Friedrich-Alexander-Universität

Pattern Recognition Lab

Analysis of Different Optimization Strategies for an Adversarial Chest X-ray Anonymization Approach

Master's Thesis in Data Science

submitted
by

Raja Atreja

born 25.12.1996 in Hathras, India

Written at

Lehrstuhl für Mustererkennung (Informatik 5)
Department Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg.

Advisor: Kai Packhäuser, M. Sc.,
Mathias Öttl, M. Sc.,
Prof. Dr.-Ing. habil. Andreas Maier

Started: 01.05.2024

Finished: 01.11.2024

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 01.11.2024 Raja Atreja

Acknowledgements

First and foremost, I thank God for the blessings and strength provided throughout this journey.

I am deeply thankful to Prof.Dr. Andreas Maier for introducing me to this thesis opportunity. Without his guidance in the field of Deep Learning, this work would not have been possible.

I am incredibly thankful to Kai Packhäuser for his constant support and guidance throughout my thesis journey. His continuous sharing of knowledge and experience and his progress reviews have been instrumental to my growth.

I would like to express my warmest gratitude to Mathias Öttl for his unwavering cooperation and valuable insights. I thank him for sharing his immense knowledge and promptly addressing my queries.

Kai and Mathias always built a progressive environment where I could safely discuss my ideas and feel motivated throughout my journey.

A special thanks would be dedicated to my roommate Priyanka Singh for helping build a calm and supportive environment where I could calmly think and openly discuss my ideas to get an honest analysis. A warm thanks to all my friends for their support and well wishes, especially Ekaansh Khosla, who sometimes helped me with some deep-learning concepts.

Lastly, I would like to express my deepest thanks to my parents, who motivated me throughout the entire journey of my master's course. Their prayers and wishes for me are the key factors of my success. Thank you for your love and support throughout my life.

Raja Atreja

Übersicht

Moderne Re-Identifizierungsnetzwerke stellen eine große Bedrohung für die Privatsphäre des Patienten dar, wenn medizinische Bilder für Forschungszwecke freigegeben werden. Diese Arbeit untersucht fortschrittliche Optimierungsstrategien zur Verbesserung der Privatsphäre und des Nutzens von anonymisierten Röntgenbildern mit Hilfe von Deep-Learning-Techniken. Frühere Studien wie PriCheXy-Net erreichten einen hohen Nutzen und Datenschutz durch die Verwendung eines U-Netz-Generators und zweier Hilfsmodelle in einer Minimax-Architektur. Unsere Studie konzentriert sich auf die Verbesserung des PriCheXy-Net-Modells durch die Einbeziehung des Swin-Transformers als Hilfsklassifikator und die Einführung eines Hilfsdiskriminators, um den Realismus der generierten Bilder zu verstärken. Die Methodik umfasst umfassende Experimente mit dem ChestX-ray14-Datensatz, bei denen die Leistung der Modelle PriCheXy-Net, PriSwin-Net und PriSwin-Dis verglichen wird. Die Integration des Swin-Transformers führte zu einer signifikanten Verbesserung der Klassifizierungsgenauigkeit und erreichte einen AUC von 83,2%, verglichen mit 75,4% mit dem DenseNet-basierten Klassifikator. In ähnlicher Weise führte die Hinzufügung eines Hilfsdiskriminators in PriSwin-Dis zu einem verbesserten Bildrealismus und Datenschutz, wobei das Modell eine AUC von 82,9% und niedrigere Verifizierungswerte erreichte, was auf ein geringeres Risiko der Re-Identifizierung hinweist. Die Ergebnisse unterstreichen die Bedeutung fortschrittlicher Modellarchitekturen und Optimierungsstrategien, um ein optimales Verhältnis zwischen Privatsphäre und Nutzen zu erreichen. Diese Forschungsarbeit unterstreicht die entscheidende Rolle von Transformator-basierten Modellen und kontradiktorischen Ansätzen bei der Entwicklung robuster Anonymisierungsmethoden, die die Privatsphäre der Patienten schützen und gleichzeitig den diagnostischen Nutzen medizinischer Bilder erhalten. Durch die Weiterentwicklung dieser Methoden wollen wir zur Entwicklung sicherer und effektiver medizinischer Bildgebungssysteme beitragen und damit letztlich die Patientenversorgung und den Datenschutz verbessern. Diese Anonymisierungsmethoden könnten für Anwendungen eingesetzt werden, die die gemeinsame Nutzung medizinischer Daten beinhalten. Zum Beispiel, um große Datensätze für Forschungszwecke, Bildungsanwendungen für Medizinstudenten usw. öffentlich zugänglich zu machen.

Abstract

Modern re-identification networks pose a great threat to the patient's privacy while sharing medical images for research purposes. This thesis investigates advanced optimization strategies to enhance the privacy and utility of anonymized chest X-ray images using deep learning techniques. Earlier studies like PriCheXy-Net achieved high utility and privacy by using a U-Net generator and two auxiliary models in a minimax architecture. Our study focuses on improving the PriCheXy-Net model by incorporating the Swin Transformer as an auxiliary classifier and introducing an auxiliary discriminator to enforce realism in generated images. The methodology involves comprehensive experiments on the ChestX-ray14 dataset, comparing the performance of the PriCheXy-Net, PriSwin-Net, and PriSwin-Dis models. The integration of the Swin Transformer demonstrated significant improvements in classification accuracy, achieving an AUC of 83.2%, compared to 75.4% with DenseNet-based classifier. Similarly, the addition of an auxiliary discriminator in PriSwin-Dis resulted in enhanced image realism and privacy, with the model achieving an AUC of 82.9% and lower verification scores, indicating reduced risk of re-identification. The results underscore the importance of advanced model architectures and optimization strategies in achieving a superior privacy-utility trade-off. This research highlights the critical role of transformer-based models and adversarial approaches in developing robust anonymization methods that protect patient privacy while maintaining the diagnostic utility of medical images. By advancing these methods, we aim to contribute to the development of secure and effective medical imaging systems, ultimately improving patient care and data privacy. These anonymization methods could potentially be used for applications that involve medical data sharing. For instance, to make large datasets publicly available for research purposes, educational applications for medical students, etc.

Contents

Chapter 1

Introduction

Medical Imaging Techniques (MITs) are non-invasive methods that help to look inside a body without opening it through surgery. Medical practitioners use these techniques to diagnose different diseases and problems. Thus, MITs have been a crucial part of medical examinations. Techniques include X-ray radiography, X-ray Computed Tomography (CT), Magnetic Resonance Imaging (MRI), ultrasonography, optical imaging, Positron Emission Tomography (PET), and Single Photon Emission Computed Tomography (SPECT), Fluoroscopy, Nuclear Medicine [Kas+15].

Among these techniques, **X-ray radiography** is the most widely accessible and used. According to WHO 2016, 3.6 billion medical diagnoses such as X-rays were performed [Wor16]. During COVID-19, due to the lower number of CT machines for medical imaging, Chest X-ray Radiography (CXR) was used dominantly [Jac+20] worldwide. Thus, the number of X-rays performed increased drastically. Figure 1.1 shows that between December 2022 and December 2023, X-rays were the most commonly performed imaging procedure on National Health Service (NHS) patients in England [NHS24].

The increasing number of chest radiographs requires deep-learning-based diagnostic systems, including disease identification and classification. Examples of such systems are automatically recognizing abnormalities in chest radiographs [Gue+19b; Gue+19a] and detecting tumors in mammography [Aks+16]. Thus, large datasets of these digital medical records must be created and released publicly to train these deep-learning-based systems. However, developing such systems for detecting and classifying CXR images has its share of difficulties, the most important being **patients' privacy**.

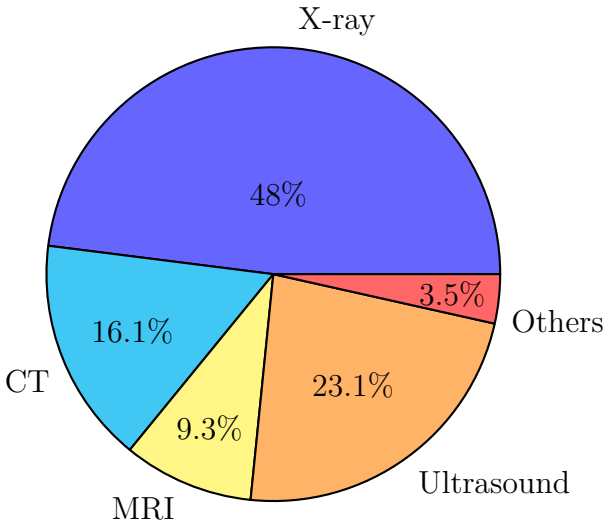


Figure 1.1: Count of imaging activity in England, on NHS patients, December 2022 to December 2023[1].

1.1 Importance of Chest X-ray Anonymization

Medical images, like chest radiographs, contain sensitive personal details in various forms, like the patient's **metadata** associated with the X-ray or any visual information specific to a patient [Wil+20]. These pieces of information about the patients are bound by privacy protection regulations, such as the European Union (EU)'s **General Data Protection Regulation (GDPR)** or United States of America (USA)'s Health Insurance Portability and Accountability Act (HIPAA) [The13]. Hence, it is important to remove all personal information associated with medical data before publicly releasing it for research purposes to prevent privacy breaches.

Medical CXR images also inherently contain certain **biometric information**, similar to fingerprints, which can be leveraged by deep-learning-based re-identification systems. This makes the data vulnerable to linkage attacks, posing a potential risk to patient privacy [Pac+22]. Therefore, removing all the meta-data associated with these images is not enough, as the image in itself contains private information about any patient, like the width of the chest, the size of the heart, and electronic devices like pacemakers. A malicious user or interested parties like health insurance companies can use this information to train a deep learning model to re-identify the patient and perform a **linkage attack**. To perform such a linkage attack, one can design a simple re-identification network as shown

[1]Summarised data from diagnostic imaging statistical dataset released by [NHS24].

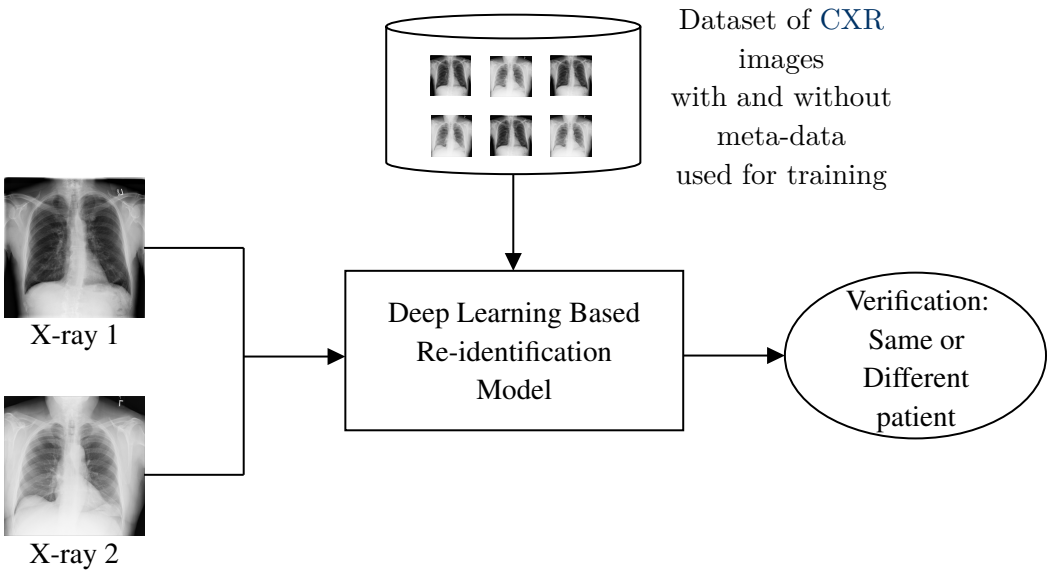


Figure 1.2: Architecture for re-identification of X-ray images with deep learning-based verification model.

in Figure 1.2, which simply requires a Deep Learning (DL) based re-identification network and anonymized image dataset for training. An example of such a deep-learning-based re-identification architecture is **Siamese Neural Network (SNN)**, which takes two chest X-ray images and classifies whether they belong to the same patient or not [Pac+22].

Therefore, it becomes important to properly anonymize medical images before releasing them to the public in order to reduce the possibility of linkage attacks, enhancing **privacy** but, on the other hand, keeping their **utility** as high as possible.

1.2 Challenges in Maintaining Privacy and Data Utility

Privacy of a medical image can be ensured by various methods like removing all the meta-data or obscuring certain parts of an image with **black boxes** [Pac+22]. However, these methods do not offer complete security from linkage attacks. These methods may also reduce the utility of the image as black boxes might hide crucial information required for diagnoses.

To overcome privacy issues, **perturbation-based anonymization** methods were also used on medical images [Kai+20] e.g., differential privacy [Dwo+06; Dwo08], later extended to image data with the DP-Pix method [Fan18; Fan19], which combines pixelization and

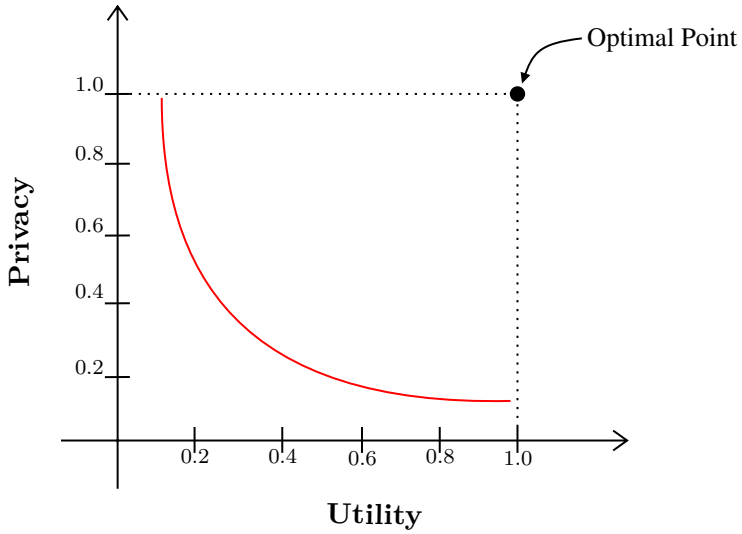


Figure 1.3: Privacy-Utility trade-off.

adding random noise based on a Laplacian mechanism [Dwo+06]. This approach addresses the privacy of an image but inherently loses its utility because of pixelization and blurring.

Hence, when a deep learning model attempts to anonymize an image, it changes the image or certain part of the image to secure the patient's privacy. However, due to the image transformations, visual information about the disease might be lost, reducing the image's utility. Therefore, as the privacy of an anonymized medical image increases, its utility decreases and vice-versa. This creates a **privacy-utility trade-off** [Zho+22] as shown in Figure 1.3. In an ideal case, the privacy and utility of an anonymized image should be 1.0 (optimal point as shown in Figure 1.3), which means that an image is modified to have no private information and at the same time retains the disease information perfectly.

In this aspect, **adversarial image anonymization** approaches aim to guarantee both data utility and image privacy. For instance, PriCheXy-Net [Pac+23b] - a model architecture comprised of three independent neural networks. It tries to balance the privacy and utility of the image generated by ensuring the non-transference of biometric identifiers and has shown promising results in this regard on chest X-ray images [Pac+23b].

1.3 Thesis Objectives

Many deep-learning models have recently been released for medical image diagnoses, such as tuberculosis detection from chest X-ray [Rah+20] or automatic COVID-19 detection mechanisms [Rah+22]. Publicly released datasets are used to train these models. This

poses a limitation that, often, these datasets are not properly treated for removing patients' private information [Pac+22]. Patients' privacy also hinders hospitals from releasing medical imaging data for public research [Kai+20].

This thesis tries to address the **optimization of privacy and utility** of anonymized CXR images and improve the trade-off between both so that more datasets can be released publicly without risking patient privacy.

We will explore the areas of improvement in the performance of **PriCheXy-Net** [Pac+23b] by modifying the existing architecture. Our area of focus in medical imaging will remain on Chest X-ray Radiography (CXR) images from the **ChestX-ray14** [Wan+17] dataset. This work's main contributions include:

- Current architecture of PriCheXy-Net [Pac+23b] includes a Dense-Net [Hua+17] based auxiliary classifier. We will replace this Convolutional Neural Network (CNN) [Sim+14]-based classifier with models having more capacity like **Swin transformer** [Liu+21].
- Applying an additional **Discriminator loss** function to enforce realism in the images generated by the architecture and analyze its effect on other auxiliary models present in the architecture.
- Concluding the effect of each auxiliary model in the architecture and optimizing them by performing hyper-parameter tuning (e.g., properly weighting the loss functions).
- Investigate the lower dimensional space before and after 'anonymization' to analyze the effect of the applied anonymization architecture. And perform an in-depth analysis like quantifying the intra-class difference between images from before and after anonymization.

These objectives collectively aim to improve the performance of PriCheXy-Net [Pac+23b] and build a robust and reliable architecture that can anonymize Chest X-ray images to prevent linkage attacks and keep their utility high at the same time. This could potentially help organizations to publish medical image data for research purposes without compromising patients' privacy.

Chapter 2

Background

2.1 Medical Imaging

Medical Imaging was first performed by Wilhelm Conrad **Röntgen** in Würzburg, Germany in 1895. He was the first person to capture an x-ray image with a simple cathode ray tube called a Crookes tube [Sea+79], when he noticed that certain invisible rays could pass through human skin more easily than through bones or metal. Later, he was awarded a Nobel Prize for the discovery of X-rays [Bra08].

Since then, medical imaging has experienced significant growth, playing vital roles in biomedical engineering and clinical practice. Its contribution has been instrumental in improving our understanding of the disease process and in making new discoveries in the pathophysiology of diseases, as well as in developing new diagnostic methods [Ana+12].

Various techniques can be employed to **visualize** the interior of the human body. These distinct methods are founded on the transmission of signals through the body, which interact with the body's tissues. Auditing the signals emitted from the body can produce an image representing the patient's internal structures [Kas+15]. The basic concept of medical imaging systems is shown in Figure 2.1. Some notable techniques include X-ray radiography, X-ray CT, MRI, ultrasonography, optical imaging, PET, and SPECT, Fluoroscopy, and Nuclear Medicine.

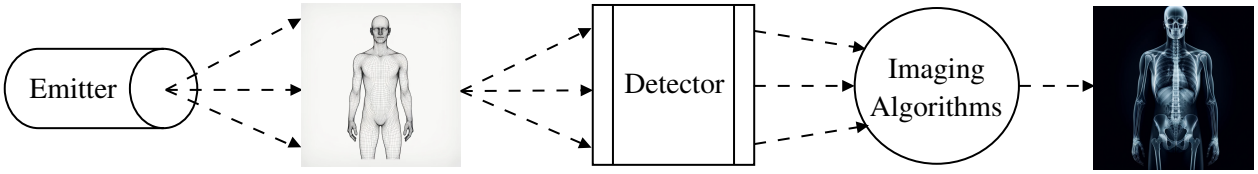


Figure 2.1: Concept of a medical imaging system[1]. A general medical imaging system has a source that emits certain signals or rays capable of passing through a human body. The interaction of these emitted rays with the human body is then recorded by a detector which converts this information into electrical signals. At last, these signals are processed by imaging algorithms to produce an image that helps in visualizing the internal structure of the human body.

2.1.1 X-rays Radiography

The electromagnetic spectrum in Figure 2.2 shows that the wide spectrum of light is divided into multiple ranges, varying from very long radio waves (used in MRI), extending over microwaves, infrared, visible light, and ultraviolet light to x-rays (used in radiography) [Sue17].

X-rays are - similar to visible light - **electromagnetic radiation** waves discovered by Röntgen in his experiments with a fluorescent screen, on which he observed the light emitted from an X-ray tube. He saw that this was emitting a novel type of radiation in addition to light. Hence, because of the mysterious nature of these rays, he named them X-rays. And only a few months later, radiographs were adapted to clinical usage [Sue17].

