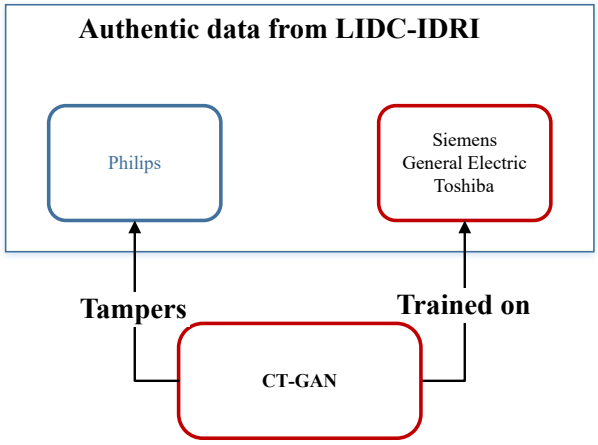


Figure 7.3: The CT-GAN is not trained on Philips slices to simulate the attacker's limitation in the device fingerprint. However, the tampered slices are indeed from Philips devices.

The original attack featured the ability to choose between nodule injection or removal. However, the experiments in this thesis exclusively make use of the injector model. This has been done because of the added complexity to acquire training samples for the remover model. Specifically, the authors employed a nodule detection algorithm to extract benign micro nodules from healthy CT scans. However, this approach was deemed too complex for the exploratory purposes of this thesis. Furthermore, as the proposed solution of this thesis should serve as a general countermeasure against GAN tampering, the type of modification is of less importance. As such, this thesis will only feature the injector model of the original attack. However, several other modifications to the original implementation have been made, which may affect the produced samples.

1. **Selection of training samples.** To train the model for this thesis, nodule coordinates from the LIDC-IDRI annotations [76] have been extracted and used to acquire the training samples. To simulate the data discrepancy between attacker and defender as described in section 3.4.3, the GAN is not trained on samples which originate from Philips devices. Although the extracted nodules had the same dimensions as the nodule sizes specified in the original attack ($10mm < x < 16mm$), the total amount of potential nodules differs (272 vs 169). This may be due to the use of a different set of annotations, or the use of additional filtering before the selection of training samples. As such, the training set is slightly different from the original attack. Despite the discrepancy in the training data, Figure 7.4 illustrates that the network was still able to train effectively.

2. **Training parameters.** For the training phase, the original authors extracted cubes with a dimension of $32 \times 32 \times 32$. The inner part of the cube is masked, and it is the objective of the GAN to inpaint the masked portion of the cube. The original work considered masks with a dimension of $16 \times 16 \times 16$[2], positioned in the middle of the extracted cube. However, during the experiments of this thesis, these parameters caused a substantial amount of failed injections on the test samples. As such, the parameters have been adapted to use a substantially larger

[2]The provided source code of the authors specified a different set of dimensions. Specifically, $20 \times 20 \times 20$.

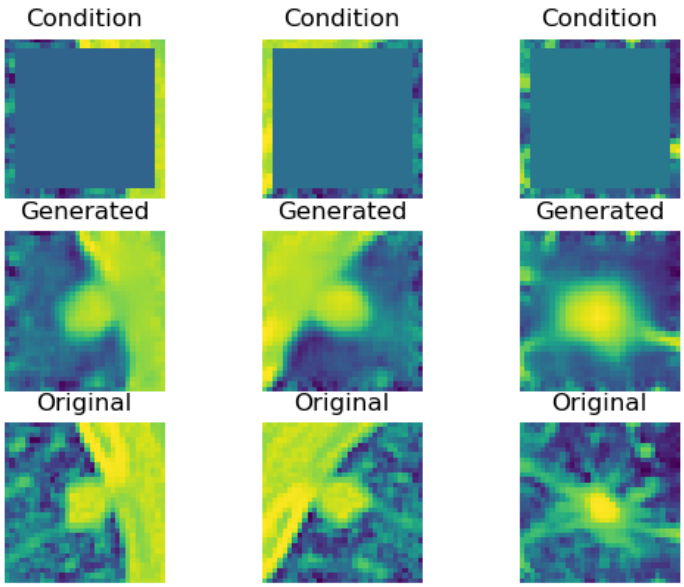


Figure 7.4: Samples taken during the training phase of CT-GAN. The first row depicts the input to the generator, while the second presents the generated results. The last row shows the unmasked image of the original sample.

mask of $28 \times 28 \times 28$ (Figure 7.5). Despite this more difficult objective, the second image of Figure 7.6 highlights the model's ability to perform noticeable injections.

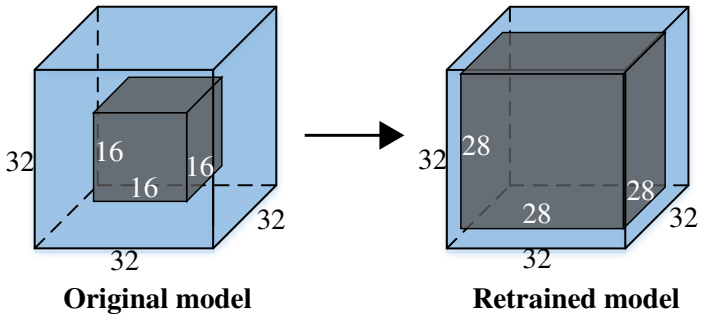


Figure 7.5: The modified mask that is used during the training phase of the GAN.

3. **Injection procedure.** During the injection of nodules, i.e. after the training phase, the cubes undergo an additional processing step to touch up the samples and make the produced nodules even more realistic. However, the calculations used during the touch up step consistently produced malformed values which negated the injection by altering all values within the tampered cube to 0. As a consequence, the injected, black cube essentially removed part of the image. As the samples used during this experiment come directly from the unprocessed DICOM images, it may the case that the authors performed additional preprocessing that is not mentioned in the original paper. To remedy the situation, a constant in the calculation has been adapted to match

the values of the original LIDC-IDRI images.

In addition, the original work injected the entire cube into a CT volume. However, as all the experiments in this thesis process individual slices, only the middle slice within the cube is used for the injection. Nevertheless, the GAN is still trained in its original 3D setting.

Despite these modifications to the original implementation, the model still produced samples which closely resemble the original CT-GAN attack. The second image in Figure 7.6 presents an example nodule generated by the retrained CT-GAN model.

Figure 7.6: A real nodule from the original LIDC-IDRI dataset (1), and generated nodules by CT-GAN (2) and VK-GAN (3).

VK-GAN

The GAN provided by van Kampen[3] is also a CGAN, based on the architecture of Isola et al. [39] and Costa et al. [18]. The network was initially trained to assist in lung nodule segmentation tasks. Specifically, the GAN produces whole images of chest CT slices. This includes the nodules, which are placed at predetermined, realistic locations. Although the network performs inconsistently with respect to the generation of anatomically-correct lungs, the nodules are generally produced with sufficient quality. Figure 7.7 depicts examples of the entire images produced by VK-GAN. Although the slices possess a large amount of irregularities, the nodules remain distinguishable.

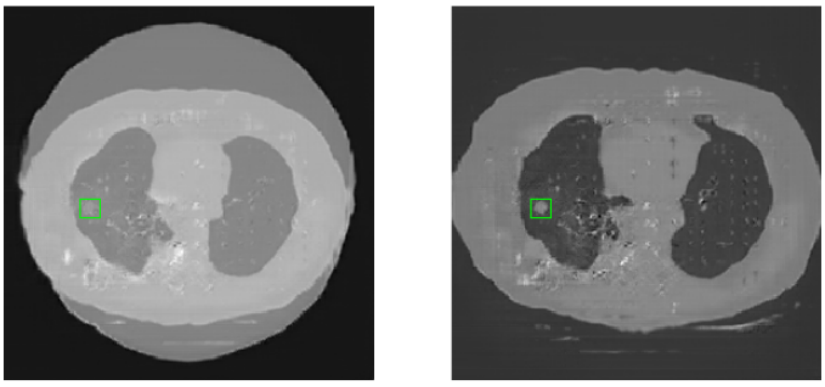


Figure 7.7: Slices generated by the VK-GAN. The green marker highlights the nodule. Images taken from the original work.

The network was exclusively trained on samples from the LIDC-IDRI dataset, including samples from Philips. Although this certainly affects the assumption on the data discrepancy of the attacker

[3]The pretrained model was graciously provided by the author, and is currently pending publication.

and defender, it is highly likely that the initial objective, as well as the large mix of data, prevented the model to learn an accurate and complete fingerprint of Philips devices. As such, the defender still has considerably more knowledge of a specific Philips device fingerprint than the attacker. Although the difficult objective of the VK-GAN leads to inconsistent quality of the produced samples, this contrast between the CT-GAN and VK-GAN provides an additional point of interest. Specifically, the gap in quality should cause the proposed solution to be more successful in detecting fabricated samples from the VK-GAN than the CT-GAN. The third image of Figure 7.6 depicts a nodule produced by the VK-GAN. Although the quality is notably worse, the nodule is still clearly recognizable.

7.2.2 Detector Networks

The independent detector networks used during the experiments are also neural networks, trained in a supervised binary classification setting. Since the detectors each process different input samples, their respective architectures also differ from each other.

Detection on Individual CT slices

The classifier of the first experiment will process the entire, uncropped noise pattern of a slice as input. The input of this classifier is identical to the input of the successful classifier used in section 6.3.2 of the previous chapter. As such, the same architecture has been applied for this experiment as well. However, to account for the binary classification task, the top layer of the architecture has been changed to a single neuron with the sigmoid function in the final activation layer. The original network architecture, with 4 neurons as the top layer, is described in section 6.3.2 and illustrated in Figure 7.8.