These electromagnetic radiations from X-rays consist of photons. The energy E of these photons can be represented by their frequency f and wavelength λ, i.e.,

$$E = hf = \frac{hc}{\lambda} \tag{2.1}$$

where c is the speed of light in a vacuum, and h is Planck's constant.

Description of an X-ray Machine

X-ray radiography imaging machines require five major components and a patient to perform a medical image radiograph, as shown in Figure 2.3, which are as follows:

- An **X-ray source** - a light-emitting X-ray ray tube.

[1]Human and X-ray images have been generated by ChatGPT v4o.

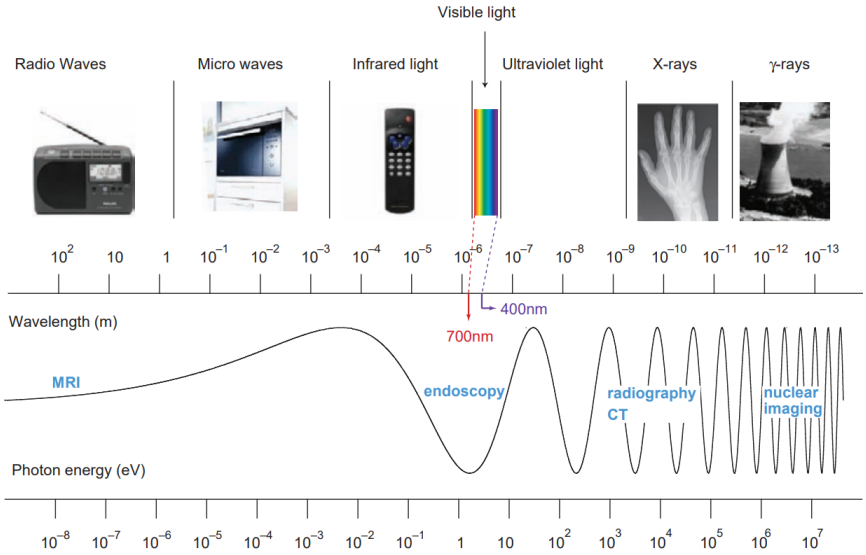


Figure 2.2: The electromagnetic spectrum [Sue17].

- Low-energy photons are eliminated by an **aluminum filter** because they lack the energy to pass through the patient and never reach the detector, making them unusable for imaging.
- A **collimator** is a device to limit the patient area in which an X-ray is performed.
- A patient whose X-ray is to be performed.
- A collimating scatter grid that absorbs the scattered photons.
- A **detector** - a screen-film combination with a film positioned between an image intensifier and a camera. Nowadays, flat-panel detectors are used with more recent X-ray machines.

X-ray Procedure

To perform an X-ray, radiating X-ray light is emitted from an X-ray tube (source), which then passes through an aluminum filter. By doing this, photons with low energy are filtered away, and the photon ray's mean energy is increased. A collimator device is then used to limit the number of photons that must be passed through. This allows the targeted X-ray scanning of the patient, and only a particular area of the patient is exposed to the rays. When these high-energy photons collide with the patient's bones, they produce a scatter. These scattered photons are then captured by a collimating scatter grid placed behind

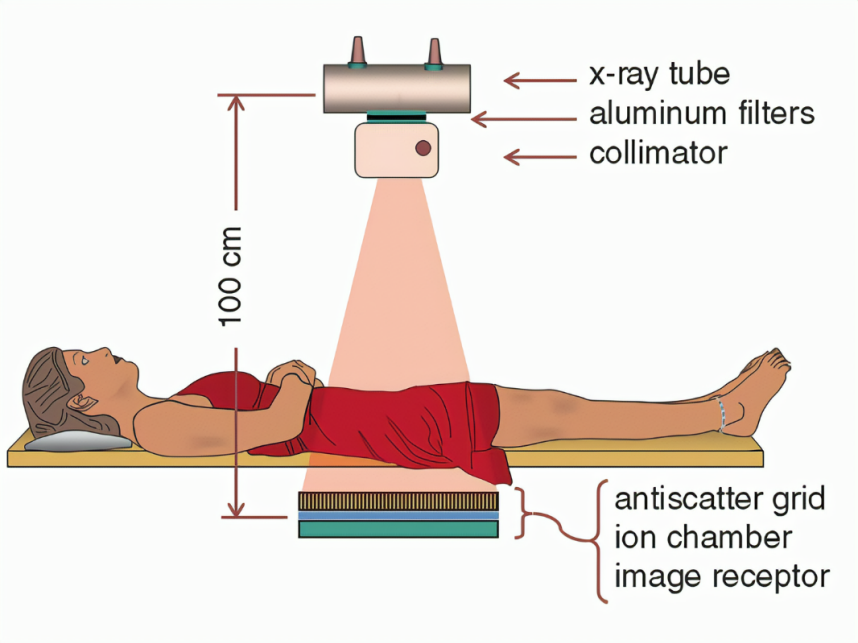


Figure 2.3: Schematic representation of X-ray radiography imaging procedure [2].

the patient. Finally, dispersed photon positions are recorded by a detector that combines a camera and an image intensifier to produce an image, as shown in Figure 2.6. In the following points we will discuss the X-ray procedure in detail:

- **X-ray Generation** [Mai+18]: The X-ray generation starts with a specially designed device known as an X-ray tube, typically consisting of a vacuum-sealed glass chamber housing a cathode and a metal anode. When the cathode's filament is heated, it emits electrons via thermionic emission, due to the thermal energy overcoming the material's binding energy. These electrons are then accelerated toward the anode by an applied voltage. Upon striking the anode, the electrons undergo rapid deceleration, leading to the emission of X-rays. This emission occurs through two primary mechanisms: characteristic radiation and Bremsstrahlung. In the former, an electron from the cathode displaces an inner-shell electron in the target atom, emitting an X-ray photon with energy corresponding to the difference between the involved electron shells. The latter process, Bremsstrahlung, occurs when an electron is deflected by the nucleus of an anode atom, losing kinetic energy which is subsequently released as a continuous spectrum of X-ray photons [Mai+18].

[2]Geomerty of Projection Radiography (https://radiologykey.com/radiography-3).

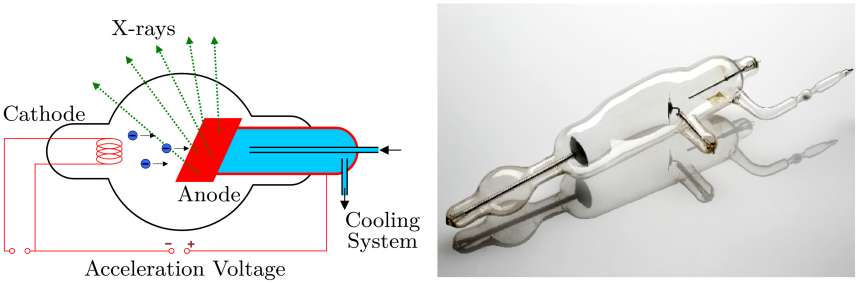


Figure 2.4: Vacuum X-ray tube. The image on the left illustrates the schematic representation of how electrons travel from the cathode to the anode, generating X-ray photons & the right image is taken from the Science Museum, London which shows an early version of a vacuum X-ray tube [Mai+18].

- **X-ray Matter Interaction** [Mai+18]: X-rays are capable of penetrating different types of matter, including human tissue, making them invaluable for medical imaging. The extent of this penetration, however, varies depending on the material's properties and the energy of the X-ray photons. When X-rays pass through matter, they interact in several ways: they can be absorbed, elastically scattered, or inelastically scattered. In medical imaging, these interactions lead to a reduction in the intensity of the X-ray beam, a process known as attenuation. Attenuation is influenced by various factors such as changes in photon count, direction, and energy, all of which are energy-dependent [Mai+18].

  The degree of attenuation as X-rays pass through a material can be described mathematically by **Lambert-Beer's law** (Equation (2.2)). According to this law, when a monochromatic X-ray beam passes through a homogeneous material with an absorption coefficient μ the intensity of the X-ray beam decreases exponentially with the thickness x of the material. The relationship is expressed as:

$$I = I_0.e^{-\mu x} \tag{2.2}$$

  where I_0 is the initial intensity of the X-rays, and I is the intensity after passing through the material. This principle is fundamental in X-ray CT, where the fractional transmitted intensity is used to reconstruct images of internal structures. In real-world applications, however, X-ray beams are often poly-energetic, leading to a more complex interaction that must account for the varying energy spectrum of the X-rays and the energy-dependent attenuation coefficient [Mai+18].

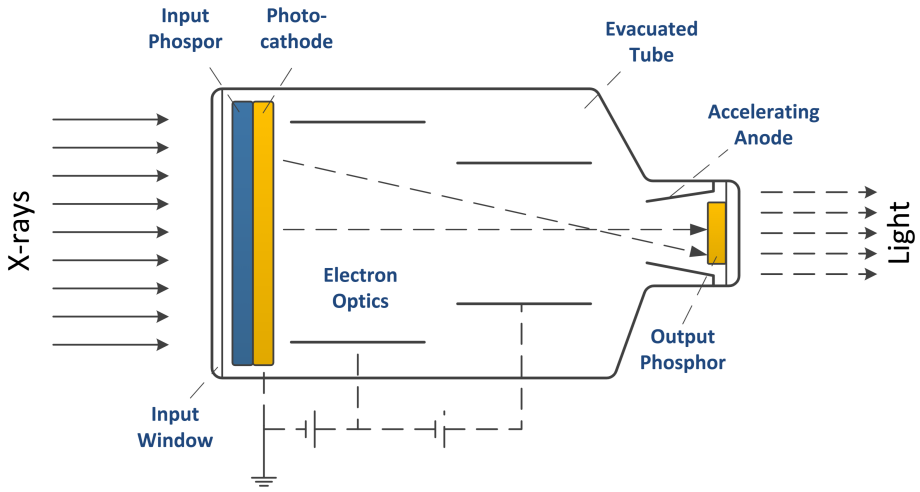
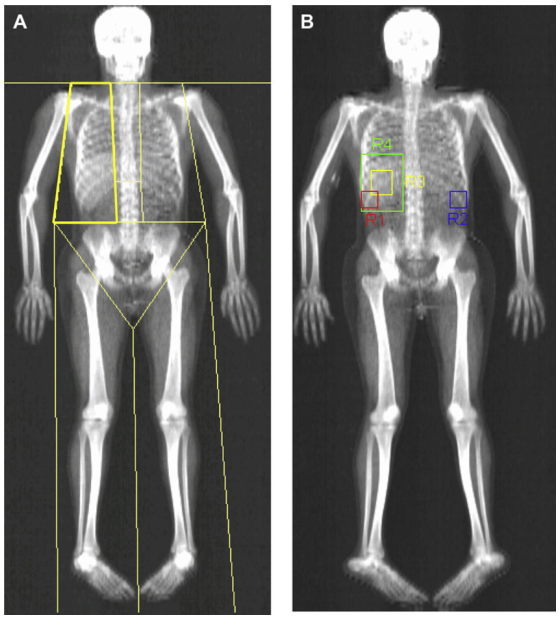


Figure 2.5: Illustration of an image intensifier detector. X-rays are first transmuted into light. This light is subsequently converted into electrons, which are directed by an optical system toward a fluorescent screen or film. The screen or film then reverts these electrons into the light once again, resulting in the creation of the final image [Mai+18].

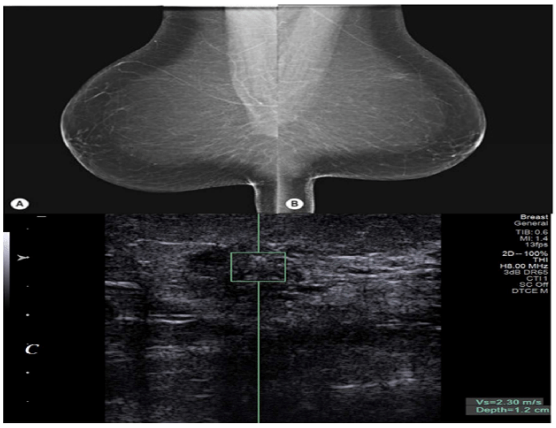
- **X-ray Image Acquisition** [Mai+18]: X-ray imaging has evolved significantly, transitioning from traditional X-ray films to advanced detection systems that enhance image quality and reduce radiation exposure. Modern X-ray imaging systems convert X-rays into visible light and then into electronic signals to create images. An example representation of one such image intensifier is shown in Figure 2.5. This process is facilitated by devices such as X-ray image intensifiers, which are vacuum tubes that transform X-rays into visible images through several key steps. Initially, X-rays pass through an input phosphor layer, which converts the X-ray photons into light photons. These light photons then trigger the photoelectric effect within a photocathode, resulting in the emission of electrons [Mai+18]. These electrons are then accelerated and focused onto an output phosphor by an electron optic system. When these electrons strike the output phosphor, they are converted back into visible light, which can be captured by imaging devices like film or television camera tubes and generate the image [Mai+18].

Clinical Usage and Safety Concerns of X-rays

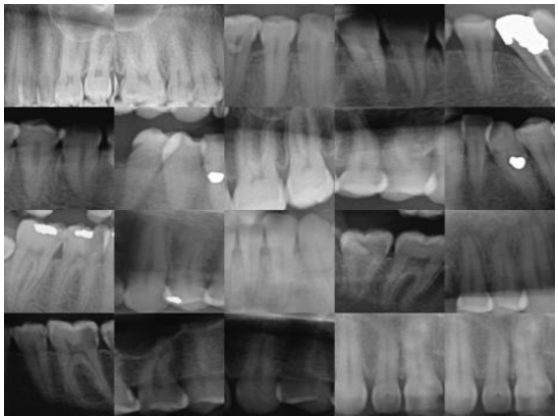
There are various commonly used radiography examinations for all parts of the human body, which constitute the majority of radiological tests. The most typical inquiries involve the following:



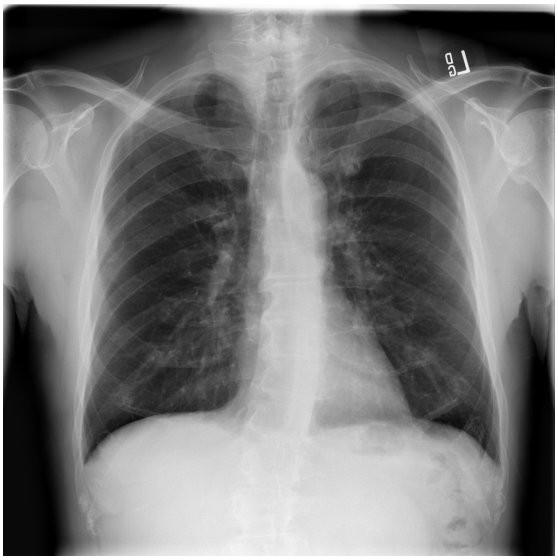
(a) Skeletal X-ray [She+10]



(b) Mammography [Rat+18]



(c) Dental X-ray [Lia+23]



(d) Chest X-ray [Wan+17]

Figure 2.6: Commomnly performed medical X-ray radiography image types.

- Skeletal X-rays (see Figure 2.6a)
- Mammography (images of the breasts) (see Figure 2.6b)
- Dental X-rays (images of the teeth and jaw) (see Figure 2.6c)
- Chest images (radiographs of the thoracic cavity and heart) (see Figure 2.6d)

Though there have been concerns around the **biological effects and safety** of medical radiographs due to the exposure of high energy radiations [Gel+05], there has been no evidence of threshold of energy below which the probability of damage becomes zero [Sue17].

2.1.2 Overview of Thoracic-based Diseases

In this thesis, our focus remains on the thoracic-based X-ray images. Thoracic diseases occur in the area around the chest. These diseases are some of the major diseases that occur as they affect the lungs, heart, and surrounding organs (Figure 2.7). Thoracic conditions range from angiocardiopathy to the common cold, asthma, and bronchiolitis to Chronic Obstructive Pulmonary Disease (COPD), tuberculosis, lung cancer, cystic fibrosis, and pulmonary hypertension [Fro23]. They are majorly categorized into:

- **Respiratory diseases** are diseases that occur due to breathing problems or a body's gaseous exchange functions. The areas affected by these types of diseases are the patient's lungs and air tract system. Due to respiratory diseases, the patient suffers from breathlessness and chest pain. Examples of such diseases include COPD, asthma, and pulmonary fibrosis [Wor23].
- **Cardiovascular diseases** mainly impact the heart and arteries of the patient. These types of diseases have conditions like irregular heart rhythms and blockage in arteries, which result in ineffective blood flow in the heart, leading to chest pain, fatigue, restlessness, and shortness of breath. They are the main cause of heart attack or stroke. Examples of these diseases are heart failure, pericarditis, and Coronary Artery Diseases (CoADs) [Gaz+06].
- **Other thoracic-based diseases** include diseases like pleural effusion [Lig02] and mediastinal tumors [Shr+06]. Heart failure, pneumonia, and malignancy are the primary causes of pleural effusion. This disorder, known as pleural effusion, is the result of the buildup of excessive fluid in the pleural space surrounding the

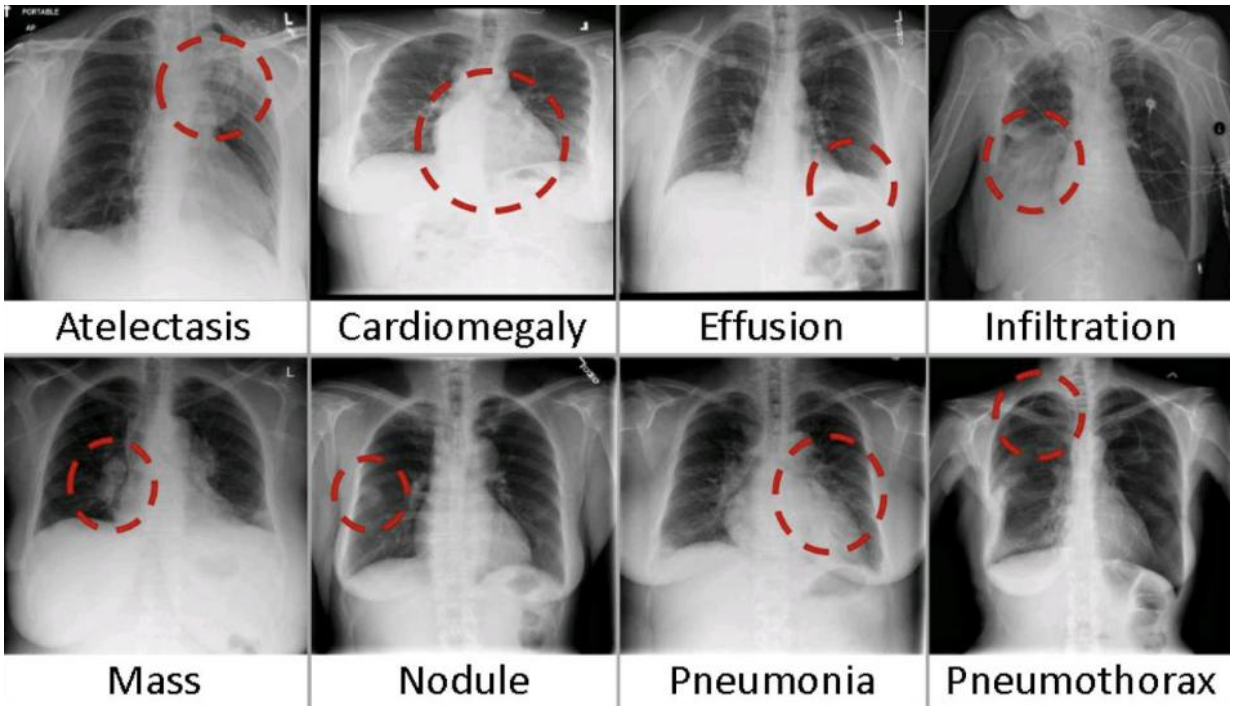


Figure 2.7: Some examples of thoracic-based diseases [Wan+17].

lungs [Kar+12]. Mediastinal masses are infrequent tumors that develop in the thoracic region of a patient, resulting in symptoms such as cough, chest pain, hoarseness, and weight loss [Shr+06].

2.1.3 Privacy Issues in Medical Images

The rise of medical advancement, especially in the area of diagnosis, resulted in a large number of medical imaging procedures being performed for diagnostic purposes. As the number of medical records grew, the **digitalization** of these records made their management easier. Digitalization also helped in the faster generation of reports for diagnosis. Their storage, maintenance, and retrieval became easier than ever because of fast computing systems and wireless networks [Li+05]. This contributed to delivering better healthcare for the patients.

Nevertheless, the ease of accessing and distributing data presents a potential threat to the privacy and confidentiality of patients and their personal information. To address the issue of privacy in medical images, various steps are being taken [Ach+03; Jin+12; Cao+03; Coa+00] to provide security in the following aspects for data access control:

- **Confidentiality** - Medical images of the patients should be accessed under proper security rules and only by authorized personnel. Any patient's personal information should be deleted if that is not crucial for the diagnosis of the disease.

- **Integrity** - Data should not be modified by anyone at any point of storage during the management or even while archiving the images.

- **Authentication** - Medical images should originate from a valid source, and only personnel with proper access rights should be able to access them.

Since the advancement in Artificial Intelligence (AI), Computer Aided Diagnostics (CAD) systems with the help of Computer Vision (CV) techniques and traditional Machine Learning (ML) [Aer+14; Lam+17] algorithms have shown promising results and revolutionized the field of medical imaging and diagnosis [Kai+20]. And because of these technological advancements, these security measures have become even more necessary. Malicious users can take advantage of DL based methods by utilizing the publicly available medical images to create malicious attacks on patients' privacy as described in Section 1.1 & Figure 1.2 [Pac+22].

Apart from these security measures, some **anonymization** (data anonymity) techniques are also implemented to protect patient privacy by removing certain information that reveals the patient's information directly or indirectly [Nat+21]. These privacy-inducing methods include techniques such as:

- **Substitution**, typically known as pseudonymization, is a technique where the key identifying factor, such as the name of the patient, is replaced by some other random value (replacing the whole word), keeping all the other important information intact [Nat+21].

- **Scrambling** involves mixing, introducing, or removing certain characters of the data before saving. This makes the data non-readable, and hence the patient's data becomes useless [Lee+17].

- **Masking** hides part of the data with random characters to protect private information. This creates data that is similar to the original data but makes the patient's private information unreadable because of the introduction of random characters in the middle of words, names, addresses, etc. [Nat+21].

- **Personalised anonymization** is a technique that allows the user to anonymize the data with their own anonymization technique. This can be performed using any script or an application. This randomized anonymization makes the data harder for privacy attacks [Nat+21].

- **Data Defocusing** is an approximation method. It is done by replacing original values with the approximated values of the original data. This makes it impossible to reidentify the patient to which the data belongs to [Ali+16; Lee+17]. Approximation in the data could be introduced by replacing exact ages with age ranges, adding a small random number to each numerical data point, etc.

- **Privacy protection with encryption** - Encryption is a widely used technique that protects sensitive information by encoding the data with the help of mathematical algorithms. The advantage of this type of technique is that the data can be decrypted with the help of a key and can be used to obtain the original data again. This protects the data from unauthorized access by third parties. Even if someone obtains the data, they will not be able to access it without a decryption key, as computations on encrypted data are not trivial [Lin+16].

These methods protect the patient's identity and other sensitive information. But, there are other privacy concerns which are further discussed in Chapter 3.

2.2 Deep Learning in Medical Imaging

ML techniques have been widely used in medical imaging and diagnosis. Many ML models, for example, have been created for the purpose of **disease detection and classification** in medical images [Wan+12; De 16]. ML models are being rapidly used in the medical imaging field for various applications like CAD and diagnosis, interpreting radiology reports, and medical image analysis [Suz17].

ML is a branch of AI that encompasses a collection of techniques designed to identify patterns in data by understanding its underlying statistical distribution. This understanding of underlying pattern enables any ML model to predict the future data or make certain decisions under the given conditions [Mur12].

Diseases and problems in medical imaging are too complex to be accurately approximated by a simple mathematical equation. For example, a lesion might occur in different shapes in different patients [Los+10]. These different shapes of lesions are very difficult to be

approximated by a single mathematical equation. To accomplish this task accurately, the ML model understands the given training data and learns the different shapes of lesions. When fully trained, a ML model can precisely detect the lesion for a given image by approximating from the learned examples. ML methods are categorized into three categories:

- **Supervised Learning** - attempts to learn the underlying distribution function of the data by observing both input and output. It functions as a student-teacher learning paradigm. The model learns from the input data and predicts the value for a test point. It then checks its predicted value against the real output for validation. If the prediction is wrong, the model gets punished and updates itself to accommodate the correction. Such techniques include linear regression, logistic regression, k-nearest-neighbor, decision trees, random forests, etc [Has+09].

- **Unsupervised Learning** is called unsupervised because there is no correct output provided, and thus, there is no teacher. Algorithms work on their own to find out interesting observations in the underlying data. Clustering and feature reduction techniques are the primary applications of unsupervised learning algorithms. It uses the previously learned features to find new features in the unseen data [Mah20]. K-means clustering, principal component analysis, and other techniques are examples of such techniques.

- **Reinforcement Learning** is a type of ML paradigm closely connected to decision theory and control theory. In it, the machine performs some actions to interact with the environment and change its state in order to earn rewards. The algorithm aims to maximize the reward for performing any action on the environment [Gha03].

Recently, **Deep Learning (DL)**, a part of ML, has grown exponentially as a reliable methodology to enhance the performance of existing ML based solutions and derive solutions for the problems that were not possible to solve efficiently earlier [Kim+18]. Adaptation towards DL based solutions has been increasing due to its versatility, high generalization capacity, high performance, and application in a wide range of areas [Ana+21].

DL emerged as a part of computer vision for both supervised and unsupervised learning tasks and gained popularity after an event in 2012 [Suz17]. A CNN based model (Section 2.3.2) for ImageNet classification in a computer vision competition outperformed every other model with minimal error rate [Kri+12]. Since then, in almost every technological area, DL based solutions are being developed [LeC+15].

DL based solutions are well suited to medical imaging data. They directly use medical images as input and train themselves according to the given task of classification or detection of diseases. DL based diagnostic systems have helped radiologists and expert medical professionals to provide timely treatment by reducing diagnostic time and deducing more accurate diagnosis [Lun+18; Aki+12]. In the following Section 2.3, we will discuss some basic but essential DL networks used in modern medical diagnostic solutions.

2.3 Networks

In this section, we will discuss some fundamental Neural Networks (NNs) which are crucial in the understanding of any DL based solution. We will start with Artificial Neural Networks (ANNs) (Section 2.3.1) to get a basic understanding of a NN and then look into the specifics of a phenomenal architecture called CNN (Section 2.3.2), which revolutionized the field of DL. Then, we dive into some advanced combinations of NNs such as generator & discriminator (Section 2.3.3) and encoder-decoders (Section 2.3.4).

2.3.1 Artificial Neural Network

Artificial Neural Network (ANN) or NN is a ML method that derives its motivation from the biological design of the human brain. It refers to the broadening of mathematical models that are used to represent organic nerve systems [Abr05]. ANN uses the idea of how a brain functions and can perceive and store information through various neurons. Therefore, to comprehensively understand how an ANN works, it is important to understand how a biological neuron (see Figure 2.8) perceives information.

The human brain consists of massive, deeply connected NNs, through which information travels as electrical signals through biochemical processes. These electrical signals enable humans to perceive their environment (e.g., object recognition) and interact (e.g., moving of objects) with it accordingly. The human brain is capable of processing large amounts of information because of its highly parallel computing structure of neurons.

There are around 10 billion interconnected neurons inside a human brain [Abr05]. On a microscopic level, these neurons are placed head-on one after the other. The information travels as an electric impulse from one neuron to the next. A biological neuron has the following parts which process the information in the form of electrical signals:

- **Dendrites** are the little projections that emerge from the cell body and receive the electrical impulses transmitted from previously connected neurons.

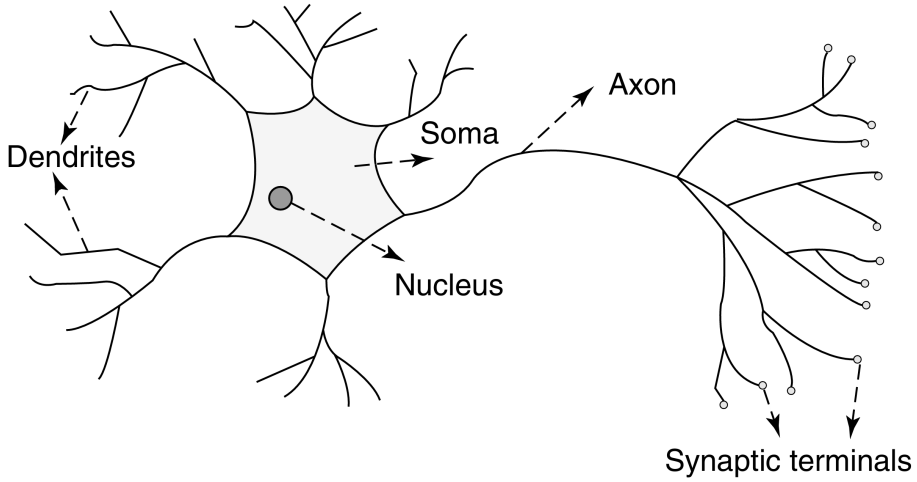
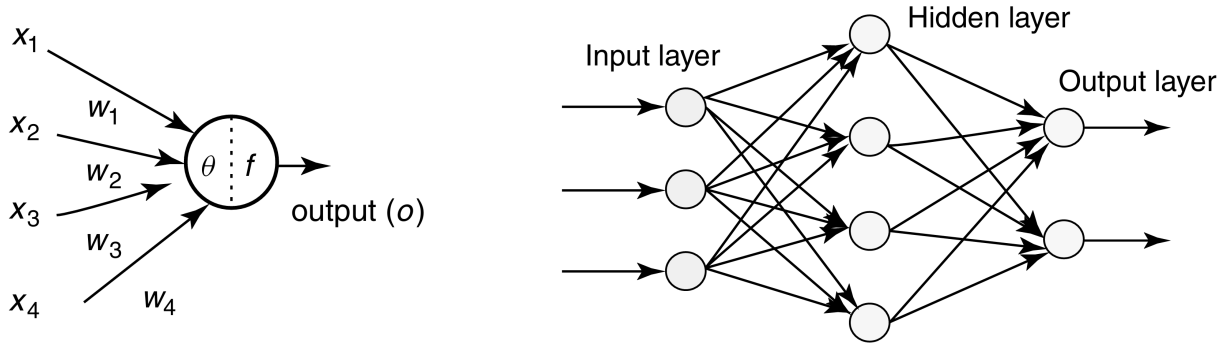


Figure 2.8: Mammalian neuron [Abr05]



(a) Artificial neuron

(b) Multilayered artificial neural network

Figure 2.9: A single artificial neuron and a multilayered neural network. [Abr05].

- **Soma** is the head of the neuron where the **nucleus** is located.
- **Axon** is the neuron's tail, which helps transmit nerve impulses rapidly to the next connected neuron.
- **Synaptic terminals** protrude from the end of the Axon and connect the end of one neuron to the dendrites of the next neuron to pass the electrical impulse through a chemical process.

The basic processing element of ANN is an artificial neuron or **perceptron**. Based on the works of Warren McCulloch and Walter Pitts, Frank Rosenblatt created the first perceptron during 1958 [Ros58]. An artificial neuron takes multiple inputs $x_1, x_2, \cdots, x_n$ and outputs a single output O(see Figure 2.9a). The output signal O is represented by the

following equation:

$$O = f(net) = f\left(\sum_{j=1}^{n} w_j x_j + b\right) \tag{2.3}$$

where the function $f(net)$ is the activation function of the neuron, w_j is the weight associated with input x_j and b is the bias. The variable net is the scalar product of the weight vector and corresponding inputs,

$$net = w^{\mathrm{T}} x = w_1 x_1 + \cdots + w_n x_n \tag{2.4}$$

where T represents the transpose operation applied to a matrix. The output value O is determined based on a specified condition involving the threshold limit θ [Abr05],

$$O = f(net) = \begin{cases} 1 & \text{if } w^{\mathrm{T}} x \geqslant \theta \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Therefore, the output O of a neuron is determined by the cumulative sum of all inputs, taking into account their respective weights, exceeds the threshold value θ. Alternatively, the result is zero. This is represented by Equation (2.5), also known as the *Binary activation function*.

Artificial neurons, inspired by biological neurons, have desirable characteristics to perform complex tasks because of the following properties:

- **Non-linearity** allows better fitting and learning of the underlying structure of the data. Generally, a single perceptron can produce a linear decision boundary but non-linearity can be achieved by using non-linear activation functions. This is supported by the **universal approximation theorem**, which says that in theory, a single hidden layer is sufficient to model any function if a locally bounded and piecewise continuous activation function is used.

- **High parallelism** provides fast processing and tolerance towards hardware failure.

- **Learning** through examples and **adaptivity** to new data allows the system to adapt its internal structure to the changing environment by using updated data.

- The **Generalising nature** of ANNs allows itself to predict the results for unseen data [Bas+00].

Algorithm 1 Perceptron learning rule algorithm [Abr05].

1: Initialize all the connections w_i with random weights.
2: Pass a sample x_i of the training dataset to the input nodes.
3: If the output d_k of the perception is wrong or different from the real value y_k, update all connection weights w_i according to Equation (2.6) for a given learning rate η. This equation calculates the change in weight δw_i required to accommodate the error.
4: Go back to step 2.

Artificial Neural Network Training

An ANN is made up of multiple perceptrons arranged in multiple layers representing a multilayered artificial neural network (Figure 2.9b). An ANN typically comprises three layers: an input layer, a hidden layer, and an output layer. The input layer has all the input neurons, the hidden layer(s) has the neurons where all the learning takes place, and the output layer has the neurons for the network's output.

A neural network learning works on the principle of *Perceptron learning rule*. Algorithm 1 shows how the perceptron learning rule works and updates its weight based on the following equation:

$$\delta w_i = \eta \left(d_k - y_k \right) x_i \tag{2.6}$$

When a **multilayered feed-forward neural network** is trained in *batch* mode, the dataset is divided into smaller batches, and the network updates its weights based on the squared average error (E) computed across each batch (Equation (2.7)). Batch processing is a compromise between processing the entire dataset at once (which can be computationally expensive and memory-intensive) and processing each sample individually (which can lead to noisy updates and slow convergence). By using batches, we achieve a balance that allows for more stable updates and faster convergence, making training more efficient.

After computing the error, the weights (w_{ij}) are then updated according to Equation (2.8) one by one based on the error computed. A partial derivative of the error (δE) with respect to the weights indicates how the error changes in response to small changes in the weights. This gradient is calculated through a process called *backpropagation*, which efficiently computes these gradients for each layer in the network. Once the gradients are obtained, they guide the update of weights in the direction that minimizes the error, a process known as *gradient descent*. While backpropagation calculates the gradients, gradient descent uses these gradients to optimize the network's parameters [Abr05].

$$\mathrm{E} = \sum_{i=0}^{n} \frac{1}{2} (y_i - \hat{y}_i)^2 \tag{2.7}$$

$$\Delta w_{ij}^{(t)} = -\eta * \frac{\delta \mathrm{E}}{\delta w_{ij}} + \alpha * \Delta w_{ij}^{(t-1)} \tag{2.8}$$

where α represents the momentum of the gradient descent, which decides the effect of previous weight changes on the direction toward global minima, η is the learning rate, t represents the current time step or iteration number in the training process.

Certain factors or hyperparameters can be adjusted to improve training and create a robust and reliable neural network [Abr05]. These factors are as follows:

- *Number of neurons* in the hidden layers influences the network's ability to learn and represent complex patterns in the data. More neurons can allow the network to capture more intricate features, potentially improving its performance on the given task. However, too many neurons can lead to overfitting, where the network becomes too tailored to the training data and fails to generalize well to new data. Thus, the number of neurons must be chosen carefully to balance learning capacity and generalization.

- Choosing the *initial weights* of the network in the multi-dimensional weight space is critical in the learning process of the network. During the training of the network, if the weights are poorly initialized, they might lead to poor generalization of the network towards global minima.

- Choosing the *learning rate* controls the change in weights the network is allowed to do during each epoch. Too small a learning rate will slow down the network's ability to learn and a large learning rate might overshoot the global minima and converge too fast, leading to poor generalization.

Activation functions

Activating functions are used in ANNs to get an output from an input signal, which further acts as input for the next layer in the network. They introduce non-linearity in the network to approximate both linear and non-linear data. Examples of such activation functions include Binary (Equation (2.5)), Sigmoid, Tanh, Rectified Linear Unit (ReLU), Softmax, etc [Sha+17].

- *Sigmoid* is the most widely used non-linear activation function whose values range between 0 and 1, making it particularly useful for binary classification problems. It can be defined according to the Equation (2.9). The sigmoid activation function is smooth, differentiable, and non-linear, allowing neural networks to capture complex patterns.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.9}$$

  For inputs far from zero, sigmoid function can lead to the vanishing gradient problem as its derivative becomes very small, slowing down the learning during backpropagation. Additionally, sigmoid can cause output saturation for large input values, leading to near-zero gradients and hindering weight updates, while also contributing to internal covariate shift, which slows down training convergence.

- The hyperbolic tangent function, often known as *tanh*, is comparable to the sigmoid activation function except that tanh is symmetric around the origin, having values in the range -1 to 1 represented by the Equation (2.10). The tanh activation function is smooth, differentiable which can improve optimization efficiency. Its zero-centered output helps with faster convergence, especially in deep networks, and its non-linearity allows the network to capture complex relationships in the data.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.10}$$

  This activation function can suffer from the vanishing gradient problem, particularly for large positive or negative inputs where it saturates and gradients approach zero, slowing down learning in deep networks. Additionally, tanh can contribute to internal covariate shift, necessitating careful initialization and normalization techniques to address this issue.

- *ReLU* output zero only when the output of the linear transformation is zero. It is a good option because of its trait that not all neurons are stimulated at the same time. Equation (2.11) represents ReLU mathematically. The non-linearity of ReLU enabling the network to learn complex patterns efficiently. Its sparse activation only stimulates neurons with positive inputs, improving efficiency and reducing overfitting, while also avoiding the vanishing gradient problem, which allows for more consistent

weight updates in deep networks.

$$f(x) = \max(0, x) \tag{2.11}$$

ReLU activation function can encounter problems like "dying ReLU", where neurons become inactive if they consistently output zero, resulting in zero gradients and halting learning for those neurons. Additionally, while ReLU helps reduce the vanishing gradient issue, it doesn't fully address internal covariate shift, though techniques like batch normalization can help mitigate this.

- *Softmax* is an activation function that combines many sigmoid functions. The softmax function is used for multi-class classification problems, as opposed to binary classification problems, which are handled by Sigmoid. Equation (2.12) output the values between 0 and 1 for each input, representing their probability for each class. The Softmax function converts raw scores (logits) into a probability distribution where the probabilities sum to 1, making it ideal for multi-class classification tasks. It offers a probabilistic output that aids in decision-making and model confidence evaluation, and it can handle multiple classes in a single output layer by assigning probabilities to each class.

$$\sigma(x_j) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}} \qquad \text{for } i = 1, \cdots, J \tag{2.12}$$

Softmax function can be computationally expensive when dealing with a large number of classes, as it requires exponentiating all class scores and computing their sum. Additionally, it is sensitive to class imbalances, often assigning higher probabilities to more frequent classes, which can lead to biased predictions.