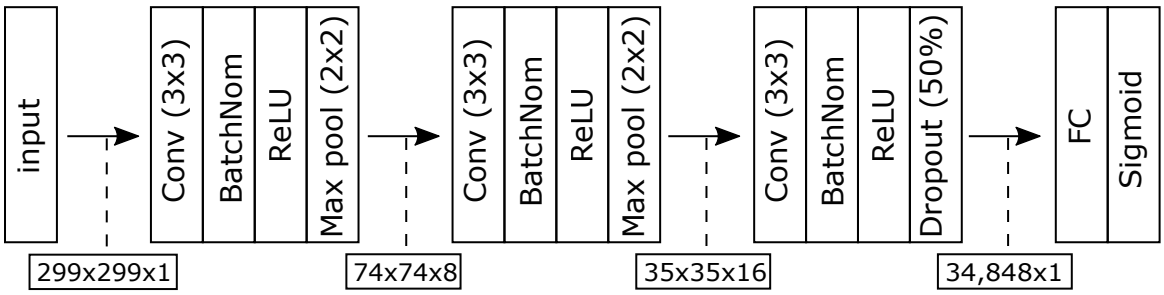


Figure 7.8: The CNN architecture used for the detection of tampered CT slices.

Detection of Injected Nodules

The classifier of the second experiment is exclusively trained on noise patterns of patches which originally depicted nodules. This approach is motivated by the smaller scope of the problem; the relative size of a potential distortion is much larger in individual patches than for an entire CT slice. By purely training the classifier on the underlying noise pattern of patches which contain nodules, the classifier is more likely to detect distortions caused by the fabricated nodules of the GANs. For example, the CT-GAN, given its training setup (Figure 7.3), is likely to either mimic noise patterns from the non-Philips devices, or even embeds its own fingerprint. As such, the synthetic nodules are more likely to be rejected. The patches depicted in Figure 7.1 provide an example of noise patterns from the different sources used during the experiments. Given the desired practicality and independence of the proposed solution (section 3.3), it is important to note that the current setup is still a realistic setting

in practice. A nodule segmentation network could first extract all nodules present in the scan, while a second network extracts the noise pattern and determines the authenticity.

For this experiment, the data samples for the nodules are obtained from the LIDC-IDRI Nodule Size Report [76], which contains annotations of the nodules present in the LIDC-IDRI dataset, including the diameter and location of each nodule. However, as only a small subset of the slices contain a nodule, there is substantially less training data available for the detector. The total dataset shrinks to a little over 1200 samples; with only 200 nodules captured from Philips devices. Although the classifier in this experiment focuses on a smaller problem, the experimental setup presented in Figure 7.2 remains the same. Specifically, the legitimate samples, coming from Philips devices are labelled '0', while all other samples are assigned label '1'.

Neural Network Architecture

As the dimensionality, as well as the amount of data itself is significantly decreased, a new architecture is picked. First iterations of the experiment considered the use of a simple architecture proposed for the MNIST dataset since the image dimensions are relatively similar ($50 \times 50 \times 1$ vs. $32 \times 32 \times 1$). However, given the small amount of data available, this architecture was also deemed to large. The final architecture for this task possesses the same structure, but has less convolutional layers. Figure 7.9 illustrates the shallower architecture, along with its input images.

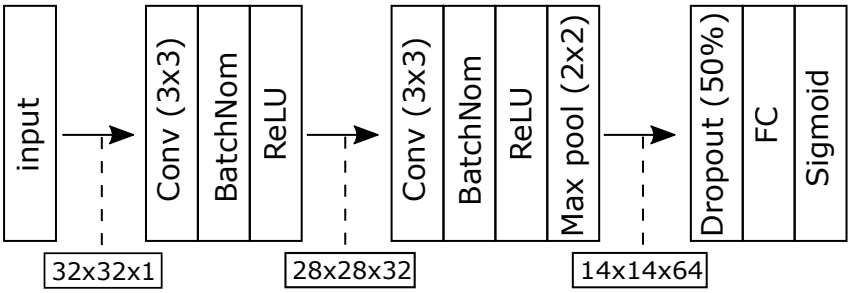


Figure 7.9: The CNN architecture used for the classification of injected nodules.

7.3 Results

Although the aforementioned networks are trained and evaluated on separate types of data samples, the experimental setup as visualized in Figure 7.2 persists for both experiments. Specifically, during the evaluation, slices, or nodules, captured by Philips devices are considered legitimate (labelled '0'), while the samples from the evaluated GANs are labelled '1'. This section details the result of the evaluation phase.

7.3.1 Detection on Individual CT slices

The results are presented in Table 7.1 and Figure 7.10. As can be seen, the samples of VK-GAN are all detected perfectly, while the injections by the CT-GAN are not detected at all. Despite the significant gap, these results are not entirely surprising. The VK-GAN will, if at all, possess a fingerprint that contains artefacts from all possible manufacturers contained in the LIDC-IDRI dataset (Philips, Siemens, General Electric, Toshiba). As such, the Philips fingerprint will only have a rela-

GAN	Accuracy	Precision
CT-GAN	49.15%	0%
VK-GAN	99.15%	98.33%
Average	74.15%	49.16%

Table 7.1: Detection results on full-sized CT slices.

tively minor presence, prompting the detector to correctly classify the samples from the VK-GAN as 'Non-Philips'.

The same is likely to hold true for the injections from the CT-GAN. The majority ($> 99\%$) of the slice will carry the Philips fingerprint, while only a small portion is distorted. Since the main goal of the detector classifier is to generalize over the data, this distortion is evidently not sufficient to reject the tampered slices, and consequently flip the label from '0' (Philips) to '1' (non-Philips).

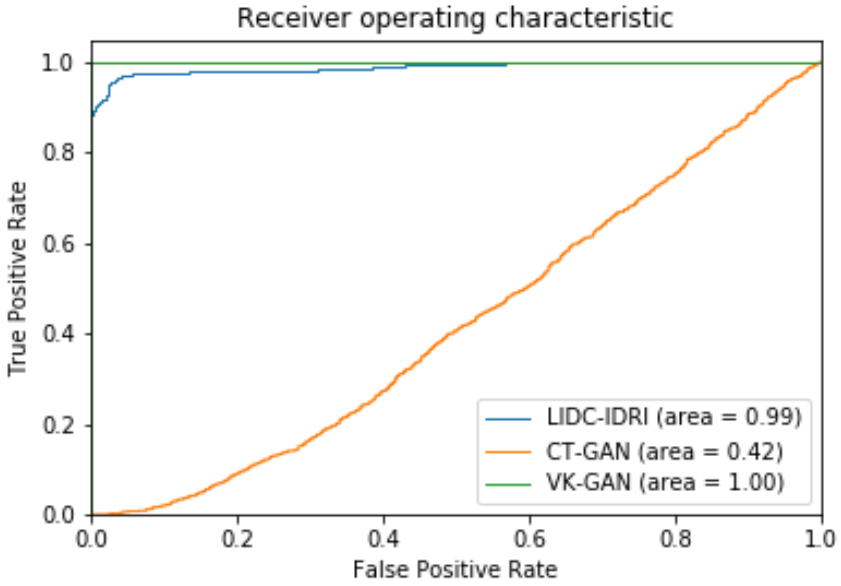


Figure 7.10: The AUROC for the detection on full CT slices.

7.3.2 Detection of Injected Nodules

The results presented in Table 7.2 and Figure 7.11 show that with a smaller scope, samples from both GANs are now detected with a high accuracy. However, the obtained results may not be an accurate reflection of the performance within a practical setting. Specifically, the classifier exhibited unexpected behaviour during the training phase. As presented in Figures C.1 and C.2 of appendix C, the classifier was able to quickly fit on the training data, yet failed to make accurate predictions on the validation data (60%). As such, the results presented in Table 7.2 significantly diverge from the expected testing performance, and should be taken with skepticism.

A likely explanation may be that the classifier, although having a relatively simple setup, is still too large or simply not suitable for the current task. This may be due to the type and dimensions of

GAN	Accuracy	Precision
CT-GAN	84.09%	89.47%
VK-GAN	95.45%	91.66%
Average	89.77%	90.57%

Table 7.2: Detection results on individual patches which contain a nodule.

the images, but may also be caused by the small amount of available data. Another, albeit unlikely, explanation may be that there is only subtle difference between the nodules created by the various *real* devices, and the decision boundaries may be much closer together now. As such, it may be the case that, even though real nodules from the LIDC-IDRI dataset are nearly indistinguishable, the generated nodules from the GANs are confidently predicted as non-Philips by the classifier.

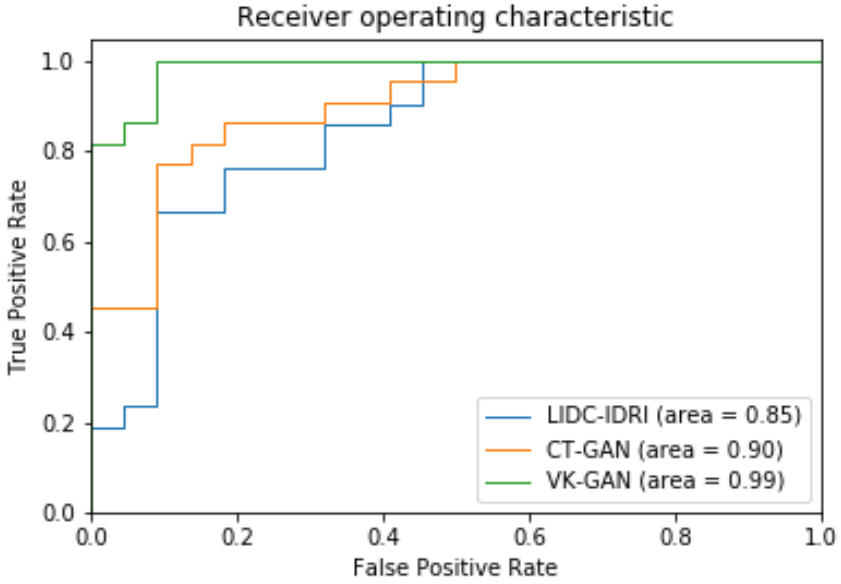


Figure 7.11: The AUROC for the detection of tampered nodules.