Initialization Techniques

The goal of an ANN during its training is to find a solution to an optimization problem by reaching the global minima of the given problem. The algorithm iterates over the training data to reach the global minima and continuously updates the network weights. When dealing with large amounts of data, this process can be somewhat time-consuming. When the network is initialized with sensible weights, algorithms converge faster as the weights of the network are already close to the global minima in the weight space. Examples of such initialization techniques are Random Initialization, Data-driven initialization, and Initialization by pre-training [Nar+22].

In *random initialization*, the network's weights are either initialized randomly or based on an optimal interval with a maximum bound. These random values are sometimes even scaled to maintain the variance between the input and output layers of the network.

Data-driven initialization technique involves finding the weights of the network that are faster to converge for the given specific problem and data. It also tries to find weights that help avoid problems such as vanishing or exploding gradients.

The method of *initialization by pre-training* is widely used to obtain the network weight by pre-training the network in an unsupervised training task. This unsupervised training helps in finding weights that are sub-optimal. It ensures effective optimization and faster convergence.

Optimization Techniques

Training a NN on tasks such as Natural Language Processing (NLP), image classification, etc, is computationally expensive. Therefore, it requires various optimizations to improve the network's accuracy, speed, and stability [Shu23]. Some of these optimizations include Stochastic Gradient Descent (SGD), Adaptive Gradient (AdaGrad), Adaptive Moment Estimation (ADAM) [Qi+19], etc.

SGD [Shu23] is a fast and easy-to-implement optimization technique. It optimizes the model parameters $\theta^{(t)}$ in the direction of the negative gradient (∇) of the objective function $L(\theta)$ with respect to the batch size $\mathcal{B}$ of training samples. After estimating the gradient as shown in Equation (2.13), model parameters $\theta^{(t+1)}$ are updated according to Equation (2.13).

$$g^{(t)} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla L_i \left(\theta^{(t)}\right) \tag{2.13}$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta . g^{(t)} \tag{2.14}$$

ADAM is a combination of AdaGrad and momentum techniques [Kum+20]. It is one of the best optimization approaches in DL. It updates the weights w^{t+1} of any epoch at $t+1$ as shown in Equation (2.19). The use of gradients during the implementation of ADAM enables it to calculate the results faster and makes it memory efficient. It also avoids the issue of getting stuck in local minima by leveraging the concept of momentum.

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1 - \beta_1) \nabla_w L^{(t)} \tag{2.15}$$

$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1 - \beta_2) \left(\nabla_w L^{(t)}\right)^2 \tag{2.16}$$

$$\widehat{m}_w = \frac{m_w^{(t+1)}}{1-(\beta_1)^{(t+1)}} \tag{2.17}$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1-(\beta_2)^{(t+1)}} \tag{2.18}$$

$$\mathrm{w}^{t+1} \leftarrow w^t - \eta \frac{\widehat{m}_w}{\sqrt{\hat{v}_w} + \epsilon} \tag{2.19}$$

In Equations (2.15) and (2.16), β_1 and β_2 control the exponential decay, $\widehat{m}_w$ in Equation (2.17) is the moving averages of the gradient, $\hat{v}_w$ in Equation (2.18) is the squared gradient, and ϵ is a small scaler to prevent decision by 0 in Equation (2.19) [Kin+14].

Loss Functions

During the training of a NN, it is crucial to penalize the network for making incorrect predictions to help effective learning. Some commonly used loss functions include Huber loss, Mean Squared Error (MSE), Binary Cross Entropy (BCE) loss, Mean Absolute Error (MAE), etc.

MSE loss [Lax+18] function is represented as the squared difference between expected output $o_n^{(i)}$ and predicted output $y_n^{(i)}$. It is commonly used in regression tasks and is given as follows:

$$L_{MSE} = \frac{1}{M}\sum_{i=1}^{M}\sum_{n}\frac{\left(y_n^{(i)} - o_n^{(i)}\right)^2}{N} \tag{2.20}$$

where M is the training sample count and N is the class count.

BCE loss [Ho+19] is a loss function used in binary classification tasks. The standard BCE loss is given by Equation (2.21). It measures the dissimilarity between the actual binary label y and the predicted probabilities $\hat{y}$.

$$L_{BCE}(y, \hat{y}) = -[y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})] \tag{2.21}$$

For calculating loss in multi-class classification, the sum of all individual BCE losses are minimized as shown in Equation (3.8).

2.3.2 Convolutional Neural Network

A Convolutional Neural Network (ConvNet) [LeC+89] is a multilayered deep neural network architecture. Convolutional, pooling, and Fully-Connected (FC) layers are the three types

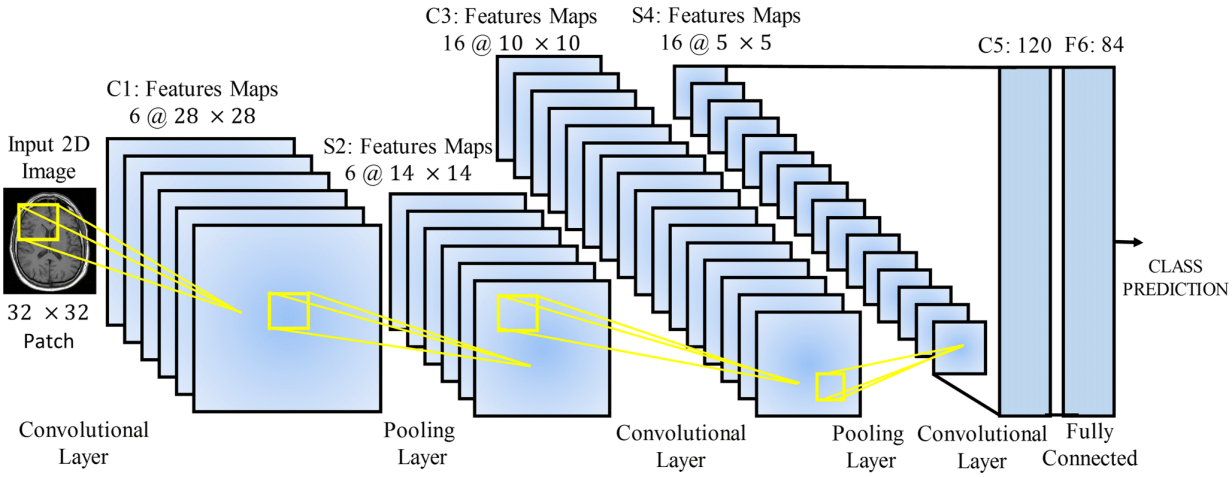


Figure 2.10: A typical CNN architecture for medical image classification [Anw+18].

of layers that make up CNNs in most cases. [Zha+17a]. A combination of these layers forms a CNN architecture. A typical CNN architecture with a combination of these three layers along with an input and output layer predicting an output class is shown in Figure 2.10. CNN is the backbone to many advanced and latest architectures like ZFNet[Zei+14], VGGNet [Sim+14], GoogleNet [Sze+15], and ResNet [He+16].

A **Convolutional Layer** takes local rectangular patches as input from the input layer (in the initialization layer) and *feature maps* (in the hidden layers). A 2D convolution with a filter is then applied on these patches [Zha+17b]. The resulting convolution is fed into a ReLU unit to increase the speed on that network as ReLU ($max(0, x)$) introduce sparsity in the network and reduce the number of calculations [Kri+12]. In the convolutional layer, the same filter maps are shared between one type of feature map, and different filters are assigned to different types of feature maps. This allows the network to detect patterns in an image [Zha+17b]. These shared and sparse kernels in the convolution layers reduce the number of parameters significantly compared to FC layers. Additionally, convolutional layers exhibit equivariance, ensuring that input transformations like translations result in corresponding changes in the output. This makes them ideal for processing structured data like images.

Pooling is an important concept of CNN, which reduces the burden of the network by simply down-sampling or summarizing features along the spatial dimensionality of the given input. This reduces the number of parameters within that activation [Osh+15]. There are a couple of techniques to apply pooling like L_P, mixed, stochastic, mixed, and max pooling, etc [Gu+18].

FC Layer along with the output layer gives the classification probability of the given input. A feature vector is created by combining the feature maps in the initial FC layers. Neurons equal to the number of classes are present in the final FC layer. They combine with an activation function like Softmax to produce classification scores [Zha+17a].

To build a model with high accuracy, initialization of the network weights and tuning of the network hyperparameters like Learning Rate (LR), batch size, Number of epochs, optimization strategy, etc., is required. During training, the network updates these weights till a stopping criterion is met or the Number of epochs is completed. The model with the lowest possible validation loss is chosen as the final network after training is finished [Zha+17a].

2.3.3 Generator & Discriminator in GANs

Generative Adversarial Networks (GANs) are networks that are commonly used for image synthesis tasks such as image translation, generation, super-resolution, etc [Li+21]. These two concepts of Generator $G(z)$ and Discriminator $D(x)$ work in a minimax game to generate a realistic image as represented in Equation (2.22). This type of training in the network in a competing environment is called an Adversarial Network [Liu+19].

$$\min_{G} \max_{D} \left\{ \mathop{\mathbb{E}}_{x \sim \mathcal{D}_{tr}} [\log\ D(x)] + \mathop{\mathbb{E}}_{z \sim \mathcal{P}_z} [\log\ (1 - D(G(z)))] \right\} \tag{2.22}$$

During the training of the GAN architecture, the purpose of the **Generator** G is to update itself regularly to learn the underlying distribution of the real data. It generates fake data by converting random noise to a complex data distribution and adding it to the real data as shown in Figure 2.11. Generator G converts the input z into a fake image to fool the Discriminator D by optimizing the following objective function given in Equation (2.23), where f_G is the loss of the Generator [Li+21].

$$\mathcal{L}_G = \mathbb{E}_{z \sim p(z)} [f_G(-D(G(z)))] \tag{2.23}$$

The **Discriminator** D gets two types of images as inputs, either a real image or a fake image generated by G. Its task is to differentiate between both the types of images and try to win over the Generator G in a minimax game. To accomplish their optimization goals, the Generator and Discriminator are updated alternately using adversarial terms. Discriminator optimizes itself based on both losses calculated from real data and fake data as represented in Equation (2.24) [Li+21].

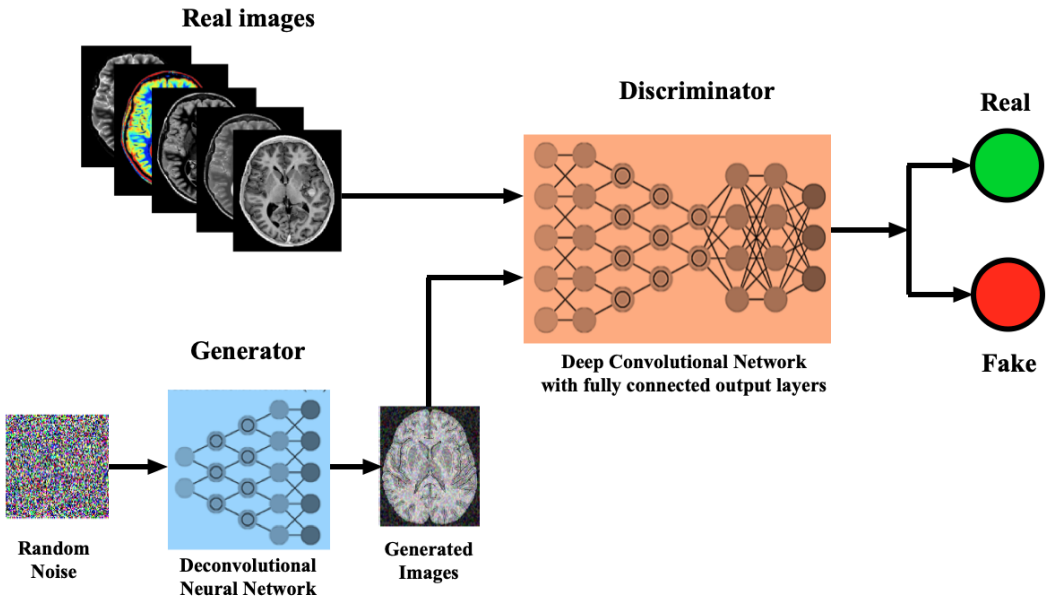


Figure 2.11: The architecture of a GAN involves a Generator that attempts to deceive the Discriminator in distinguishing between "real" and "fake" images while both entities engage in a competitive relationship. [3]

$$\mathcal{L}_D = \underbrace{\mathbb{E}_{x\sim p_{\text{data}}}\left[f_D\left(-D\left(x\right)\right)\right]}_{\mathcal{L}_{D_{real}}} + \underbrace{\mathbb{E}_{z\sim p(z)}\left[f_D\left(D\left(G\left(z\right)\right)\right)\right]}_{\mathcal{L}_{D_{fake}}} \tag{2.24}$$

2.3.4 Encoder-Decoder Structures

Encoder-Decoder structure is a part of DL that is widely used in the development of various NN architectures. It is made up of two architectures: an Encoder and a Decoder. The idea behind this is to make machines capable of understanding complex types of data and use it for efficient processing of the data. To accomplish this, features are extracted from the data using an encoder to create a simpler representation of it. This simpler representation is then later used by a decoder to generate the required output (see Figure 2.12). Some examples of real-world applications based on encoder-decoder architectures are transformer models [Vas+17], text/image/video data summarization [Kum24], machine translation [Cho+14], image captioning [Xia+19], etc.

Encoder represents the first part of this architecture. It takes the input as a sequence or an image and converts it into a latent representation based on the application it is being

[3] Types of Deep Neural Networks (https://mri-q.com/deep-network-types.html).

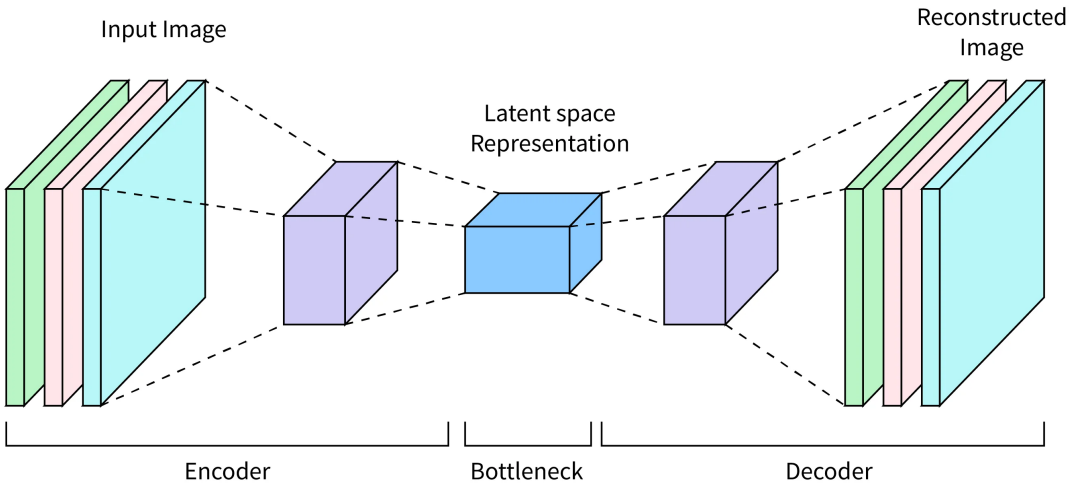


Figure 2.12: Structure of an Encoder-Decoder architecture. [4]

used for. For example, an image will be represented numerically by its latent vector having information about its most important features [Kum24]. This latent vector is then passed through multiple layers of the network for dimensionality reduction. This reduces the size of the vector but stores the same level of information about the data, which is further read by the decoder later [Ali23].

The **Decoder** then takes this latent representation of the data and generates an output sequence based on the required application. Consequently, it initially uses a variety of sampling approaches to increase the dimensionality of this reduced data before reconstructing the data from this underlying representation. After the reconstruction of the latent representation of the data, the features are then converted to the required output in the later layers of the decoder [Ali23; Kum24]. Lately, for the relevant reconstruction of the data, *attention mechanism* is being used in the encoder-decoder architectures.

2.4 Architectures

In this section, we will discuss some critical architectures that are used in this thesis. Section 2.4.1 shows the working of U-Net architecture, which is a significant part of PriCheXy-Net architecture [Pac+23b]. As a discriminator, Residual Neural Network (ResNet)-50 architecture is discussed in Section 2.4.2 and is used along with PriCheXy-Net [Pac+23b] in this work. Swin Transformer discussed in Section 2.4.3 is used as an auxiliary classifier

[4] Autoencoders (https://www.scaler.com/topics/deep-learning/autoencoders/).

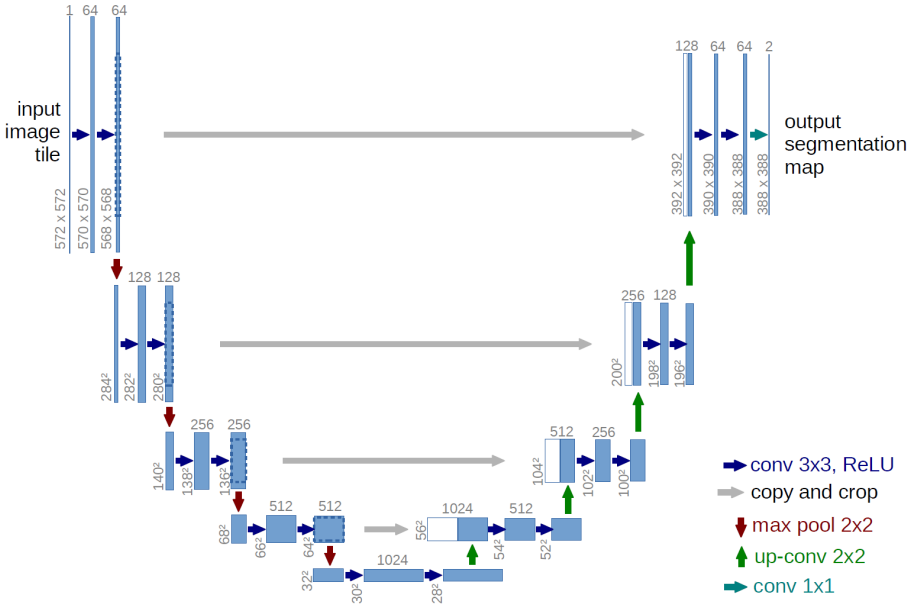


Figure 2.13: The multi-channel feature maps are displayed in blue frames in the U-Net architecture described above. The number of channels is stated on the top of each unit. The dimensions of the box (x and y) are indicated in the lower left corner. White boxes are feature maps that have been duplicated. Diverse operations are illustrated by arrows [Ron+15].

in modified PriCheXy-Net [Pac+23b]. Another significant architecture is SNN [Pac+22], discussed in Section 2.4.4, which has state-of-the-art performance in performing linkage attacks.

2.4.1 U-Net

U-Net architecture has been used in various NN architectures and has shown its robustness in various biomedical semantic segmentation applications [Ron+15]. It has shown high accuracy, especially with imaging applications where the data was labeled and belonged to binary classification applications [Har+21].

The architecture of U-Net is a 'U' shaped architecture as shown in Figure 2.13. The descending part of the U-Net represents an encoder or contraction path, and the ascending part represents the decoder or expansion path. Each encoder convolution layer works in conjunction with its decoder counterpart layer, which complements it in the network for efficient data processing. Each combination of this encoder-decoder structure in U-Net creates a unique feature tailored to each segmented class [Har+21].