7.4 Discussion

The experiments in this chapter have shown that detecting manipulations performed by GANs remains a challenging task. Specifically, a simple binary classifier, which examines the entirety of a CT slice, is not suitable to detect the manipulations injected by the CT-GAN. However, the perfect detection rate on slices from the VK-GAN indicates that larger distortions may not go unnoticed. As such, if the CT-GAN would need to distort multiple patches within an image, the attack may be easier to detect. It would be interesting for future research to investigate how the amount and size of the manipulations affect the overall detection rate. Nevertheless, for a detector that obtains full slices as input, a sliding window [40] or localization technique [20] will likely yield better performance.

Analogously, the detection of tampered nodules shows more promising results. By reducing the input dimensions and scope of the problem, the relative size of the potential distortion is significantly

higher. As such, the classifier is more likely to detect any tampered samples. However, section 7.3.2 highlighted that the chosen architecture of the classifier is likely suboptimal. Although the classifier accurately detects the manipulations of either GAN with an average accuracy of 89.77%, the unsatisfactory performance on unaltered LIDC-IDRI samples make the overall results less reliable. As such, further research, with an optimized network, and a larger dataset, would be required to verify the true potential of the classifier.

Finally, unsupervised learning may be an alternative approach to detect the tampered images. This approach would treat the problem as an anomaly detection [14], or one-class classification [13] problem, as opposed to binary-classification. The goal of the network then becomes to learn an accurate representation of the fingerprint by exclusively training on Philips data. Samples from GANs or the remaining classes of the LIDC-IDRI dataset are likely to diverge from this representation which consequently facilitates the final detection.

7.5 Conclusion

This chapter detailed the experiments which evaluated the potential benefit of device fingerprints to facilitate the detection of tampered CT images. The two experiments treated the problem as a binary-classification objective; a sample is either authentic, or tampered. The first experiment evaluated a classifier which receives noise patterns of entire CT slices as input, while the second experiment featured a classifier which exclusively examined the noise pattern of patches which contain nodules. The experiments featured two independently developed GANs, CT-GAN and VK-GAN, which were used to produce the tampered images and synthetic nodules. In both experiments, the detector was solely trained on samples from the LIDC-IDRI dataset. During the subsequent evaluation phase, samples from both GANs were added to the test set to evaluate the performance of the classifiers.

Section 7.3 presented the results of the classifiers. The classifier of the first experiment, which made decisions based on entire CT slices, was not able to detect any samples tampered by CT-GAN, yet managed to correctly classify all the samples produced by the VK-GAN. Previous section 7.4 argued that the manipulations performed by the CT-GAN are likely not large enough to be picked up by such a classifier.

With an average accuracy of 89.77% and 90.57% precision, the results of the second experiment demonstrated that a narrower scope of the problem yields more promising results. Indeed, the classifier which exclusively focused on patches containing nodules reached substantially higher performance than the classifier based on uncropped CT slices. However, as already discussed in sections 7.3.2 and 7.4, the result should be taken with some skepticism, and a more extensive study is required to validate the results.

Although the second experiment highlighted that a binary classifier has the potential to detect tampered samples, an alternative approach may also be suitable for the task. Specifically, the use of an unsupervised learning algorithm will prompt a neural network to exclusively learn an accurate representation of legitimate samples. Other samples, such as the tampered images, will only fit the learned representation to a limited extent, making the tampered samples stand out from the legitimate ones.

7.5.1 Answer to Research Questions

Finally, This chapter will answer the final research questions introduced in section 4.1.

R3: Can device fingerprints be leveraged to support the detection of artificially generated CT imagery? The experiments certainly highlighted the potential of device fingerprints in supporting the detection of tampered CT images. The results of the first experiment furthermore indicate that larger distortions may lead to a higher detection rate. However, only two extreme cases were featured: a single distorted patch and an entirely distorted image. As such, further experiments are required to verify this hypothesis. Nevertheless, the second experiment showed that a binary classifier may indeed detect tampered images, even if the classifier is not trained on tampered samples.

R4: Is the detection affected when a wider range of generative adversarial networks are evaluated? The second experiment showed that there is indeed a significant difference in the detection rate of samples from different GANs. However, further research is required to investigate which elements contribute to this gap. Potential factors may be the quality of the produced samples, or the training data of the GAN which may affect the fingerprints embedded in the samples.

Chapter 8

Discussion and Conclusion

This thesis was motivated by the significance of the recently published CT-GAN attack [63]. This attack demonstrated how malicious GANs may be deployed in practice to tamper with medical images by selectively injecting or removing malign nodules within CT scans. The attack furthermore highlighted that hospitals, due to their criticality and complexity, remain vulnerable to various security breaches. This thesis identified that a pragmatic countermeasure, which does not rely on defence in depth, does not yet exist, and consequently proposed a solution which identifies GAN-tampered images solely based on image data. Specifically, the solution implicitly learns definitive noise patterns from CT scanners, i.e. device fingerprints, which are inadvertently embedded within all authentic CT images by the devices themselves. Distortions to this device fingerprint may then indicate a tampering attack. Several experiments have been performed in this thesis which aimed to determine if such device fingerprints do indeed exist for CT scanners, and whether these fingerprints may be leveraged to detect tampered images.

This final chapter of the thesis will briefly revisit the experiments, their results, and discuss the answers to the research questions that were defined in chapter 4. This chapter will furthermore evaluate the significance of these results for the proposed solution. In addition, several potential directions for future work are proposed which should further improve the results presented in this thesis. Finally, the chapter is concluded with a few personal words on some recurring elements that influenced this project and the resulting thesis.

8.1 Discussion

Chapters 6 and 7 have performed multiple experiments to answer the research questions defined in chapter 4 and help evaluate the proposed solution from chapter 3; a classifier which detects tampered images based on distortions in device fingerprints, which are naturally embedded by CT scanners. As only little related work exists on the presence of CT scanner fingerprints, this thesis introduced new experiments which aimed to verify this claim.

The presented results from the scanner classification tasks (chapter 6) showed that, with an accuracy of 92.5%, scanners from a variety of manufacturers are indeed accurately distinguishable from one another, and that the usage of noise patterns further boosts the performance to 93.6%. These experiments positively answered research questions R1 and R2, and confirmed that CT scanners indeed carry a certain notion of a device fingerprint.

The subsequent experiments from 7 then set out to answer research questions R3 and R4. Specifically, R3 posed the initial question if device fingerprints are reliable instruments to detect CT image tampering by GANs. The subsequent question R4 motivated the investigation into tampered images from an additional GAN to evaluate whether the detection by device fingerprints is suited as a general-purpose countermeasure.

The results presented in section 7.3 have shown that a classifier which processes full-sized CT scans as input is likely unsuited to detect the small sections distorted by the CT-GAN. However, given the stellar performance on the detection of images by the VK-GAN, the effectiveness of the classifier may improve when the number and size of te distortions within an image grow. Similarly, the results of section 7.3.2 indicate that by narrowing the scope of the classifier, and applying the detection solely on patches which contain nodules, the overall detection performance rises significantly to an average accuracy of 89.77%, with a precision of 90.57%. The individual results in Table 7.2 furthermore show that, although the type of GAN does indeed affect the overall detection rate, the classifier can still reliably detect either distortion. However, given the irregularity of the results, further experiments are required to confirm the presented performance.

To conclude, the proposed solution certainly has the potential for deployment. The extraction of definitive noise patterns have consistently shown to be a beneficial set of features to identify groups of CT scanner devices, and may also be utilized to facilitate the detection of image tampering attacks. However, given the poor detection rate on full-sized CT images, the solution may be modified as a sliding-window detector, which scrutinizes individual patches of a CT image before making an assessment on the entire scan. Alternatively, the solution may be jointly deployed with a nodule segmentation network which extracts the relevant portions from a scan. However, this latter approach would be unable to detect nodule removal attacks, which have not been considered in this thesis.

8.2 Conclusion

The recently published CT-GAN attack demonstrated a malicious application of a GAN, capable of autonomous tampering of CT scans. Although existing techniques, such as digital signatures, are excellent measures to ensure image integrity, they are generally not deployed in hospitals due to the complexity of a successful implementation. This thesis proposed a novel solution, specifically tailored to verify the integrity of CT scans, and detect tampering attacks by GANs. The solution acts independently of existing infrastructure, and solely requires image data from the CT scans to make its assessment. This independence facilitates the painless deployment within hospitals which may otherwise choose not to implement any security controls which verify the integrity of medical images. The solution harnesses definitive noise patterns present in CT images which are produced by all authentic CT scanners. Significant distortions in these fingerprints would indicate an attack.

This thesis furthermore introduced several research question which were aimed to evaluate the effectiveness of the solution. Specifically, the questions, and consequent experiments, investigated if the extraction of a definitive noise pattern is indeed analogous to the device fingerprint of a CT scanner. The experiments showed that CT scanner manufacturers may be classified with 92.5% accuracy based on their produced images. The accuracy further increased to 93.6% by using the noise patterns as input instead. These results have shown that CT scanners indeed posses a certain notion of a device fingerprint.

The final set of experiments, performed in chapter 7, then evaluated if the fingerprints may facilitate

the detection of image tampering by GANs. The presented results indicate that the fingerprints are indeed a promising tool to detect image tampering within CT scans. Although a classifier cannot accurately detect small distortions within a complete CT slice, the detector may nevertheless distinguish injected, generated nodules from legitimate ones. With an average accuracy of 89.77%,and 90.57% precision, the solution certainly has the potential to be deployed in the future.