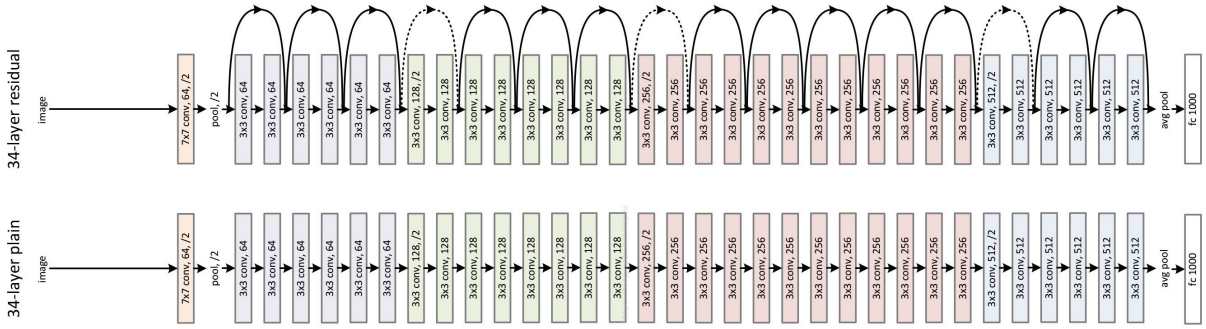


Figure 2.14: Architecture of ResNet-50. For simplicity, the two representations show only 34 layers. A ResNet-50 residual network with 34 parameter layers is represented by the bottom architecture, and a standard network with 34 component layers is represented by the top architecture. An increase in dimensions is shown by the dotted shortcuts [He+16].

The U-Net architecture's contracting path, as illustrated in Figure 2.13, comprises two 3x3 unpadded convolutions that are repeated, followed by a ReLU and a 2x2 max pooling with stride 2 for downsampling. In the process of expanding the remaining half of the network, the feature maps are upsampled, and one 2x2 and two 3x3 convolutions are conducted, each followed by a ReLU. Each 64-component feature vector is mapped to the necessary number of classes using a 1x1 convolution in the final layer of U-Net. So, the network now contains a total of 23 convolutional layers [Ron+15]. Generally, a U-Net can have as many layers as required according to the respective tasks.

2.4.2 ResNet-50

ResNet is one of the most famous DL architecture introduced by Microsoft in 2015 through a research paper [He+16]. ResNet-50 architecture is the modification of fifty deep NN layers in the ResNet architecture, which has been trained on one million images from the ImageNet database.

ResNet paper introduced by [He+16], popularized the use of *Skip Connections* [Bis95] in NNs. Skip connections improved the performance of NN architecture by just skipping any layer that has a negative impact on the network's performance [Muk22]. This can be formally represented by $y = F(x) + x$, where y and $F(x)$ are the outputs of the final and previous layers, respectively, and x is the skip connection. Most ResNet models use double or triple-layer skips with ReLU nonlinearities and batch normalization [Ike+21]. Along with the use of skip connections, ResNet-50 architecture makes use of a convolutional block

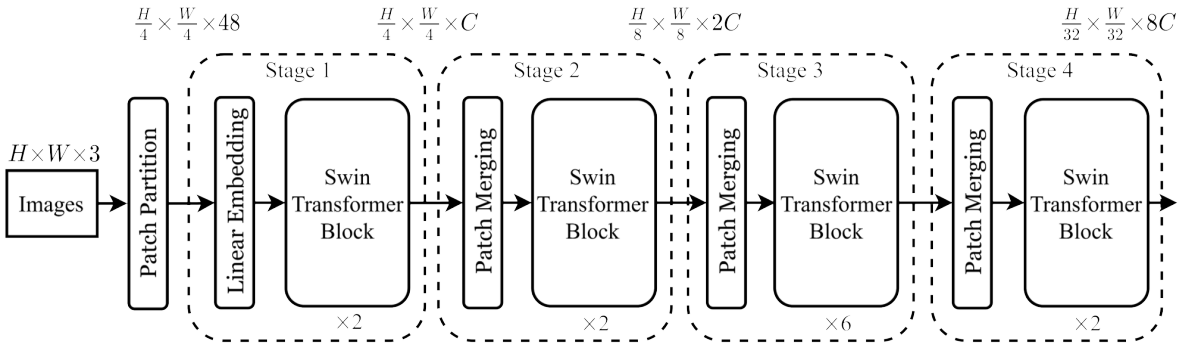


Figure 2.15: The architecture of a Swin Transformer [Liu+21].

embedded with average pooling. A softmax activation function is used for classification applications in the last layer of ResNet.

ResNet-50 architecture is made up of five convolutional layers namely conv1, conv2_x, conv3_x, conv4_x and conv5_x [Ike+21]. Following the loading of the input picture, it passes through the conv1 layer, a layer with 64 filters and a 7x7 kernel size, and a max pooling layer, which in both scenarios has a stride length of 2. Due to the way residual networks are connected, the subsequent layers of conv2_x are arranged in pairs. This process is carried out up until the fifth convolutional layer, at which point the fully connected layer applies average pooling. Finally, softmax is used for classification [Ike+21]. ResNet-50 architecture is explained more clearly by [Ji+19], where they implemented identification of diseases in tomography images with the help of ResNet-50 architecture.

2.4.3 Swin Transformer

The Swin Transformer (Swin-T) model is a state-of-the-art transformer-based general-purpose CV architecture. It was first introduced by researchers at Microsoft in 2021 in their research paper [Liu+21] and was based on Vision Transformer (ViT). The use of transformers in this architecture and its adaptive computing capability enables Swin-T to achieve higher accuracy in CV tasks such as classifying images and detecting objects [Hug22].

The efficiency in the performance of Swin-T is mainly attributed to the concept of *Shifted Windows* [Liu+21]. In standard transformer-based image classification applications, a global self-attention mechanism is used for overall network computation, which leads to a quadratic complexity. Shifted windows work on the concept contrary to what is used in standard transformers. It uses patches of $M \times M$ local self-attention modules to divide the

complex features into simpler processing parts [Liu+21]. This makes the overall learning of the network faster with targeted feature learning.

A smaller version of Swin-T architecture is shown in Figure 2.15. A patch-splitting module is first used to divide an RGB input image into non-overlapping patches. A linear embedding layer projects the combined RGB values of each patch to a dimension (C). During the four stages of the architecture, Multiple Swin-T blocks incorporating enhanced self-attention are applied to these patches, with resolutions at $(H/4 \times W/4)$, $(H/8 \times W/8)$, $(H/16 \times W/16)$, and $(H/32 \times W/32)$. These stages create a hierarchical representation like VGG and ResNet, making the architecture suitable for various vision tasks [Liu+21]. The last layer of Swin-T can be modified to accommodate multiple attention heads according to the required number of classes in the classification task.

2.4.4 Siamese Neural Network

Siamese Neural Network (SNN) is a NN architecture that has two subnetworks that are exactly the same, with the same parameters and weights, and the same configuration, joined at their output layers as shown in Figure 2.16 [Bro+93]. These two identical sub-networks extract features from the feature vectors during the training of the network. The last layer of this architecture, where the output of the two identical sub-networks concatenates, calculates the similarity or dissimilarity between the given two feature vectors. This results in similarity verification where all the features are rejected as forgeries except those whose similarity score is greater than a threshold. [Bro+93].

The Siamese network has one output whose state value reflects how similar the two patterns are and two input fields for comparing two patterns. SNN architecture in Figure 2.16 represents a patient verification network [Pac+22]. It takes two images x_1 and x_2 as input of size $3 \times 256 \times 256$, which are processed by two identical pre-trained ResNet-50 sub-networks. In this SNN version of [Pac+22], ResNet-50 is modified by replacing its classification layer with a layer consisting of 128 output neurons. This produces z_1 and z_2 feature representations corresponding to each ResNet-50 sub-network. Both the branches are then merged to calculate the absolute difference between the sigmoids of the two feature vectors. Finally, an FC layer reduces the dimensionality to one neuron, which is then further processed by a last sigmoid activation function σ. This finally produces the output as similarity score $\hat{y} \in [0, 1]$ [Pac+22].

SNN architectures have proven to be better than other NN in re-identification tasks as they help learn from semantic similarity by focusing on learning embeddings that place

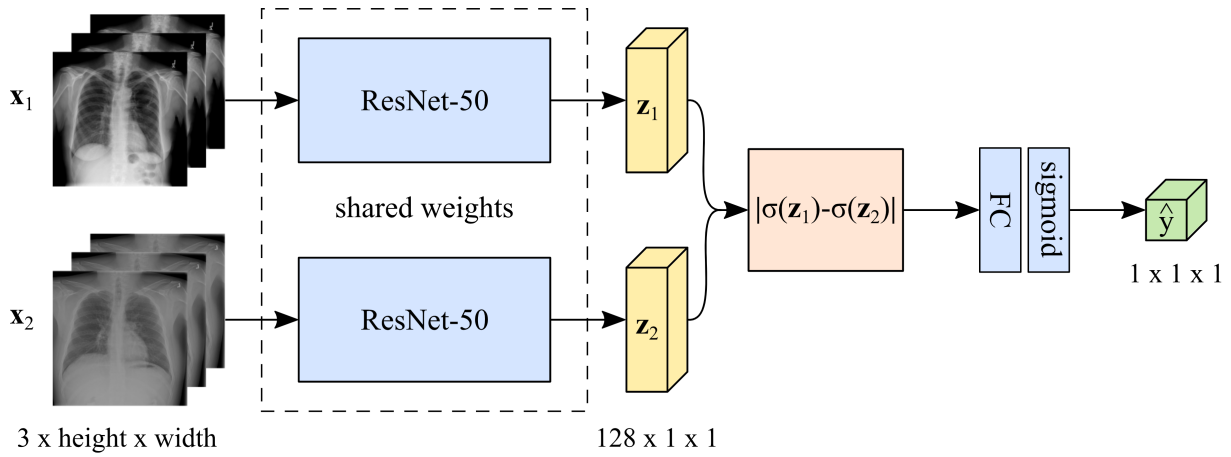


Figure 2.16: SNN architecture [Pac+22] used for patient verification on the ChestX-ray14 [Wan+17] dataset. The two ResNet-50 blocks serve as feature extraction blocks and share the same network parameters. z_1 and z_2 are the feature representations. Orange block represents the merging to produce the final output $\hat{y}$.

the same classes together. SNNs are more robust to class imbalance and therefore have high precision in classification tasks [Ben22]. On the other hand, since SNNs involves learning from quadratic pairs, they are slower during the training and also do not produce probabilities of the output belonging to a particular class [Ben22].

2.5 Multi-class Classification

Classification is one of the most important parts of ML and DL techniques. The categorization of data into their respective classes has been fundamental to DL tasks. In this thesis, the classification of medical images comprises a crucial part where a Swin-T discussed in Section 2.4.3 is modified in the last layer to classify chest x-ray images into 14 different categories of diseases [Tas+22].

A multiclass classification NN model is shown in Figure 2.17. The data has k category of classes which is passed into a NN model. The hidden layers of this model try to understand the underlying structure and categorization of the data. This output, in the case of multiclass classification, is converted into probabilities $P_1, P_2, \cdots, P_k$ by a softmax function. Each data point is then assigned to a class according to Equation (2.25).

$$Class\ i = \underset{i \in [1,k]}{\operatorname{argmax}}(P_i) \tag{2.25}$$

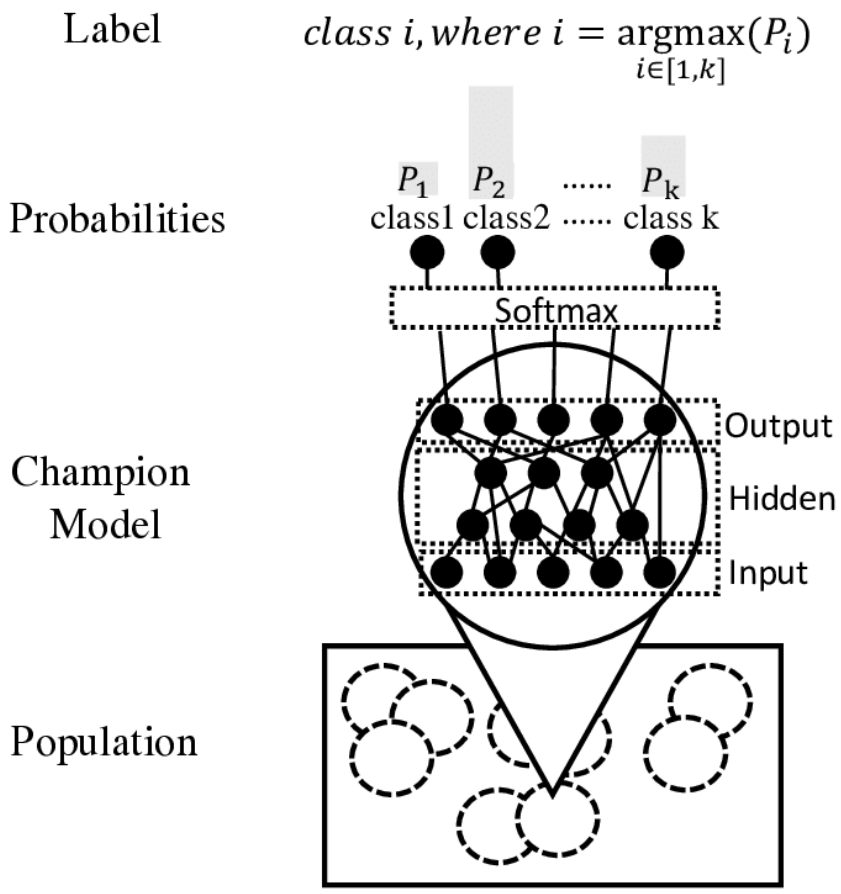


Figure 2.17: A typical architecture of a Multiclass classification model. The data is represented by population, and NN is represented by the champion model. The classification is shown by probabilities, which are further categorized into labels accordingly [Gao22].

Any classification task can be categorized into two types based on the different classes included in the data:

- *Binary Classification* is a type of classification where the data is categorized into only two classes. For this type of classification task, only a single classifier is required. It encodes the two classes as either 0 and 1 or true and false or negative and positive. This is often called *One-Hot-encoding* and can classify whether the data point belongs to one class or the other. The sigmoid activation function is widely used to get the binary classification output in the form of 0 and 1.

- *Multiclass Classification* or k-class classification technique is used to classify the data having two or more class labels. In the case of medical imaging, a multiclass classifier can classify the images into different types of diseases the corresponding patient is affected by. In this type of classification, the number of classifier models depends on the underlying classification technique used [Rif08]. Different techniques in which multiclass classification can be implemented are:

  - *One vs. Rest* classification approach is used when a dataset has N number of distinct classes. N different binary classifiers are built or the output layer of the classification network is adapted to accommodate N number of neurons, each corresponding to one specific class. For the i_{th} class, considering if all the positive examples belong to class i and all the negative examples are not in class i, classification is done in Equation (2.26) [Rif08]. This approach ensures recognition of one specific class and treats all other classes as an entire set of other classes.

$$f(x) = \underset{i}{\operatorname{argmax}}\ f_i(x) \tag{2.26}$$

  - *One vs. One* or *All vs. All* classification approach is also used with a dataset that has N number of distinct classes, the difference being that it builds only $N(N-1)$ number of binary classifiers. In this approach, each classifier tries to distinguish between each pair of classes i and j. One vs. One classification can be represented by Equation (2.27), where f_{ij} is the classifier, i represents the positive class, and j represents the negative class [Rif08].

$$f(x) = \underset{i}{\operatorname{argmax}} \left(\sum_{j} f_{ij}(x) \right) \tag{2.27}$$

Chapter 3

Related Works

The notable advancement in MITs enabled many German clinics to process and store digital datasets. However, properly curating and annotating this digital medical data is time-intensive, hindering the releasing of medical datasets with corresponding free text reports publicly for research purposes [Wol23]. Also, it is crucial for any medical dataset to strictly comply with GDPR rules to avoid any breach of privacy [Eur16].

3.1 Overview of Existing Anonymization Techniques

Over the years, researchers have developed quite a few methods to anonymize the images and introduce privacy [Ruc+16]. Some of these methods have been shown in Table 3.1 with their respective effects in Figure 3.1.

An anonymization system that is used for privacy enhancement in digital images should have the following properties [Max+20]:

- *Anonymization* - The output should fabricate a new identity from the input image in order to conceal the identity of the individual in the original image.
- *Control* - Users have complete control over how the real person and the fake identity are mapped, as the generated images' false identities are managed by a control vector.
- *New identities* - The generated new identities that are not present in the training set must be present in the generated images.
- *Realistic* - The output must appear realistic in order to be utilized by advanced detection and recognition systems.

Filter	Parameter
Blackener	Opacity
Pixelization	Size of squares
Gaussian Blur	Standard deviation
Gaussian Noise	Standard deviation
Average Blur	Neighbouring area
Motion Blur	Length and angle of the motion
Speckle noise	Standard deviation
Salt and Pepper Noise	Noise density

Table 3.1: Privacy filters along with the name of the corresponding method [Ruc+16].

Figure 3.1: Pictures of faces obscured with filters. Clean faces, blackener, pixelization, gaussian blur, and gaussian noise are arranged from left to right on top; average blur, motion blur, speckle noise, and salt and pepper noise are arranged on the bottom [Ruc+16].

- *Temporal consistency* - For tasks such as action recognition or person tracking, it is important to guarantee pose retention and temporal consistency in films.

In the following sections, we will discuss various anonymization techniques that anonymize private information embedded in the image. These methods are categorized into three major categories. In the Section 3.2, We will discuss all the traditional obfuscation methods, followed by Section 3.3 in which effects of methods like Differential Privacy (DP) on the privacy of images are discussed. Finally, advanced techniques like PriCheXy-Net and GAN based methods are discussed in Section 3.4.

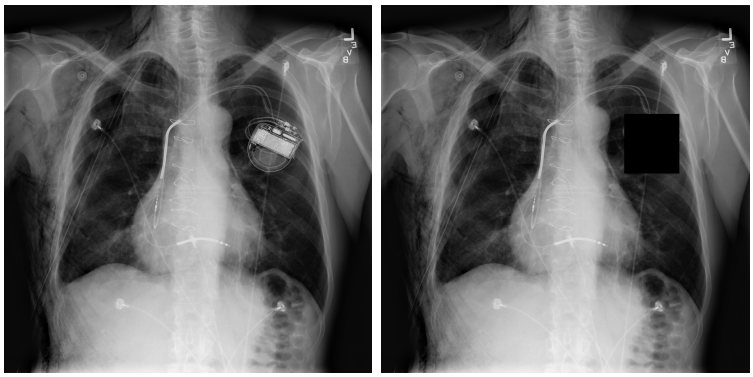


Figure 3.2: Anonymization through black-box where information of pacemaker (on left) inside the chest is removed by introducing black-boxes on the image (on the right-hand side).

3.2 Conventional Anonymization Techniques

Conventional anonymization methods or traditional obfuscation methods include techniques as follows:

- *Blackbox* application [Pac+23b] is a method used to hide information in an image that could potentially harm the privacy of a person. In medical imaging, this technique can be used to hide visual information that is private to a patient, e.g., necklaces, pacemakers, jewelry, or any other visual artifact that is personal to the patient. To hide these types of visual information, black or white [Fu+00] boxes of various sizes are directly applied over the images as shown in Figure 3.2. A major disadvantage of this technique is that applying black boxes on a medical image could potentially hide critical information about the disease and make the image useless for diagnosis.