8.3 Future Work

Although the proposed solution has certainly demonstrated potential, this thesis and its experiments are first and foremost exploratory work. As such, there are still several avenues to investigate that are likely to improve the results.

Network optimization. All of the neural networks applied in the experiments are based on existing architectures and setups, with slight modifications to suit the dimensions and the amount of data. As such, there is still ample opportunity to improve the existing classifiers. Specifically the classifier used in the experiments of chapter 7.3.2 showed irregular performance, which indicates a potential design flaw in the architecture of the network. Moreover, even the successful networks used for the experiments in chapter 6 may certainly show improved performance when additional domain expertise is applied to enhance the neural networks.

Optimization of the device fingerprint. In addition to the network optimization, the extracted noise pattern may also be substantially optimized. The current noise pattern to establish the fingerprint was originally intended for ordinary photo cameras, yet has also shown applicability for CT scanners. However, Duan et al. [22] have already shown improvements with a more complex device fingerprint, which is based on the reconstruction algorithm of CT scanners. Applying a similarly advanced technique for establishing the device fingerprint will likely improve CT scanner classification as well as image tampering detection.

Alternative approach to detect tampered images. As already briefly discussed in section 7.4, the experiments in this thesis approach the detection of image tampering as a binary classification problem. However, given the role of the device fingerprint to aid in the detection, an alternative approach may be to apply anomaly detection, or one-class classification to detect distortions within fingerprints [13, 14]. Figure 8.1 presents a first step towards anomaly detection with a simple, fully-connected autoencoder (2 hidden layers, and a 2D layer for the code). Although there is still significant overlap in the reconstruction losses, the legitimate Philips samples nevertheless have the smallest variance. As such, a more sophisticated approach may be a promising next step. Alternatively, related works from the field of image forensics, primarily focus on localization, rather than binary classification. For example, Cozzolino and Verdoliva [20] propose an effective detection tool, also based on fingerprints, which highlights abnormalities within images. Although this approach was deemed to complex for this thesis, it is highly interesting for future work.

Deployment of the proposed solution. Since the experiments have shown that the detection based on full images is likely ineffective, the solution may be adapted to achieve higher performance. For example, a classifier similar to [40] would first assess individual, non-overlapping patches, while a subsequent classifier accumulates the individual classifications to make the final decision. A similar approach may be used to learn the local fingerprint of individual patches, and then applying a classifier

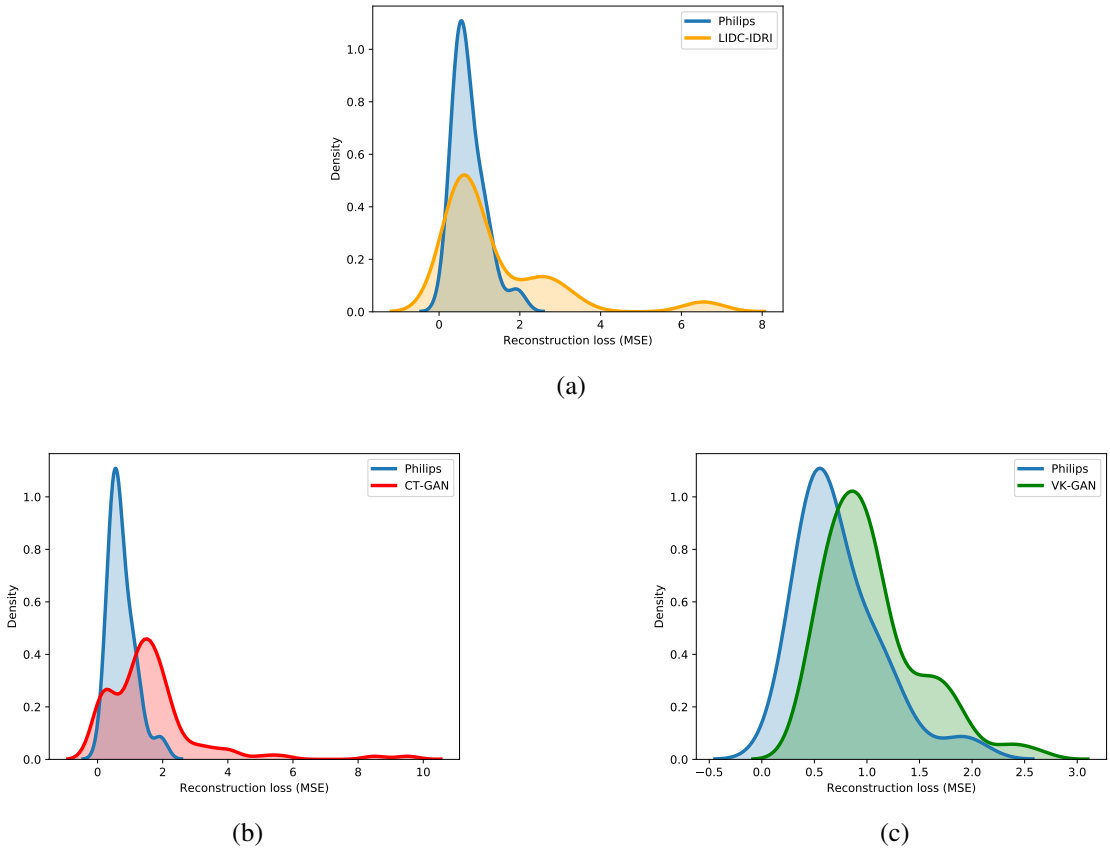


Figure 8.1: Reconstruction losses of an autoencoder trained on nodules from Philips images.

on each patch. The same approach could also be adapted to an anomaly detection problem for each patch. It is important to note that neither approach would interfere with the independence of the solution, as the cumulative input (the CT slice) remains the same.

8.4 Retrospective of the Thesis

Finally, as the last section of this thesis, I would like to share a few aspects that influenced the overall progress and final results of the thesis.

Ensuring continuity of the project. I noticed it is awfully tempting to get stuck on improving intermediate steps, even when the main goal has not been reached yet. Although the thesis started with a wide scope, the goal of the thesis became quite clear in the early stages of the process: we wanted to detect tampered images by harnessing the device fingerprints of CT scanners. However, in order to make the entire thesis meaningful, numerous intermediate steps had to be performed first: finding a potential solution, data acquisition, extraction of possible fingerprints, verifying if these fingerprint are indeed beneficial, subsequent CT scanner experiments, and finally using the fingerprints for detection. In hindsight, I lost valuable time by continuously improving and evaluating the experiments

of chapter 6, while neglecting the final goal of detecting tampering attacks. Without the guidance of my supervisors, I would have been stuck a lot longer.

Finding the right timing for decisions and trade-offs. This aspect is perhaps related to the previous point regarding project continuity. Although the amount of literature, and number of potential solutions for a specific problem is often vast, a decision has to be made at some point. Waiting too long consumes valuable time of the project, while impulsive decision-making may lead to wrong decisions. As an example, I was highly concerned with the integrity of the experiments. I poured a lot of time into finding the correct networks and suitable experimental setups, backed up by literature or domain experts (again, mostly for 6). Although these considerations lead to exceptionally good results, they also consumed a lot of time. This lack of time then solely gave us the opportunity to evaluate the tampering detection as a binary classification problem. Although it is still a valid approach, I would have loved to have more time to treat the issue as an anomaly detection problem as well.

Data acquisition. The experiments within this thesis required a sizeable amount of suitable data, i.e. CT scans. However, with the introduction of the General Data Protection Regulation (GDPR)[1] within the EU, acquiring and processing medical data has become challenging, especially within a large organization such as Philips. Even the usage of an publicized and anonymized dataset, such as LIDC-IDRI, is only approved under strict conditions. Although it was possible to use the dataset in the end, the experiments consumed more time than initially anticipated.

Data processing. Besides the data acquisition, the processing of the data was not painless either. The entirety of the LIDC-IDRI dataset contained a little over 120GB of data; loading the entire dataset into memory is not at all possible, and individual processing of slices may take up to 12 hours. As such, I have implemented an abundance of functions, often implementing multicore processing, to optimally load and process the data. Furthermore, using the data for deep learning was not a small feat either. As existing functions also expect the data to be entirely loaded into memory (or streamed), splitting data, batch generation, and even model evaluation all had to be manually implemented with great care.

[1] `https://eur-lex.europa.eu/eli/reg/2016/679/oj`

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 308–318.

[2] Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., and Li, H. "Protecting World Leaders Against Deep Fakes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 38–45.

[3] Akhtar, N. and Mian, A. "Threat of adversarial attacks on deep learning in computer vision: A survey". In: *IEEE Access* 6 (2018), pp. 14410–14430.

[4] Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., Clarke, L. P., et al. "Data From LIDC-IDRI. The Cancer Imaging Archive." In: (2015). URL: http://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX.

[5] Bappy, J. H., Roy-Chowdhury, A. K., Bunk, J., Nataraj, L., and Manjunath, B. "Exploiting spatial structure for localizing manipulated image regions". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4970–4979.

[6] Basnet, R., Mukkamala, S., and Sung, A. H. "Detection of phishing attacks: A machine learning approach". In: *Soft Computing Applications in Industry*. Springer, 2008, pp. 373–383.

[7] Bayar, B. and Stamm, M. C. "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection". In: *IEEE Transactions on Information Forensics and Security* 13.11 (2018), pp. 2691–2706.

[8] Breiman, L. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[9] Burges, C. J. "A tutorial on support vector machines for pattern recognition". In: *Data mining and knowledge discovery* 2.2 (1998), pp. 121–167.

[10] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., and Madry, A. "On evaluating adversarial robustness". In: *arXiv preprint arXiv:1902.06705* (2019).

[11] Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. "Hidden voice commands". In: *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 2016, pp. 513–530.

[12] Carlini, N. and Wagner, D. "Towards evaluating the robustness of neural networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 39–57.

[13] Chalapathy, R. and Chawla, S. "Deep learning for anomaly detection: A survey". In: *arXiv preprint arXiv:1901.03407* (2019).

[14] Chalapathy, R., Menon, A. K., and Chawla, S. "Anomaly detection using one-class neural networks". In: *arXiv preprint arXiv:1802.06360* (2018).

[15] Chen, M., Fridrich, J., Goljan, M., and Lukás, J. "Determining image origin and integrity using sensor noise". In: *IEEE Transactions on information forensics and security* 3.1 (2008), pp. 74–90.

[16] Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. "Using recurrent neural network models for early detection of heart failure onset". In: *Journal of the American Medical Informatics Association* 24.2 (2016), pp. 361–370.

[17] Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8789–8797.

[18] Costa, P., Galdran, A., Meyer, M. I., Abràmoff, M. D., Niemeijer, M., Mendonça, A. M., and Campilho, A. "Towards adversarial retinal image synthesis". In: *arXiv preprint arXiv:1701.08974* (2017).

[19] Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., and Verdoliva, L. "ForensicTransfer: Weakly-supervised Domain Adaptation for Forgery Detection". In: *arXiv preprint arXiv:1812.02510* (2018).

[20] Cozzolino, D. and Verdoliva, L. "Noiseprint: a CNN-based camera model fingerprint". In: *IEEE Transactions on Information Forensics and Security* (2019).

[21] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[22] Duan, Y., Bouslimi, D., Yang, G., Shu, H., and Coatrieux, G. "Computed Tomography Image Origin Identification Based on Original Sensor Pattern Noise and 3-D Image Reconstruction Algorithm Footprints". In: *IEEE journal of biomedical and health informatics* 21.4 (2017), pp. 1039–1048.

[23] Dwork, C. "Differential privacy". In: *Encyclopedia of Cryptography and Security* (2011), pp. 338–340.

[24] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (2017), p. 115.

[25] Fredrikson, M., Jha, S., and Ristenpart, T. "Model inversion attacks that exploit confidence information and basic countermeasures". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2015, pp. 1322–1333.

[26] Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing". In: *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 2014, pp. 17–32.

[27] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification". In: *Neurocomputing* 321 (2018), pp. 321–331.

[28] Ghoneim, A., Muhammad, G., Amin, S. U., and Gupta, B. "Medical image forgery detection for smart healthcare". In: *IEEE Communications Magazine* 56.4 (2018), pp. 33–37.

[29] Goodfellow, I. J., Shlens, J., and Szegedy, C. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[30] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[31] Goodfellow, I., Papernot, N., Huang, S., Duan, R., Abbeel, P., and Clark, J. *Attacking Machine Learning with Adversarial Examples*. Feb. 2017. URL: `https://openai.com/blog/adversarial-example-research/`.

[32] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

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

[34] Hitaj, B., Ateniese, G., and Perez-Cruz, F. "Deep models under the GAN: information leakage from collaborative deep learning". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2017, pp. 603–618.

[35] Hu, W. and Tan, Y. "Generating adversarial malware examples for black-box attacks based on GAN". In: *arXiv preprint arXiv:1702.05983* (2017).

[36] Hu, W. and Tan, Y. "Generating adversarial malware examples for black-box attacks based on GAN". In: *arXiv preprint arXiv:1702.05983* (2017).

[37] Huang, H., Coatrieux, G., Shu, H., Luo, L., and Roux, C. "Blind integrity verification of medical images". In: *IEEE transactions on information technology in biomedicine* 16.6 (2012), pp. 1122–1126.

[38] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. "Adversarial examples are not bugs, they are features". In: *arXiv preprint arXiv:1905.02175* (2019).

[39] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[40] Jain, A., Singh, R., and Vatsa, M. "On Detecting GANs and Retouching based Synthetic Alterations". In: *arXiv preprint arXiv:1901.09237* (2019).

[41] Jin, D., Xu, Z., Tang, Y., Harrison, A. P., and Mollura, D. J. "CT-realistic lung nodule simulation from 3D conditional generative adversarial networks for robust lung segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 732–740.

[42] Juuti, M., Szyller, S., Dmitrenko, A., Marchal, S., and Asokan, N. "PRADA: protecting against DNN model stealing attacks". In: *arXiv preprint arXiv:1805.02628* (2018).

[43] Kang, M.-J. and Kang, J.-W. "Intrusion detection system using deep neural network for in-vehicle network security". In: *PloS one* 11.6 (2016), e0155781.

[44] Karras, T., Aila, T., Laine, S., and Lehtinen, J. "Progressive growing of gans for improved quality, stability, and variation". In: *arXiv preprint arXiv:1710.10196* (2017).

[45] Kharboutly, A., Puech, W., Subsol, G., and Hoa, D. "Advanced sensor noise analysis for CT-scanner identification from its 3D images". In: *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE. 2015, pp. 325–330.

[46] Kharboutly, A., Puech, W., Subsol, G., and Hoa, D. "CT-Scanner identification based on sensor noise analysis". In: *2014 5th European Workshop on Visual Information Processing (EUVIP)*. IEEE. 2014, pp. 1–5.

[47] Kharboutly, A., Puech, W., Subsol, G., and Hoa, D. "Improving sensor noise analysis for CT-Scanner identification". In: *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE. 2015, pp. 2411–2415.

[48] Khodabakhsh, A., Ramachandra, R., Raja, K., Wasnik, P., and Busch, C. "Fake Face Detection Methods: Can They Be Generalized?" In: *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE. 2018, pp. 1–6.

[49] Korus, P. and Huang, J. “Multi-scale analysis strategies in PRNU-based tampering localization”. In: *IEEE Transactions on Information Forensics and Security* 12.4 (2016), pp. 809–824.

[50] Kurakin, A., Goodfellow, I., and Bengio, S. “Adversarial examples in the physical world”. In: *arXiv preprint arXiv:1607.02533* (2016).

[51] LeCun, Y., Bengio, Y., et al. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[52] Lee, S.-B., Jeong, E.-J., Son, Y., and Kim, D.-J. “Classification of computed tomography scanner manufacturer using support vector machine”. In: *2017 5th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE. 2017, pp. 85–87.

[53] Li, H., Chen, H., Li, B., and Tan, S. “Can Forensic Detectors Identify GAN Generated Images?” In: *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE. 2018, pp. 722–727.

[54] Lukáš, J., Fridrich, J., and Goljan, M. “Digital camera identification from sensor pattern noise”. In: *IEEE Transactions on Information Forensics and Security* 1.2 (2006), pp. 205–214.

[55] Marra, F., Gragnaniello, D., Verdoliva, L., and Poggi, G. “Do gans leave artificial fingerprints?” In: *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE. 2019, pp. 506–511.

[56] Menegola, A., Fornaciali, M., Pires, R., Avila, S., and Valle, E. “Towards automated melanoma screening: Exploring transfer learning schemes”. In: *arXiv preprint arXiv:1609.01228* (2016).

[57] Meng, R., Cui, Q., and Yuan, C. “A survey of image information hiding algorithms based on deep learning”. In: *Computer Modeling in Engineering & Sciences* 117.3 (2018), pp. 425–454.

[58] Metz, C. and Collins, K. *How an A.I. ’Cat-and-Mouse Game’ Generates Believable Fake Photos*. Ed. by Times, T. N. Y. [Online; posted 02-January-2018]. Jan. 2018. URL: https://www.nytimes.com/interactive/2018/01/02/technology/ai-generated-photos.html/.

[59] Mihcak, M. K., Kozintsev, I., and Ramchandran, K. “Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising”. In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*. Vol. 6. IEEE. 1999, pp. 3253–3256.

[60] Miller, T. “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence* 267 (2019), pp. 1–38.

[61] Miotto, R., Li, L., Kidd, B. A., and Dudley, J. T. “Deep patient: an unsupervised representation to predict the future of patients from the electronic health records”. In: *Scientific reports* 6 (2016), p. 26094.

[62] Miotto, R., Wang, F., Wang, S., Jiang, X., and Dudley, J. T. “Deep learning for healthcare: review, opportunities and challenges”. In: *Briefings in bioinformatics* 19.6 (2017), pp. 1236–1246.

[63] Mirsky, Y., Mahler, T., Shelef, I., and Elovici, Y. “CT-GAN: Malicious Tampering of 3D Medical Imagery using Deep Learning”. In: *CoRR* abs/1901.03597 (2019). arXiv: 1901.03597. URL: http://arxiv.org/abs/1901.03597.

[64] Mirza, M. and Osindero, S. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[65] Mo, H., Chen, B., and Luo, W. "Fake faces identification via convolutional neural network". In: *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. ACM. 2018, pp. 43–47.

[66] Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. "Universal adversarial perturbations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1765–1773.

[67] Nataraj, L., Mohammed, T. M., Manjunath, B., Chandrasekaran, S., Flenner, A., Bappy, J. H., and Roy-Chowdhury, A. K. "Detecting GAN generated Fake Images using Co-occurrence Matrices". In: *arXiv preprint arXiv:1903.06836* (2019).

[68] News, B., ed. *NHS 'could have prevented' WannaCry ransomware attack*. [Online; posted 27-October-2017]. Oct. 2017. URL: https://www.bbc.com/news/technology-41753022.

[69] O'Sullivan, D. *When seeing is no longer believing*. Ed. by Network, C. N. [Online; posted January-2019]. Jan. 2019. URL: https://edition.cnn.com/interactive/2019/01/business/pentagons-race-against-deepfakes/.