- *Blackener* method [Ruc+16] involves darkening of an image (see Figure 3.1) to hide the details of an image. This produces a darker image with fewer details of the object. The darkness of the image produced by this method is controlled by an 'opacity' filter as shown in the Equation (3.1), where opacity is represented by α. α reduces the value of each pixel of the image. The filter's effect increases with increasing α, and the darker the produced image will be. By using this method, one can reduce the sharpness of an image but, in practice, cannot hide the visual information of an image in a true sense. But, blackener combined with pixelation and blurring can highly reduce face recognition, as shown by a study presented by [Kor+13].

$$ImgBlackened = originalImg * (1 - \alpha) \tag{3.1}$$

- *Pixelization* is an image manipulation technique that, when combined with blurring, can help in dealing with privacy issues in digital images [Tho+18]. The pixelization method changes the parts of a picture by making them hard to recognize by reducing the resolution. This is done by dividing the image into $M * M$ non-overlapping squares, where the user chooses M. The average value of that square replaces the pixels in each square according to Equation (3.2), where the block size is b, and the pixel coordinates are x and y. This method can be thought of as a way to downsample an image without changing its size [Ruc+16].

$$I_P(x, y) = \frac{1}{b^2} \sum_{i=0}^{b-1} \sum_{j=0}^{b-1} I\left(\left\lfloor \frac{x}{b} \right\rfloor + i, \left\lfloor \frac{y}{b} \right\rfloor + j\right) \tag{3.2}$$

- *Blurring* has been used as a common way to preserve privacy when sharing digital images [Pul19]. Developing a nearly irreversible blurring algorithm, like the one in [Fan18], was a great step for preserving privacy to significantly slow down or even stop attackers before they reach the unblurring step. There are multiple ways in which blurring can be applied on an image (see Figure 3.1), e.g., gaussian blur [Nix+19], average blur, motion blur, box blurring [Bel+22], etc.

  The Gaussian function and an image are convoluted together to produce a Gaussian blur. It can be applied on an image according to Equation (3.3), where x and y are the pixel coordinates and σ is the Gaussian distribution's standard deviation.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.3}$$

These methods are, therefore, widely used in anonymization applications because of their simplicity and easy implementation [Pac+23b]. On the other hand, various image restoration methods have also been developed and proved to restore images to their original form when manipulated by these conventional techniques [Ruc+16]. Using tools like de-blackening, de-pixelization, and de-blurring mentioned in [Ruc+16], we can reverse the conventional processes discussed above. Their simulations show that face recognition performance is nearly as good as with clean faces, meaning people's privacy can be compromised and is no longer protected. [Pac+22] also shows that using conventional anonymization techniques like blackbox and masking in critical areas of an image can still lead to linkage attacks and failure in safeguarding patient information.

3.3 Perturbation-based Methods

Data perturbation methods can be divided into two main types: the probability distribution approach and the value distortion approach [Liu+05]. In the probability distribution approach, the data is replaced with either another sample from the same (or estimated) distribution [Lie+85] or with the distribution itself [Lef+83]. The value distortion approach directly alters data elements or attributes by either adding noise, multiplying by noise, or using other randomization methods [Ada+89].

Some of the perturbation-based anonymization methods include techniques as described in the following Section 3.3.1 and 3.3.2.

3.3.1 Differential Privacy

Differential Privacy [Dwo+06; Dwo08] is one of the leading state-of-the-art privacy framework for anonymizing statistical datasets. Although it ensures strong privacy for each individual record, implementing standard differential privacy for non-aggregated data still remains a difficult task [Fan18]. Even though previous studies like [Jan+13] showed the limited feasibility of DP on image datasets. [Fan18] shows the feasibility of DP in image data anonymization by creating an extended privacy model and an efficient mechanism to make it work for image data. Over the last decade, a number of variants of DP have been developed for privacy enhancements, e.g.,

- *Local Differential Privacy (LDP)* [Duc+13] extends the idea of differential privacy by ensuring that individual data points are kept private even from the data analyst or learner. This method anonymizes private data and enhances privacy protection for sensitive information. The term "local" indicates that privacy is preserved at the individual level, where data providers ensure that no one can see the sensitive data. LDP directly modifies input data by introducing various types of noise before any further processing. This method allows for more precise control over the privacy of each data point [Don+22].

- *Global Differential Privacy (GDP)* is applied to the computational results subsequent to the data's processing. Some studies like [Liu+20; Wan+20] show the different applications where GDP can be applied and also show that this technique of privacy preservation is rare and difficult to implement in real life.

- *DP within algorithms* - This variant of DP includes all other techniques which leverages the concept of DP to improve their efficiency and privacy factor. Examples of such techniques are Differential Privacy - Stochastic Gradient Descent (DP-SGD) [Raj+12], Differential Privacy - Variational Auto-Encoders (DP-VAE) [Weg+22], etc. DP within algorithms works by training deep NNs which are modified to accommodate DP in the network during the training.

3.3.2 Differential Privacy - Pixelization

Differential Privacy - Pixelization (DP-Pix) [Fan18; Fan19] - The concept of image DP can be used to improve the privacy of standard image anonymization methods. One such method is DP-Pix. DP-Pix method is an improvement over the normal pixelization method discussed in Section 3.2. This method was used as a baseline method in the research conducted by [Pac+23b] for developing privacy-preserving and utility-saving methods.

DP-Pix algorithm works by performing pixelization on an input image as shown in Figure 3.3. This procedure involves calculating the average of the pixel values for each grid cell and then applying a perturbation to the pixelized image. Mathematically, the global intensity of pixelization for each $b \times b$ grid cell is controlled by Δf as represented in Equation (3.4).

$$\Delta f = \frac{255m}{b^2} \tag{3.4}$$

- *ϵ-Image DP* [Fan18; Fan19] method is a modification of DP-Pix method. It samples random noise and adapts it according to the Laplacian mechanism, i.e., adding Laplace noise with 0 mean and $\frac{\Delta f}{\epsilon}$ scale [Dwo+06]. Equation (3.5) represents the definition of ϵ-Image DP [Fan18; Fan19] which states that a randomized mechanism $\mathcal{A}$ gives ϵ-differential privacy if for any neighboring images I_1 and I_2, and for any possible output $\tilde{I} \in Range(\mathcal{A})$, where the probability is taken over the randomness of $\mathcal{A}$. The privacy budget ϵ controls the degree of privacy offered by $\mathcal{A}$, smaller values corresponding to more privacy.

$$\Pr\left[\mathcal{A}\left(I_1\right) = \tilde{I}\right] \leq e^{\epsilon} \times \left[\mathcal{A}\left(I_2\right) = \tilde{I}\right] \tag{3.5}$$

- *m-neighborhood DP* [Fan18; Fan19] is defined as when two neighboring images I_1 and I_2 differ by no more than m pixels, if they have the same dimension. This permits up

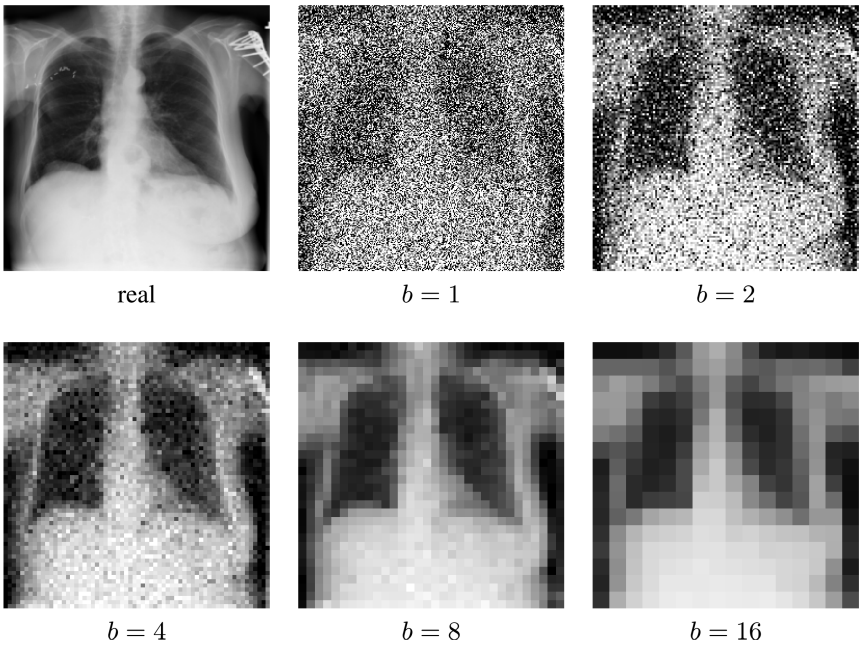


Figure 3.3: Chest x-ray images using the DP-Pix approach at various cell sizes b and m-neighborhoods, with a privacy budget of ϵ equal to 1 [Pac+23b].

to m pixels of difference between two adjacent images to help safeguard a patient's biometric information, depicted by those pixels in an image. In practice, a suitable m-neighborhood is selected by the data owner, allowing them to determine the degree of privacy protection [Pac+23b].

3.4 Adversarial Anonymization Strategies

An adversarial network is a combination of two or more NNs that work opposite to each other in order to improve the generated output. Recently, several researchers have leveraged this idea of adversarial networks to develop anonymization systems and enhance the privacy of publicly available data. For instance, there is considerable research being done in this area on the creation of synthetic images while guaranteeing anonymity, with the goal of producing completely anonymous medical image datasets [Pac+23a].

Adversarial anonymization methods include techniques as described in the following Section 3.4.1, Section 3.4.2, and Section 3.4.3.

3.4.1 GANs

Generative Adversarial Networks [Goo+14] provide a strong framework for approximating the complex underlying distribution of data. The GAN framework has shown significant advancements in synthetic image generation [Rad+15], image translation [Zhu+17; Kim+17], and various other applications [Iso+17; Cho+18; Yoo+19].

The key characteristic of the GAN framework is the presence of a generator and a discriminator, which are trained in opposition to each other in an adversarial manner [Yoo+20]. The working of the generator and discriminator inside a GAN architecture is discussed in Section 2.3.3. This framework can be described as a minimax game, where at the optimal point, the generated samples match the real data distribution.

GANs have been utilized in many studies [Shi+18; Hel+24] for privacy utilization purposes in diverse fields. [Yoo+20] anonymized medical data through data synthesis by developing a GAN based anonymization architecture (ADS-GAN). They utilized a modified conditional GANs framework to generate synthetic data, which is constrained by some identity conditions. This model, ADS-GAN, is composed of the Generator and the Discriminator as its two primary parts. Figure 3.4 provides a block diagram of the ADS-GAN model.

AnomiGAN [Bae+19] is another adversarial network developed for anonymizing private medical data. [Bae+19] proposed this GAN-based method that maintains a privacy level that is comparable to DP while offering improved prediction performance. This framework is a robust approach that utilizes any target predictive classifier to preserve the original prediction outcome. This method does not require a large dataset to achieve good prediction results, and it is also not limited to medical data applications.

3.4.2 Privacy-Net

Privacy-Net [Kim+21] is a CNN based adversarial method for segmenting medical pictures with identity obfuscation. It is a client/server privacy-preserving network for multicentric medical image analysis. The Privacy-Net model utilizes adversarial learning to hide the identity of the patient while encoding images and retaining sufficient information for the target task.

Privacy-Net architecture as shown in Figure 3.5, consists of three components:

- A U-Net [Ron+15] based encoder network G parameterized by θ_G that eliminates distunguishing features from input patient images. While training the system, the

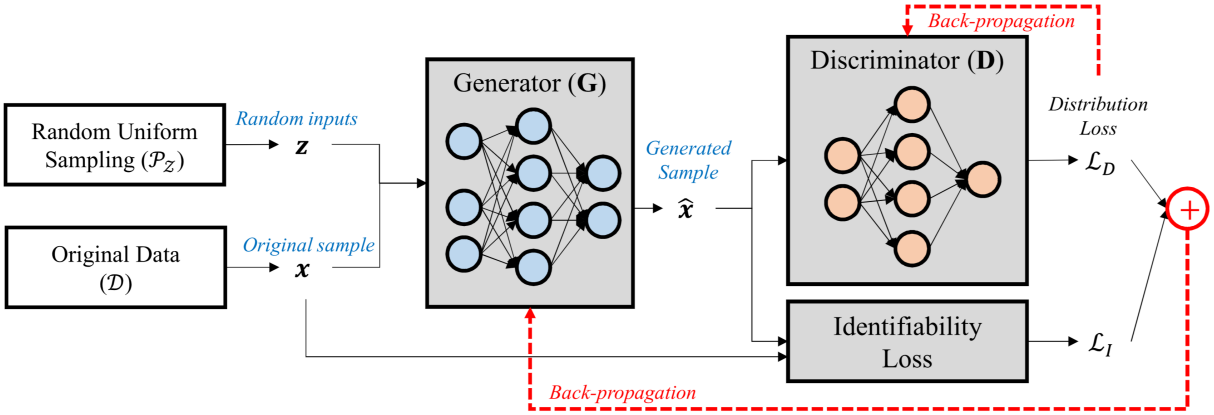


Figure 3.4: Block diagram of ADS-GAN. To create the generated sample ($\hat{x}$), the generator takes two inputs: the original sample (x) and a random vector (z). The generator receives a backpropagation of the sum of the distribution loss ($\mathcal{L}_\mathcal{D}$) and the identifiability loss ($\mathcal{L}_\mathcal{I}$). Multi-layer perceptrons are used in the construction of the discriminator and the generator both [Yoo+20].

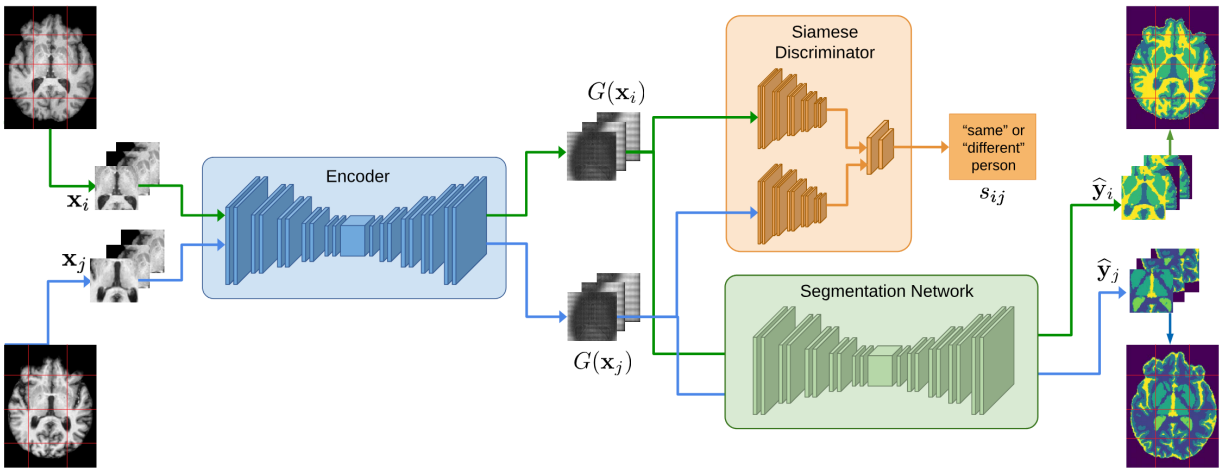
encoder is fed with the pair of images (x_i, x_j) and returns two encoded images $G(\mathrm{x}_i)$ and $G(\mathrm{x}_j)$. The output of the encoder is a feature map $G(\mathrm{x}) \in R^{H\times W\times D}$, which can be seen as an encoded version of the input image.

- An SNN based discriminator network D with parameters θ_D that tries to decode the images in order to determine the subject; in other words, it attempts to find out whether or not the two images are of the same patient.

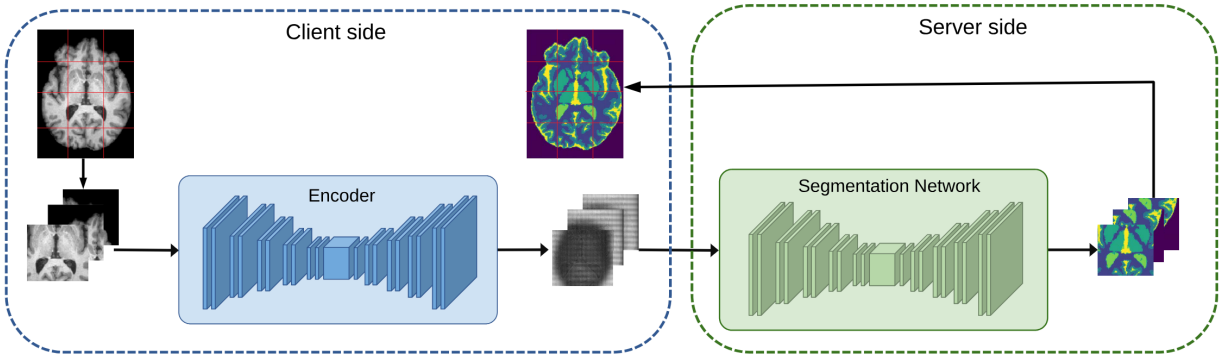
- And a medical image analysis network based on CNN represented by S having parameters θ_S that examines the inner details of the encoded images, specifically for segmentation. The objective is to estimate an accurate segmentation map. The value of y may be determined based on the encoded image $G(\mathrm{x})$.

Similar to the most famous adversarial networks, Privacy-Net uses two loss functions that are optimized together in opposite directions. During the training of the network, a segmentation loss and an adversarial discriminator loss are calculated as shown in Equation (3.6) to estimate the total loss of the network [Kim+21].

$$\begin{aligned}\min_{\theta_G,\,\theta_S}\ \max_{\theta_D}\ \mathcal{L}(\theta_G,\theta_S,\theta_D) =& \mathbb{E}_{\mathrm{x},\mathrm{y}\sim P(\mathrm{x},\mathrm{y})}\left[\mathrm{l}_S(S(G(\mathrm{x})),\mathrm{y})\right] - \\ & \lambda\mathbb{E}_{\mathrm{x}_i,\mathrm{x}_j\sim P(x)}\left[\mathrm{l}_D(D(G(\mathrm{x}_i),G(\mathrm{x}_j)),\mathrm{s}_{ij})\right]\end{aligned} \tag{3.6}$$



a) Training configuration



b) Test (deployed) configuration

Figure 3.5: (a) Training configuration of Privacy-Net architecture with an Encoder, a Siamese Discriminator network, and a Segmentation network. (b) During testing, the segmentation network is set up on the server, the discriminator is removed from the system, and the encoder processes each image individually on the client end [Kim+21].

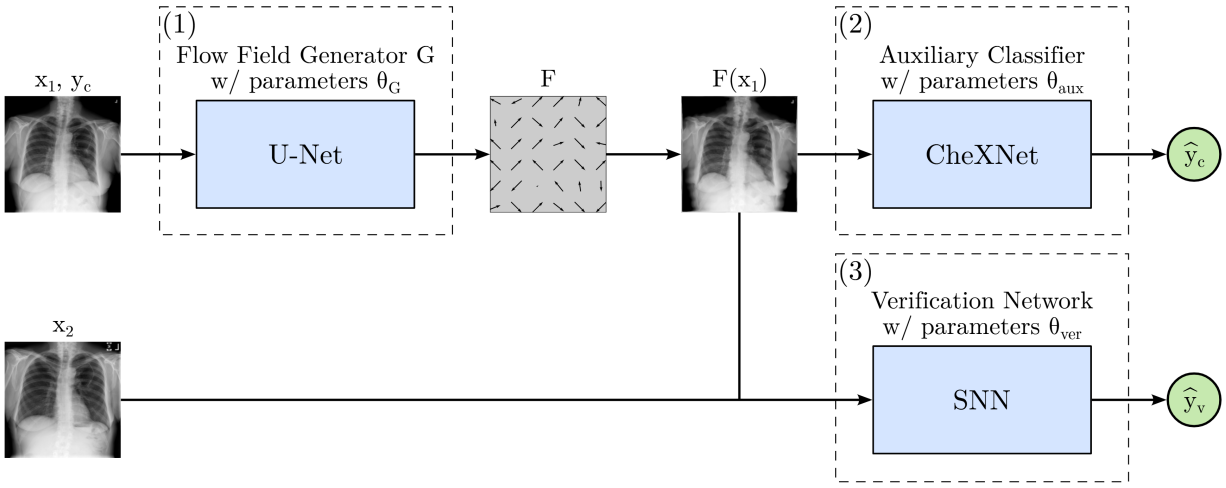


Figure 3.6: PriCheXy-Net architecture for adversarial image anonymization [Pac+23b].