[70] Pan, S. J. and Yang, Q. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[71] Papernot, N., Goodfellow, I., Sheatsley, R., Feinman, R., and McDaniel, P. "cleverhans v1.0.0: an adversarial machine learning library". In: *arXiv preprint arXiv:1610.00768* (2016).

[72] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. "Practical black-box attacks against machine learning". In: *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. ACM. 2017, pp. 506–519.

[73] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[74] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. "Distillation as a defense to adversarial perturbations against deep neural networks". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 582–597.

[75] Qian, Y., Dong, J., Wang, W., and Tan, T. "Deep learning for steganalysis via convolutional neural networks". In: *Media Watermarking, Security, and Forensics 2015*. Vol. 9409. International Society for Optics and Photonics. 2015, 94090J.

[76] Reeves, A. and Biancardi, A. "The Lung Image Database Consortium (LIDC) Nodule Size Report, Release: 2011-10-27". In: *URL http://www. via. cornell. edu/lidc* (2011).

[77] Rigaki, M. and Garcia, S. "Bringing a GAN to a knife-fight: adapting malware communication to avoid detection". In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2018, pp. 70–75.

[78] Rigaki, M. and Garcia, S. "Bringing a gan to a knife-fight: Adapting malware communication to avoid detection". In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2018, pp. 70–75.

[79] Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E., and Ahmadi, M. "Microsoft malware classification challenge". In: *arXiv preprint arXiv:1802.10135* (2018).

[80] Ronneberger, O., Fischer, P., and Brox, T. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[81] Rouhani, B. D., Chen, H., and Koushanfar, F. "Deepsigns: A generic watermarking framework for ip protection of deep learning models". In: *arXiv preprint arXiv:1804.00750* (2018).

[82] Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. "Backpropagation: The basic theory". In: *Backpropagation: Theory, architectures and applications* (1995), pp. 1–34.

[83] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. "Improved techniques for training gans". In: *Advances in neural information processing systems*. 2016, pp. 2234–2242.

[84] Samuel, A. L. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3 (1959), pp. 210–229.

[85] Schmidhuber, J. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (2015), pp. 85–117.

[86] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. "Membership inference attacks against machine learning models". In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 3–18.

[87] Su, J., Vargas, D. V., and Sakurai, K. "One pixel attack for fooling deep neural networks". In: *IEEE Transactions on Evolutionary Computation* (2019).

[88] Suwajanakorn, S., Seitz, S. M., and Kemelmacher-Shlizerman, I. "Synthesizing obama: learning lip sync from audio". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 95.

[89] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[90] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[91] Thaler, S. and Menkovski, V. "The role of deep learning in improving healthcare". In: *Data Science for Healthcare*. Springer, 2019, pp. 75–116.

[92] Thaler, S., Menkovski, V., and Petkovic, M. "Deep Learning in Information Security". In: *arXiv preprint arXiv:1809.04332* (2018).

[93] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. "Ensemble adversarial training: Attacks and defenses". In: *arXiv preprint arXiv:1705.07204* (2017).

[94] Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. "The space of transferable adversarial examples". In: *arXiv preprint arXiv:1704.03453* (2017).

[95] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. "Stealing machine learning models via prediction apis". In: *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 2016, pp. 601–618.

[96] Vincent, J. *All of these faces are fake celebrities spawned by AI*. Ed. by Verge, T. [Online; posted 30-October-2017]. Oct. 2017. URL: https://www.theverge.com/2017/10/30/16569402/ai-generate-fake-faces-celebs-nvidia-gan/.

[97] Xuan, X., Peng, B., Dong, J., and Wang, W. "On the generalization of GAN image forensics". In: *arXiv preprint arXiv:1902.11153* (2019).

[98] Yang, X., Li, Y., and Lyu, S. "Exposing deep fakes using inconsistent head poses". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 8261–8265.

[99] Yin, C., Zhu, Y., Fei, J., and He, X. "A deep learning approach for intrusion detection using recurrent neural networks". In: *Ieee Access* 5 (2017), pp. 21954–21961.

[100] Yu, N., Davis, L., and Fritz, M. "Attributing Fake Images to GANs: Analyzing Fingerprints in Generated Images". In: *arXiv preprint arXiv:1811.08180* (2018).

[101] Yuan, X., He, P., Zhu, Q., and Li, X. "Adversarial examples: Attacks and defenses for deep learning". In: *IEEE transactions on neural networks and learning systems* (2019).

[102] Yuan, Z., Lu, Y., Wang, Z., and Xue, Y. "Droid-sec: deep learning in android malware detection". In: *ACM SIGCOMM Computer Communication Review*. Vol. 44. 4. ACM. 2014, pp. 371–372.

[103] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M. P., Huang, H., and Molloy, I. "Protecting intellectual property of deep neural networks with watermarking". In: *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM. 2018, pp. 159–172.

[104] Zheng, L., Zhang, Y., and Thing, V. L. "A survey on image tampering and its detection in real-world photos". In: *Journal of Visual Communication and Image Representation* 58 (2019), pp. 380–399.

[105] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

Appendix A

LIDC-IDRI Dataset

Manufacturer	Model	# Volumes	Volume avg.	# Slices
Philips		**56**	**299**	**16 739**
	Brilliance 16	4	255	1 020
	Brilliance 16P	42	282	11 849
	Brilliance 40	6	403	2 420
	Brilliance 64	4	363	1 450
GE Medical Systems		**608**	**233**	**141 843**
	Lightspeed plus	54	135	7 265
	Lightspeed power	10	382	3 819
	Lightspeed Pro 16	69	429	29 586
	Lightspeed QX/i	88	132	11 578
	Lightspeed Ultra	135	266	35 928
	Lightspeed VCT	61	203	12 379
	Lightspeed 16	191	216	41 288
Siemens		**180**	**269**	**48 352**
	Definition	3	276	827
	Emotion 6	24	135	3 239
	Emotion Duo	1	278	278
	Sensation 16	104	288	29 917
	Sensation 64	48	294	14 091
Toshiba		**58**	**117**	**6 786**
	Aquilon	58	117	6 786

Table A.1: The distribution of scanner devices present in the LIDC-IDRI dataset.

Appendix B

Additional Experiment Results

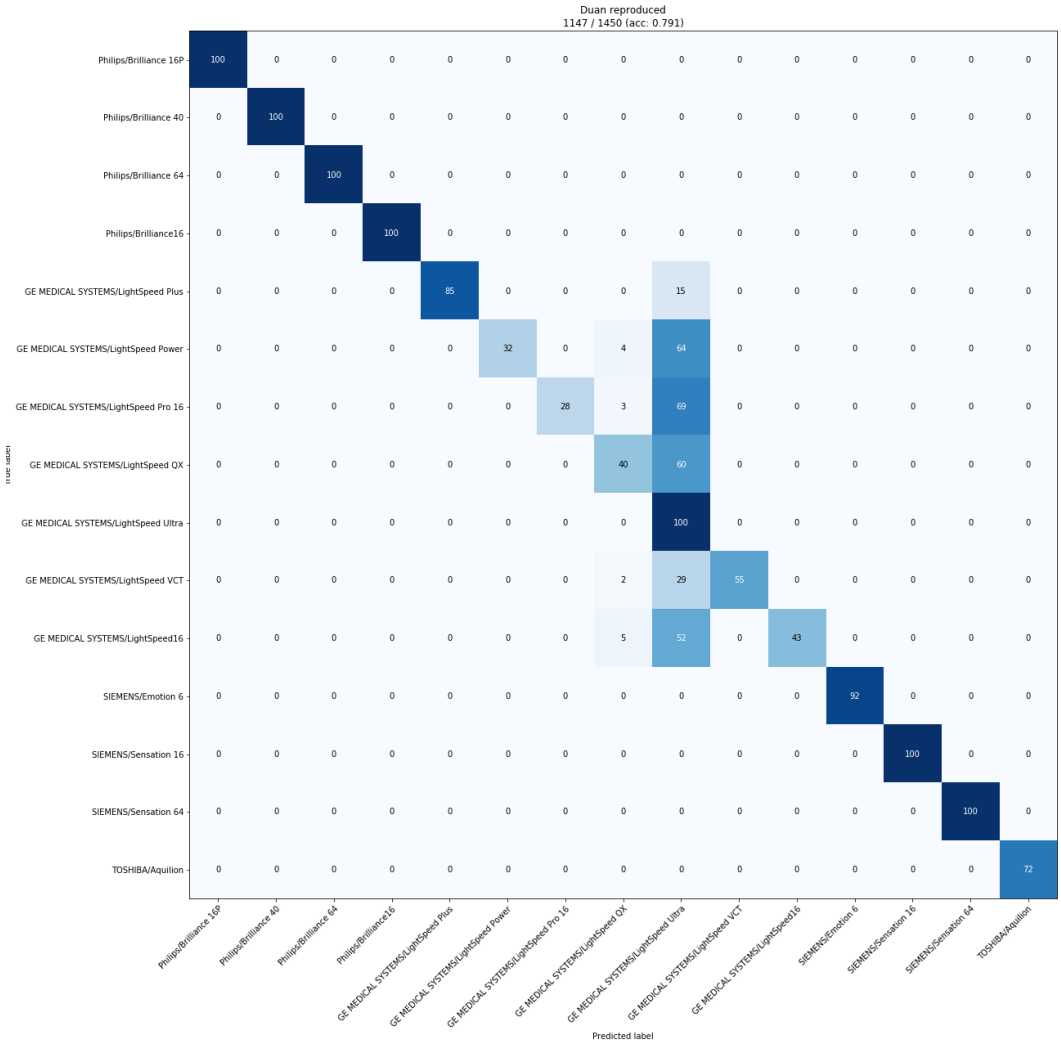


Figure B.1: Result from reproducing Duan et al. [22] with a different preprocessing step.