The encoder acquires the ability to eliminate privacy-related characteristics while preserving the ones necessary for the intended objective by concurrently fooling the discriminator and optimizing the medical analysis network. However, While Privacy-Net's original purpose was to preserve utility in MRI segmentation tasks, there is little direct application and transferability of Privacy-Net to other imaging modalities and downstream applications such as chest X-ray categorisation [Pac+23b]. One of the disadvantages of the Privacy-Net method is that the segmentation maps may still include patient-identifiable data that is vulnerable to re-identification attacks, in contrast to classification outputs.

3.4.3 PriCheXy-Net

PriCheXy-Net [Pac+23b] architecture is a state-of-the-art privacy-preserving model that also keeps the utility of images high. Therefore, this architecture is used as a baseline model in this thesis.

PriCheXy-Net architecture is a combination of three different independent NNs:

- *U-Net Generator G* parameterized by θ_G that predicts a flow field F, which is used to deform the original image x_1 of the abnormality class y_c. A Tanh activation function replaces the last layer of U-Net to forecast a 2-channel flow field F with $[-1, 1]$ as its boundaries. At the time of training, to constrain the learned deformations from F, a weighting factor μ is applied according to $F = F_{id} - \mu F$. This does not F to change the original image identity F_{id} too much, therefore maintaining its utility. After an epoch of training, the individual losses of auxiliary classification and verification

network are used to calculate the total loss of the U-Net generator. Equation (3.7) minimizes the total loss of the PriCheXy-Net architecture [Pac+23b].

$$\arg\min_{\theta_G} L\left(\theta_G, \theta_{aux}, \theta_{ver}\right) = L\left(\theta_G, \theta_{aux}\right) + L\left(\theta_G, \theta_{ver}\right) \tag{3.7}$$

- *Auxiliary Classifier* network takes its motivation from CheXNet [Raj+17] - a DenseNet [Hua+17] based classifier consisting of 121 layers. It takes the modified image $F(x_1)$ as input and classifies the anonymized image into 14 categories of class predictions $\hat{y}_c$, indicating the presence or absence of each abnormality included in the ChestX-ray14 dataset. The parameters of this auxiliary classifier, θ_{aux}, are initialized using a pre-trained model having a mean AUC of 80.5%. Auxiliary classifier loss $L_{aux}\left(\theta_G, \theta_{aux}\right)$ is a class-wise BCE represented by Equation (3.8).

$$L_{aux}\left(\theta_G, \theta_{aux}\right) = -\sum_{i=1}^{14}\left[y_{c,i}\ \log\left(\hat{y}_{c,i}\right) + \left(1 - y_{c,i}\right)\ \log\left(1 - \hat{y}_{c,i}\right)\right] \tag{3.8}$$

- *Siamese Neural Network* is used as a verification network with θ_{ver} representing its parameters. SNN [Pac+22] architecture consists of two ResNet-50. It receives the two images as inputs: a deformed image $F(x_1)$ as well as one real image x_2 from either the same or a different patient. It produces the similarity score $\hat{y}_v$ for patient verification in the range of $[0, 1]$ representing the same or different patient. For initializing the network parameters θ_{ver}, the pre-trained weights from [Pac+22] with a resulting AUC of 99.4% for a patient verification task. The auxiliary verification network is optimized with a log-likelihood verification loss as represented in Equation (3.9).

$$L_{ver}\left(\theta_G, \theta_{ver}\right) = -\log\left(1 - \hat{y}_v\right)\ \text{, with}\ \ \hat{y}_v = \text{SNN}\left(F\left(x_1\right), x_2\right) \tag{3.9}$$

Overall, the U-Net in PriCheXy-Net serves as an anonymization network intended to mask biometric data using controlled visual distortions. Concurrently, the patient verification model and the auxiliary classifier offer direction for optimizing the flow field generator through a minimax game.

The PriCheXy-Net architecture has shown promising results in anonymizing medical images to preserve privacy while at the same time maintaining their utility. Compared to the baseline, PriCheXy-Net has performed significantly better. At a deformation degree of $\mu = 0.01$, the classification score of PriCheXy-Net is 76.2%, resulting in high utility.

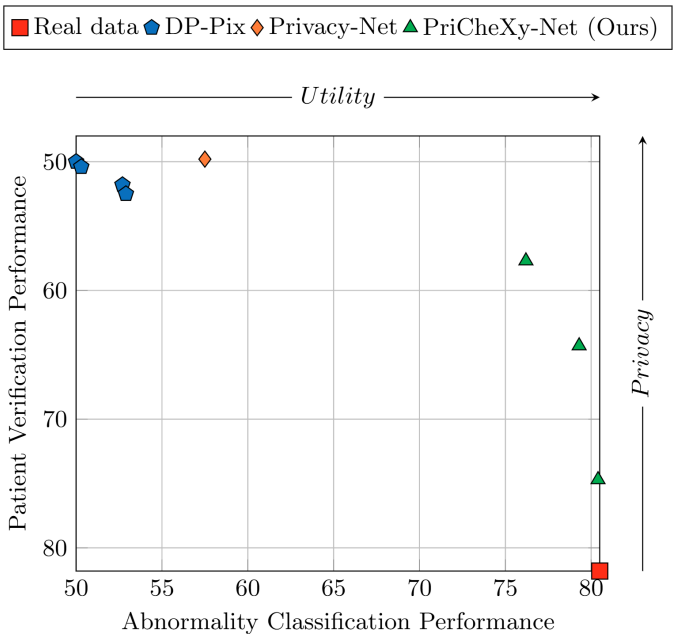


Figure 3.7: Privacy-utility trade-off results of PriCheXy-Net in comparison with DP-Pix and Privacy-Net. The level of data utility is represented by the abnormality classification performance on the x-axis, while the degree of privacy is measured by the patient verification performance on the y-axis [Pac+23b].

With a verification score of $57.7\% \pm 4.0\%$, PriCheXy-Net is highly effective in anonymizing private information in medical images [Pac+23b]. A comparison of PriCheXy-Net with other anonymization methods in terms of Privacy-Utility trade-off is shown in Figure 3.7.

Chapter 4

Methodology

4.1 Dataset Description

As discussed earlier in Section 2.1.1, chest X-ray radiography is among the most frequently used radiographic examination methods for the detection and diagnosis of various respiratory diseases. To facilitate the need for large datasets in modern DL based CAD systems, various researchers collected and curated these X-rays and applied advanced techniques to label this huge medical data accurately. Some of the most famous and large datasets compiled for chest-related diseases are mentioned in Table 4.1.

- **ChestX-ray14** [Wan+17] - Initially launched by National Institutes of Health (NIH) as **ChestX-ray8** dataset comprising 108,948 frontal-view chest X-ray images from 32,717 patients. ChestX-ray-8 dataset comprises the eight most common thoracic-based diseases, which are labeled and mined from radiological reports using NLP techniques with over 90% accuracy. These diseases include atelectasis, cardiomegaly, effusion, infiltration, mass, nodule, pneumonia, and pneumonia. Each image in the dataset can either have multiple disease labels or be marked as "No Finding" in the case of a healthy patient.

  Later, six more common thorax diseases (i.e., Consolidation, Edema, Emphysema, Fibrosis, Pleural Thickening, and Hernia) were also labeled and added to form ChestX-ray14 dataset comprising 112,120 frontal-view chest X-ray images from 30,805 unique patients. The images are rescaled to 8-bit gray-scale images with a resolution of 1024×1024 pixels. This dataset is crucial for the advancement of deep learning in medical imaging, especially for detecting and localizing diseases in chest X-rays. Therefore, this work uses it as the primary data source.

Dataset	Images	Patients	Images per patient (avg)	Finding and Diagnosis labels	Spatial Labels	Image Size
ChestX-ray8	108,948	32,717	3.3	8	0	resized (1024 × 1024)
ChestX-ray14	112,120	30,805	3.6	12	9	resized (1024 × 1024)
PLCO	185,421	56,071	3.3	14	0	original size
CheXpert	224,316	65,240	3.4	14	0	original size (reduced to 8-bits)
MIMIC-CXR	371,920	65,383	5.6	14	104	original size
PadChest	160,868	67,625	2.3	193	0	original size
VinDr-CXR	18,000	-	-	28	0	original size

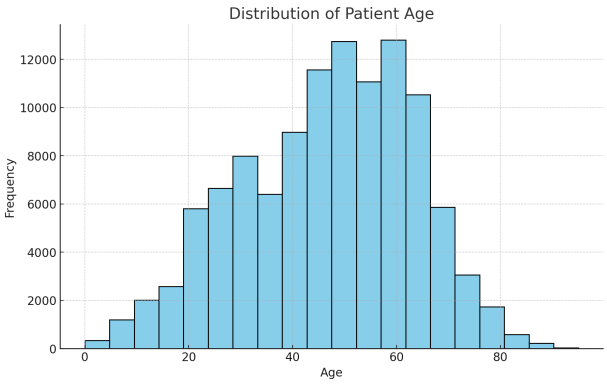
Table 4.1: A summarized comparison of some of the most utilized chest x-ray datasets.

Some other famous Chest X-ray datasets that exist in this regard are: PLCO [And12], CheXpert [Irv+19], MIMIC-CXR [Joh+19], PadChest [Bus+20], VinDr-CXR [Ngu+22].

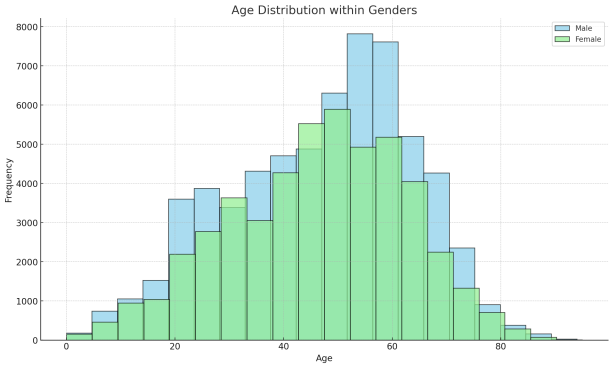
4.2 Preprocessing and Data Analysis

ChestX-ray14 dataset has 14 categories of disease labels and one label for no finding. Figure 4.1 show different plots explaining the statistical distribution of ChestX-ray14 dataset. Some of the major findings of the **data analysis** include the following observations:

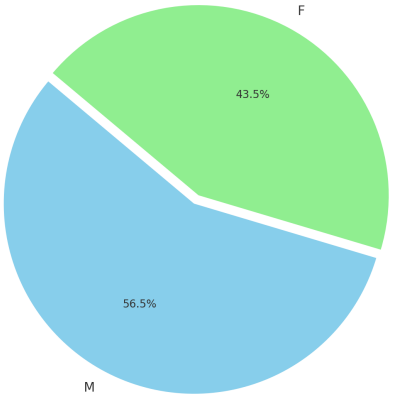
- Distribution of patient age (Figure 4.1a) - The distribution of patient ages varies widely, with a significant number of patients in the middle-aged category. Most patients are between 50 and 65 years old, indicating that medical imaging is more commonly needed for individuals in this age group.
- Age distribution within genders (Figure 4.1b) - Male and female patients display similar age distribution patterns, with most individuals concentrated in the middle-aged groups.
- Gender distribution (Figure 4.1c) - The dataset shows a slightly higher number of male patients than female patients, suggesting a minor gender imbalance.
- Disease prevalence by gender (Figure 4.1d) - Certain diseases are more common in one gender than the other. For example, conditions such as cardiomegaly in females and emphysema are more frequently diagnosed in males.
- Number of images per disease (Figure 4.1e) - The number of images per disease varies widely, with most images having no disease. This variation highlights the



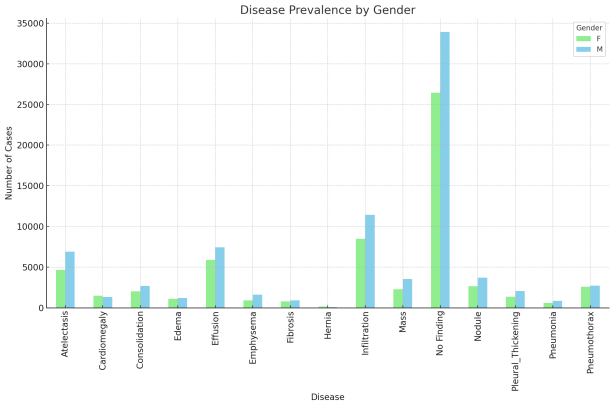
(a) Distribution of patient age



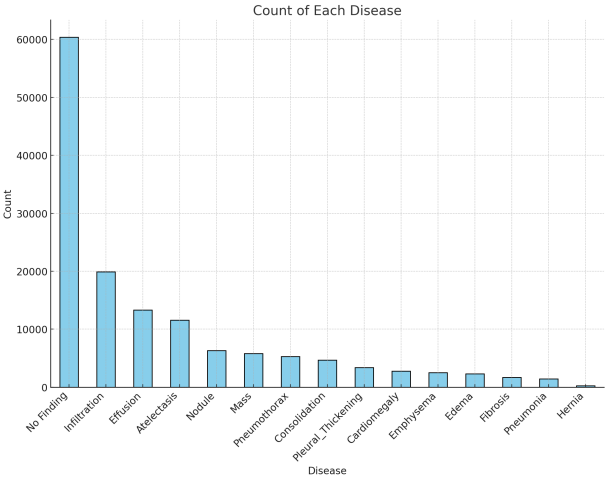
(b) Age distribution within genders



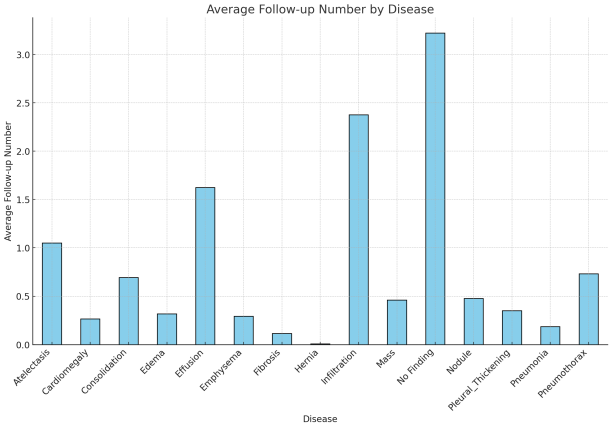
(c) Gender distribution



(d) Disease prevalence by gender

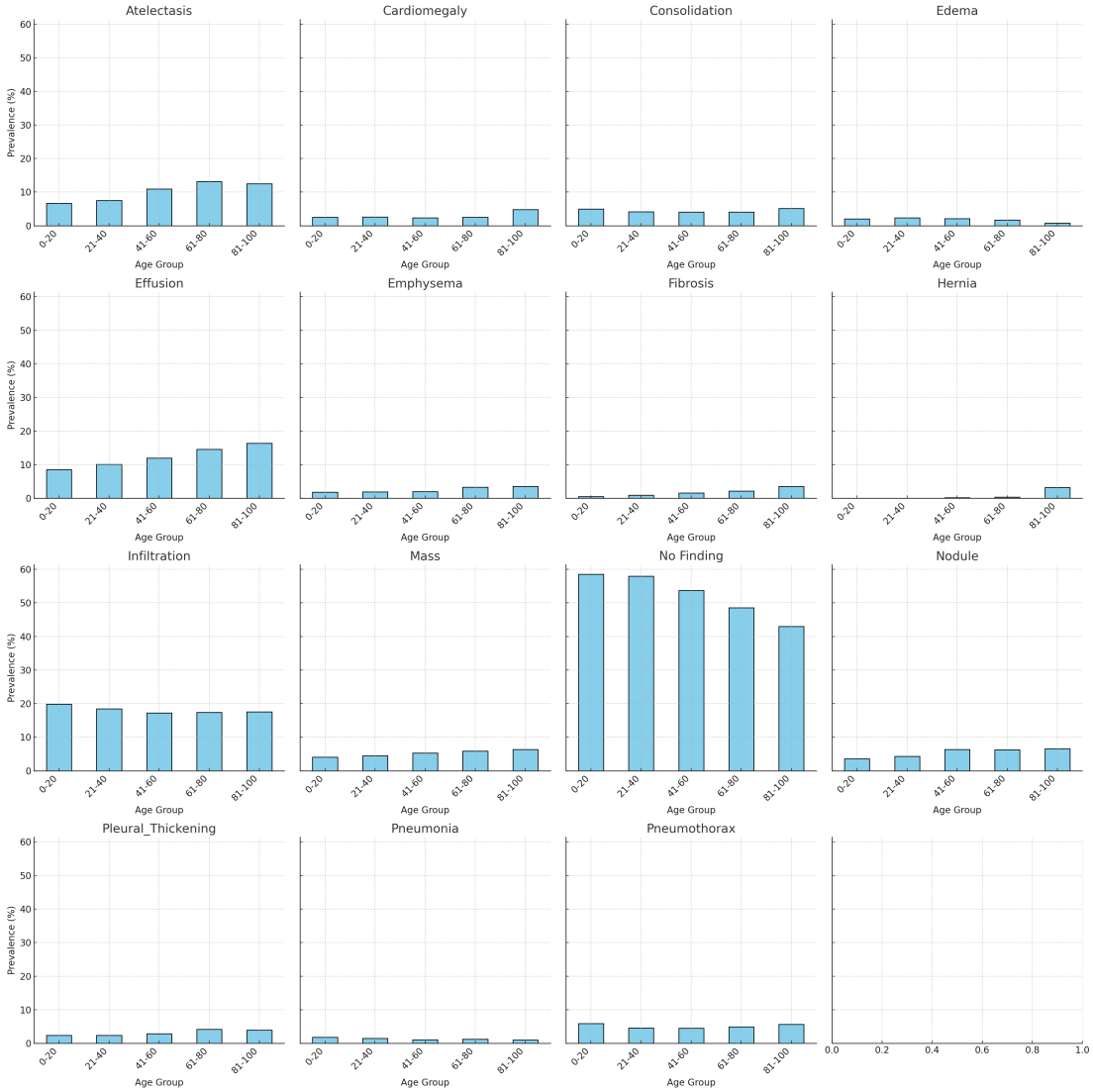


(e) Number of images per disease

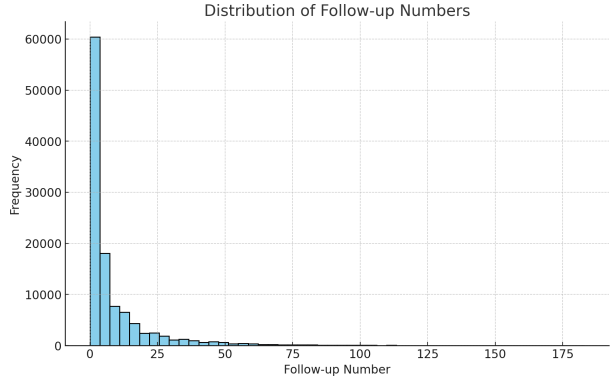


(f) Average follow-ups per disease

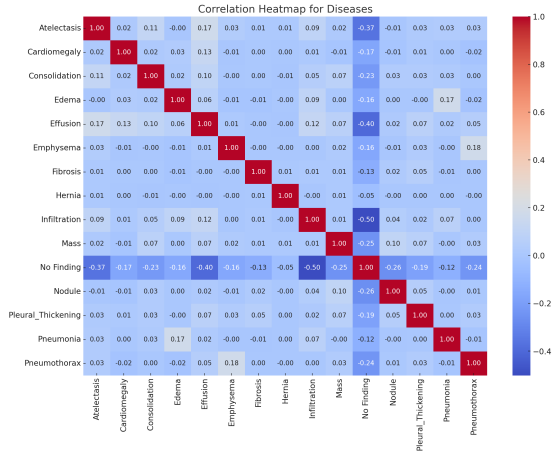
Figure 4.1: Statistical analysis of ChestX-ray14 dataset.