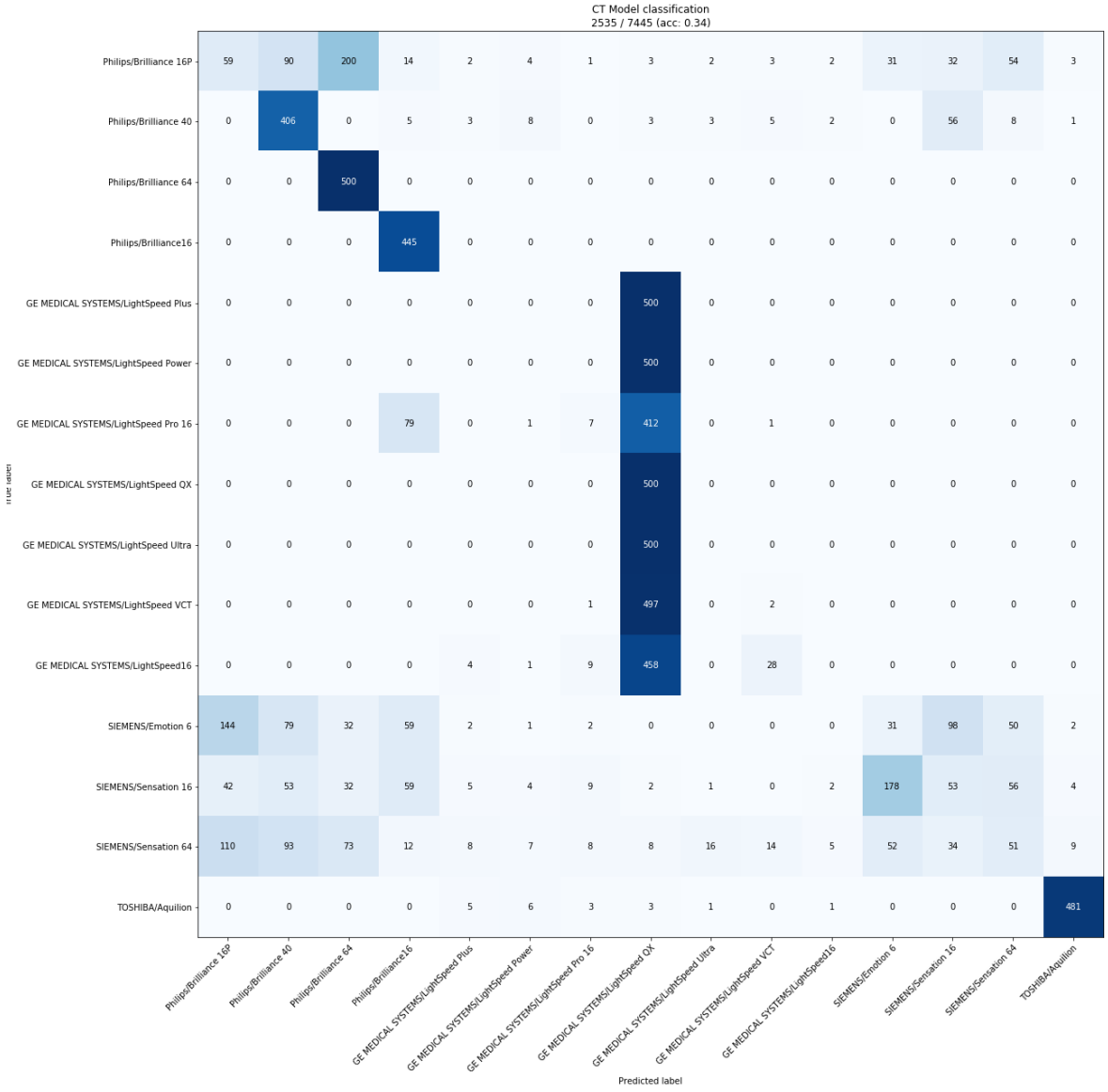


Figure B.2: Result from extending the experiment of Duan et al. [22] by using a larger dataset and applying additional restrictions.

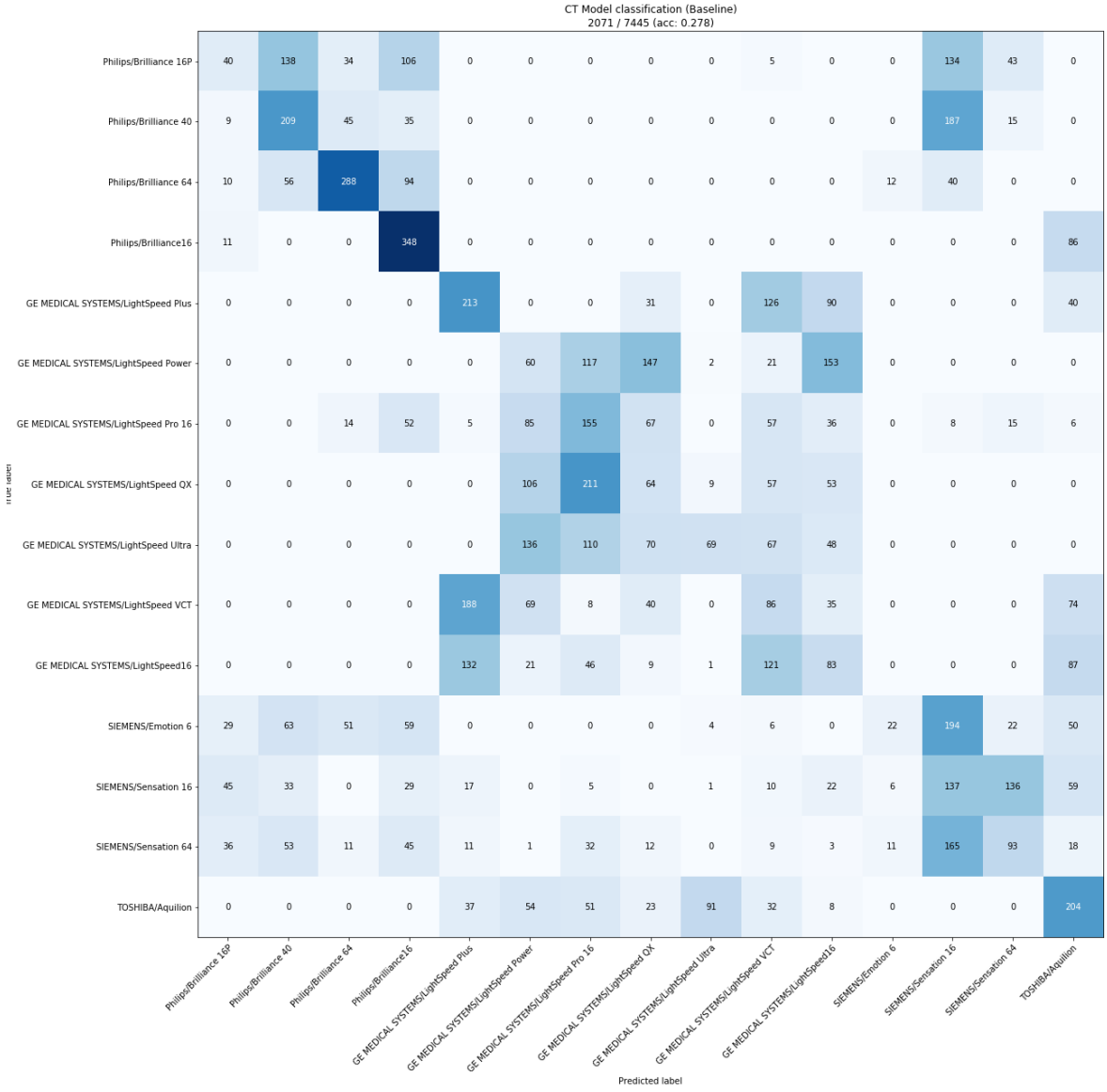


Figure B.3: Scanner model classification on original images based on the extended experimental setup.

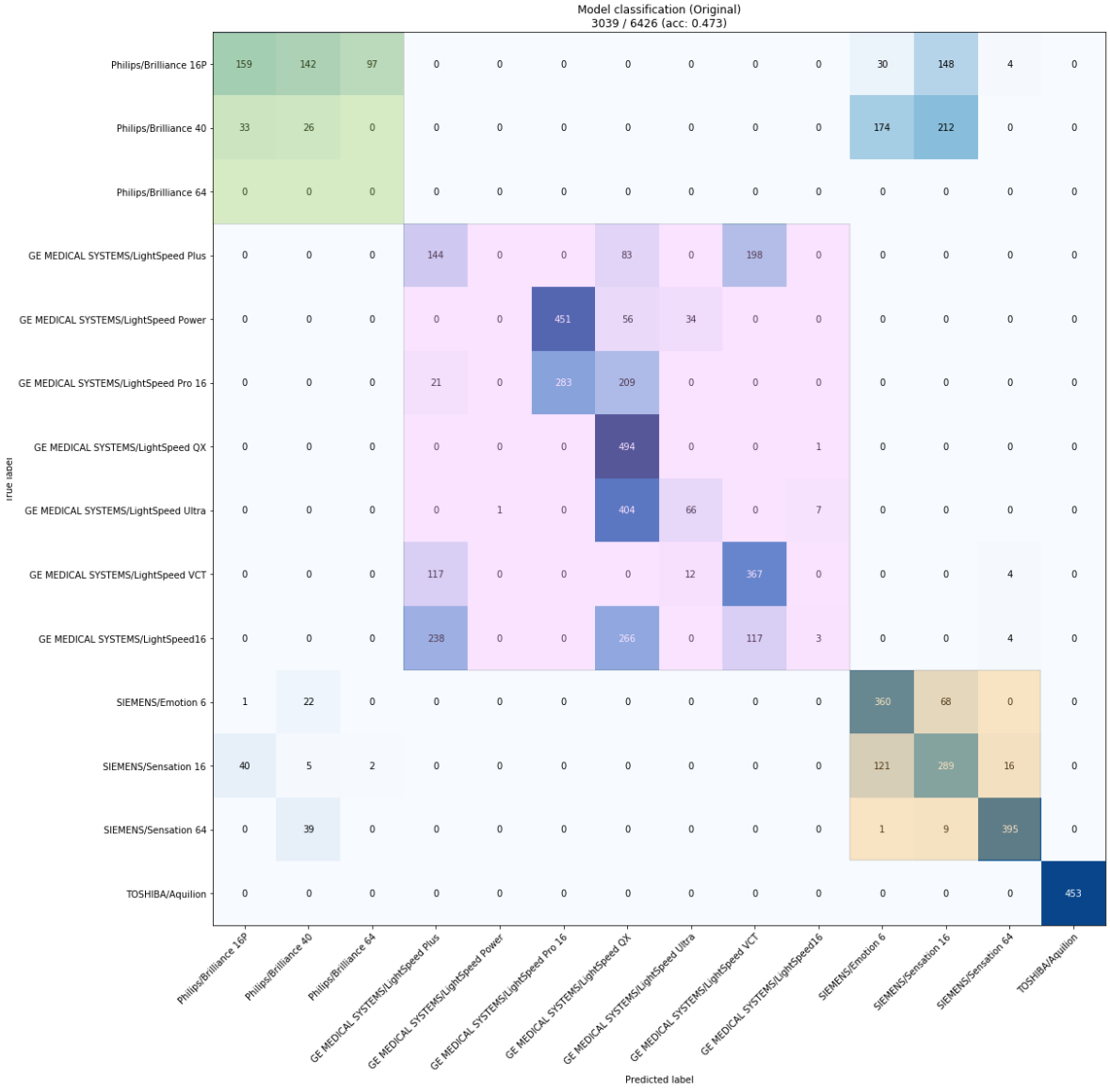


Figure B.4: Scanner model classification on original images using a neural network. The overlayed rectangles indicate models which share the same manufacturer.

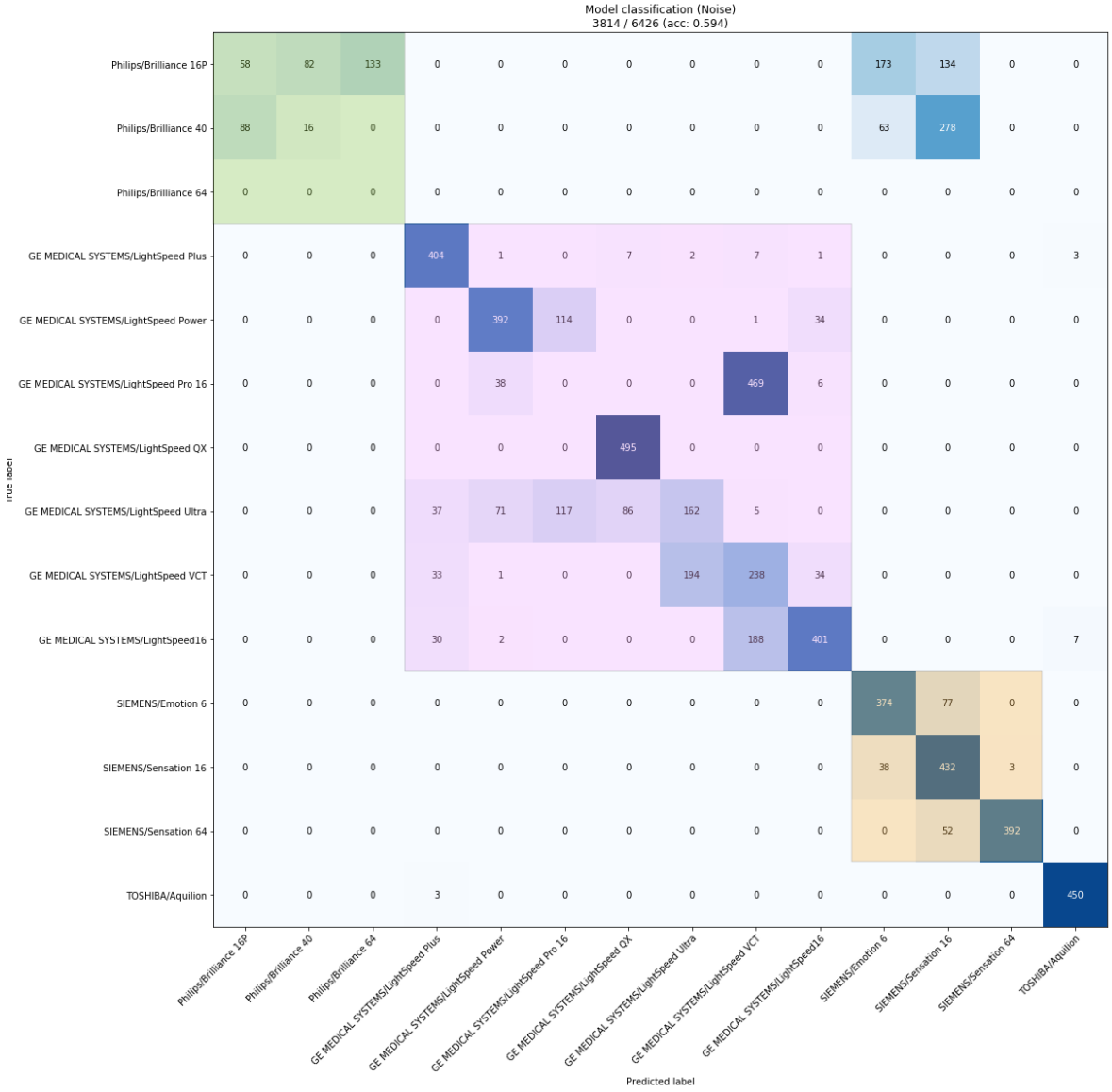


Figure B.5: Scanner model classification on noise patterns using a neural network. The overlayed rectangles indicate models which share the same manufacturer.

Appendix C

Training Graphs of Nodule Detector

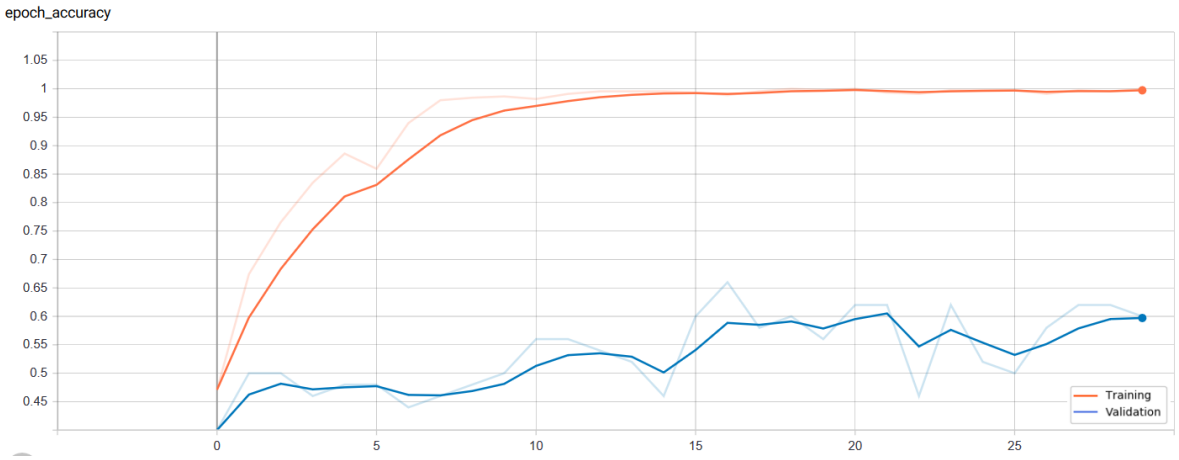


Figure C.1: Accuracy during the training phase of the nodule detector.

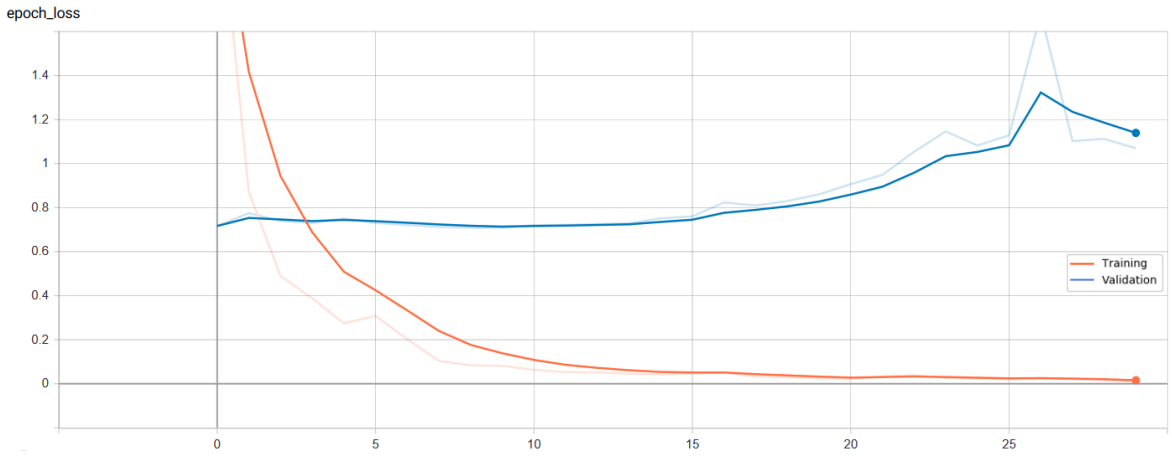


Figure C.2: Loss during the training phase of the nodule detector.