(g) Disease prevalence by Age Group



(h) Distribution of number of follow-ups



(i) Correlation of diseases in dataset

Figure 4.1: Statistical analysis of ChestX-ray14 dataset.

uneven distribution of imaging data across different diseases, possibly reflecting their incidence rates.

- Average follow-ups per disease (Figure 4.1f) - Diseases needing continuous monitoring, such as Infiltration, typically have a higher average number of follow-ups.
- Disease Prevalence by Age Group (Figure 4.1g) - Disease prevalence differs across age groups, with older individuals experiencing higher rates of conditions like cardiomegaly and emphysema. This trend highlights the increased health risks associated with aging.
- Distribution of the number of follow-ups (Figure 4.1h) - The number of follow-ups per patient is skewed, with many patients having few follow-ups and a smaller group needing numerous follow-ups. This suggests that while most conditions require limited follow-ups, a subset of patients needs extensive monitoring.
- Correlation of diseases in the dataset (Figure 4.1i) - Certain diseases show minor correlations. For instance, pneumothorax and emphysema commonly co-occur in the X-ray images.

During the data preparation step of PriSwin-Dis architecture, resizing and normalization are the two most significant data **preprocessing** techniques used. Code snippet 4.1 shows their implementation in PriSwin-Dis architecture.

- *Resizing* - Standardizing image sizes guarantees uniform input dimensions for the model, essential for consistent tensor operations and convolutional layers. This approach also lowers computational complexity and memory usage, enabling quicker and more efficient training by resizing 1024×1024 size images to 256×256.
- *Normalization* - Normalization stabilizes and accelerates the training process by adjusting the input data to possess a standard deviation of one and a mean of zero. This enhances the model's numerical stability and helps the optimization algorithm converge more quickly [LeC+02].

```
# Preprocessing function for the image
def preprocess_image(image_path):
    preprocess = transforms.Compose([
        transforms.ToTensor(),
        transforms.Resize((256, 256)),
```

```
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
                             std=[0.229, 0.224, 0.225])
    ])
    image_tensor = preprocess(image)
    return image_tensor
```

Listing 4.1: Pre-processing step in PriSwin-Dis architecture

4.2.1 Patient-wise Splitting Technique

For a more accurate and objective comparison, we use the patient-wise split gathered by [Pac+23b]. We split the data into three parts using a patient-focused approach: 20% is used for testing, 10% is for validation, and 70% is for training, i.e., 5,000 images for testing, 2,000 for validation, and 10,000 pairs of images are used for training. Each pair is balanced regarding positive (two distinct photographs from the same patient) and negative (two distinct images from different patients) samples.

4.3 Design of PriSwin-Net & PriSwin-Dis Anonymization Architecture

Our proposed PriSwin-Net & PriSwin-Dis anonymization architectures take their motivation from an existing state-of-the-art anonymization PriCheXy-Net [Pac+23b] architecture. The modifications in PriCheXy-Net architecture are motivated by the idea of improving the U-Net-generated noise in order to produce realistic-looking radiographic images. To achieve this goal and improve the privacy versus their utility trade-off after the anonymization of the images, we have proposed two architectures: *PriSwin-Net* and *PriSwin-Dis* architecture as shown in Figure 4.2 and Figure 4.3 respectively. PriSwin-Dis architecture is comprised of a U-Net generator (Section 4.3.1), a transformer-based auxiliary classifier (Section 4.3.2), an auxiliary discriminator to enforce realism into anonymized images (Section 4.3.3), and an auxiliary verification model (Section 4.3.4). PriSwin-Net architecture is identical to PriSwin-Dis, except that PriSwin-Net does not utilize an additional auxiliary discriminator network and the U-Net network is updated according to the losses of auxiliary classification and verification network.

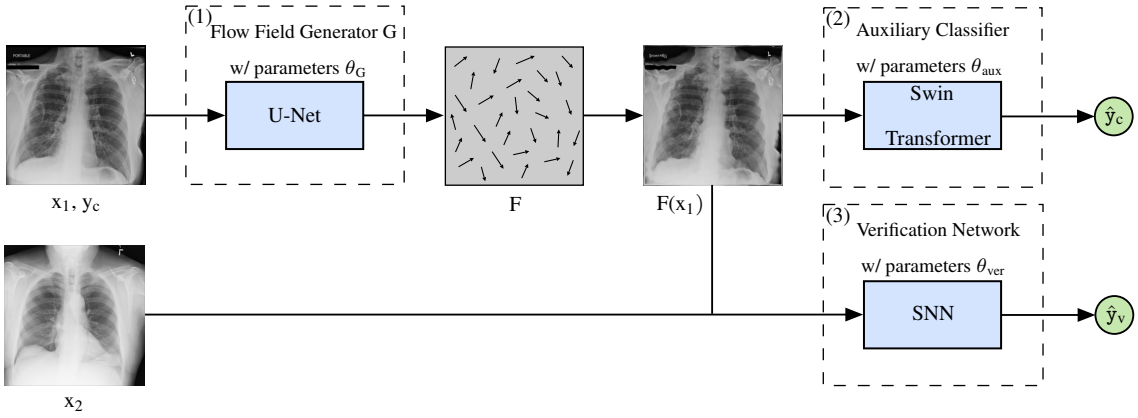


Figure 4.2: Proposed PriSwin-Net architecture.

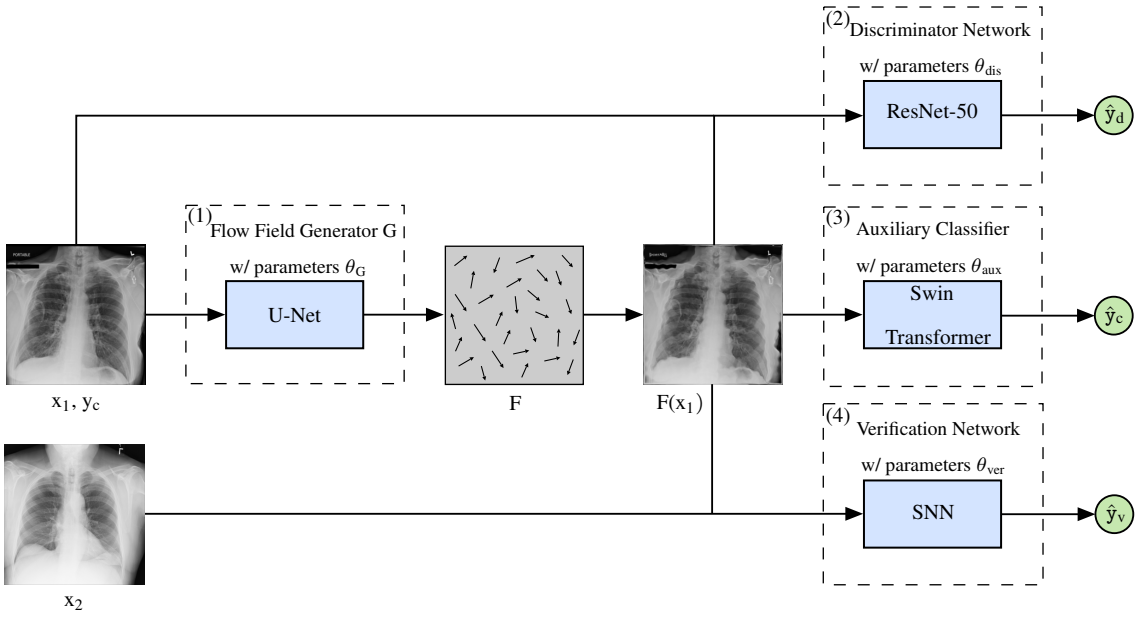


Figure 4.3: Proposed PriSwin-Dis architecture.

4.3.1 U-Net Generator

The U-Net [Ron+15] generator G is motivated from [Pac+23b] and implemented according to [Bud+19]. This network is implemented as the generator network in PriCheXy-Net as discussed in Section 3.4.3. The U-Net takes an X-ray image as input and transforms the image by applying flow fields to it. The anonymized image is then passed to the three auxiliary networks as input.

The objective loss $L(\theta_G, \theta_{dis}, \theta_{aux}, \theta_{ver})$ in PriSwin-Dis architecture is to be minimized according to Equation (4.1), where the loss from each auxiliary network (discriminator, classifier, and verification network) contributes to the total loss of the network.

$$\arg\min_{\theta_G} L(\theta_G, \theta_{dis}, \theta_{aux}, \theta_{ver}) = L(\theta_G, \theta_{dis}) + L(\theta_G, \theta_{aux}) + L(\theta_G, \theta_{ver}) \tag{4.1}$$

The PriSwin-Dis architecture aims to enable the U-Net based generator to learn targeted deformations with the help of three additional auxiliary networks. These additional auxiliary models guide the U-Net through its loss functions to control the deformations in order to preserve the image's utility and, simultaneously, deform the image enough to safeguard privacy. Each auxiliary model has a specific objective to learn, and their overall effect on U-Net is controlled by weighting these networks as shown in Equation (4.2).

$$\begin{aligned}\arg\min_{\theta_G} L(\theta_G, \theta_{dis}, \theta_{aux}, \theta_{ver}) = (w_{dis} \ * L(\theta_G, \theta_{dis})) + (w_{aux} \ * L(\theta_G, \theta_{aux})) \\ + (w_{ver} \ * L(\theta_G, \theta_{ver}))\end{aligned} \tag{4.2}$$

where w_{dis}, w_{aux}, and w_{ver} are the weights of the respective networks representing the contribution of each auxiliary model in the learning of U-Net generator.

PriSwin-Net architecture utilizes the same objective loss as PriCheXy-Net [Pac+23b] as shown in Equation (3.7).

4.3.2 Auxiliary Classifier

The original implementation of PriCheXy-Net [Pac+23b] has a Dense-Net [Hua+17] based auxiliary classifier motivated by an earlier implementation of CheXNet [Raj+17] classifier (Section 3.4.3). To overcome the limitations of the DenseNet-based classifier, we propose a larger capacity transformer-based auxiliary classifier model - Swin-T [Liu+21] (Section 2.4.3).

Taking the motivation from [Tas+22], the last layer of Swin-T is modified to accommodate 14 MLP heads (one for each disease) to classify 14 diseases present in the ChestX-ray14 [Wan+17] dataset. This modified pre-trained model of Swin-T trained on ChestX-ray14 [Wan+17] dataset with a mean AUC of 84.3% is used as a starting point in the training of auxiliary classification model in PriSwin-Dis.

In both PriSwin-Net and PriSwin-Dis, the Swin-T based auxiliary classifier takes an anonymized image $F(x_1)$ as input and tries to detect the correct disease and assign probabilities for each disease according to BCE optimizer and loss as mentioned in Equation (3.8). The goal of this auxiliary classifier is to maximize the utility of the anonymized images by controlling the deformations applied by the U-Net.

4.3.3 Auxiliary Discriminator

We employ an additional auxiliary discriminator network with the aim of enforcing realism into the anonymized images. Our proposed discriminator network is motivated by the works of [Pac+22] and is implemented as a ResNet-50 network used in SNN [Pac+22].

In order to train the ResNet-50 architecture to differentiate between an original and a deformed image, it is fed with two images: an unaltered image y^{real} and a corresponding anonymized image y^{fake}. The last layer of the network is adjusted to produce a single output value within the range of 0 to 1.

During the training of this auxiliary discriminator network, corresponding losses for the identification of real and anonymized images are calculated as in Equation (4.3) and Equation (4.4) respectively. The average of these losses together contributes towards the total loss L_{dis} of the discriminator network as shown in Equation (4.5). The total loss of the auxiliary discriminator network is backpropagated to the U-net generator.

$$L_{dis^{real}}\left(1, \theta_{dis^{real}}\right) = -\sum_{i=1}^{2}\left[y_{d,i}^{real}\ \log\left(\hat{y}_{d,i}\right) + \left(1 - y_{d,i}^{real}\right)\ \log\left(1 - \hat{y}_{d,i}\right)\right] \tag{4.3}$$

$$L_{dis^{fake}}\left(0, \theta_{dis^{fake}}\right) = -\sum_{i=1}^{2}\left[y_{d,i}^{fake}\ \log\left(\hat{y}_{d,i}\right) + \left(1 - y_{d,i}^{fake}\right)\ \log\left(1 - \hat{y}_{d,i}\right)\right] \tag{4.4}$$

$$L_{dis}\left(\theta_G, \theta_{dis}\right) = \frac{L_{dis^{real}}\left(1, \theta_{dis^{real}}\right) + L_{dis^{fake}}\left(0, \theta_{dis^{fake}}\right)}{2} \tag{4.5}$$

4.3.4 Auxiliary Verification Model

A SNN network is used as an auxiliary verification network in our proposed PriSwin-Dis architecture. This SNN based auxiliary verification network is also motivated from [Pac+23b] and implemented according to [Pac+22] as discussed in Section 3.4.3. The primary function of this auxiliary verification network is to enforce privacy such that SNN network fails to identify images from the same patient; this is done by introducing more deformations through the U-Net.

The input to the auxiliary verification network in PriSwin-Net and PriSwin-Dis is a deformed image $F(x_1)$ received from U-Net after anonymization of the original image x_1. The SNN network performs a linkage attack and tries to identify the original image corresponding to the anonymized image. Based on the similarity score $\hat{y} \in [0, 1]$ for different or same patient images, the auxiliary verification loss $L_{ver}\left(\theta_G, \theta_{ver}\right)$ of this network in both the architectures, BCE loss, is calculated according to Equation (3.9). This loss finally contributes to the total loss of the network and is propagated backward to optimize the U-Net.

Therefore, the PriSwin-Dis architecture combines four different networks that work together in a harmonized manner to enable the anonymization of medical chest radiographs with high utility and privacy. The four networks are primarily categorized into two types: the generator and the auxiliary networks. The three auxiliary networks guide the generator in generating targeted deformation in the images. Generator and auxiliary networks compete in a Min-Max game where each auxiliary network has a specific effect on the generator deformations. The auxiliary discriminator enforces realism, the auxiliary classifier preserves utility, and the auxiliary verification network tries to preserve the privacy of the image.

Chapter 5

Experimental Setup

5.1 Computational Graph Setups

In DL, computational graphs describe the flow of data through a model and the operations applied, such as addition, multiplication, or more complex neural network functions. These graphs represent how data is processed in a neural network by using *nodes* (representing weights and biases) and *edges* (tensors) [Col18]. Basically, it outlines the sequence of computations needed to convert input data into output data. A computational graph helps during the backpropagation of the network by computing gradients to update model parameters during training.

Our model PriSwin-Dis has in total four computational graphs (see Figure 5.1). This multi-graph setup enables a robust and flexible training process, where the generator learns to create data that is both private and has high utility. Meanwhile, the three auxiliary models fine-tune their specific capabilities. This setup creates an environment in which the generator is continually challenged and improved, resulting in the generation of high-utility, privacy-preserving data.

Each model within the PriSwin-Dis architecture has its own computational graph and performs different optimizations, as discussed below:

(1) U-net Generator Computational Graph (C_1)

- Nodes: Input data, convolutional layers, activation functions, and the output layer.
- Edges: Data flows through the layers, undergoing transformations such as convolutions and non-linear activations.

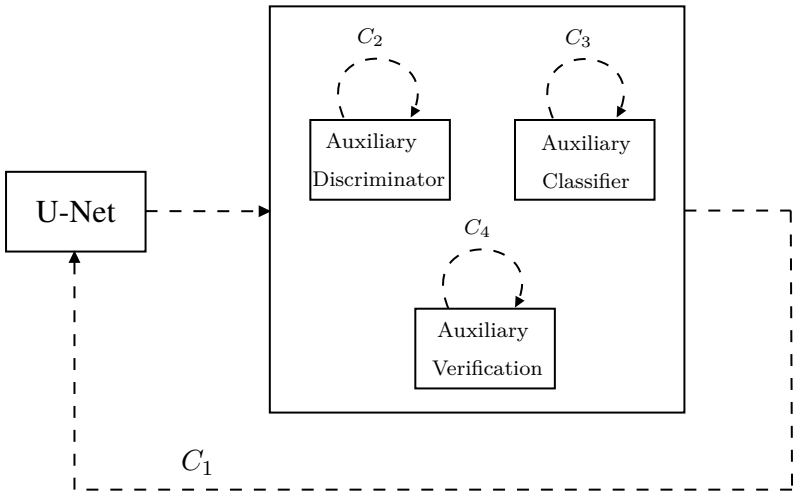


Figure 5.1: Computational graphs in PriSwin-Dis, C_1, C_2, C_3, and C_4 represent the optimization of the PriSwin-Dis architecture, Auxiliary Discriminator, classifier, and verification model, respectively.

- Loss Calculation: The generator's loss is a combination of the losses from the three auxiliary models, ensuring that the generated data meets multiple criteria simultaneously.

(2) Auxiliary Discriminator Computational Graph (C_2)

- Nodes: Input data (both real and generated), convolutional layers, activation functions, and the output layer.
- Edges: Data flows through the layers, and the discriminator outputs a probability score indicating whether the data is real or generated.
- Loss Calculation: Standard adversarial loss that penalizes incorrect classifications of real vs. generated data.
- Optimization: The discriminator is optimized independently to improve its discriminative power.

(3) Auxiliary Classifier Computational Graph (C_3)

- Nodes: Input data, convolutional layers, activation functions, and the output layer for class labels.
- Edges: Data flows through the layers, resulting in the classification of the input data.

- Loss Calculation: Cross-entropy loss, which measures the accuracy of the classification.
- Optimization: The classifier is trained to enhance its ability to classify the generated data correctly.

(4) Auxiliary Verification Model Computational Graph (C_4)

- Nodes: Input data, various layers as per the model design, and the output layer for the verification score.
- Edges: Data flows through the layers to produce a verification score.
- Loss Calculation: Verification loss, designed to check the re-identification of the generated data.
- Optimization: The verification model is optimized to refine its verification capabilities.

5.2 Experimental Platform, Frameworks, and Libraries

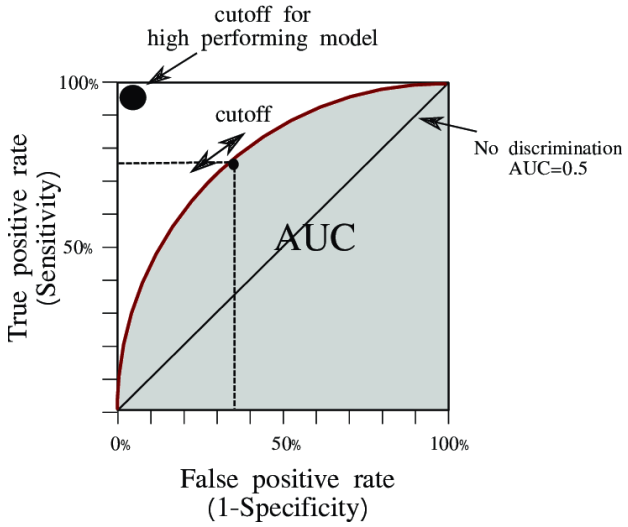
All of our experiments are performed on the servers of Erlangen National High Performance Computing Center (NHR@FAU)[1]. We utilized majorly three Graphical Processing Units (GPUs) for our experiments: Nvidia RTX 2080 Ti (11 GB), Nvidia Geforce RTX3080 (10GB), and Nvidia A100 (40GB) provided by TinyGPU service of NHR@FAU.

For our experiments with PriSwin-Dis, we used *Python* 3.12.2 with the following major libraries:

```
numpy          v1.26.4          torchvision     v0.18.1
pandas         v2.2.2           tensorboard     v2.17.0
matplotlib     v3.9.0           timm            v1.0.7
pillow         v10.3.0          scipy           v1.13.7
scikit-learn   v1.5.0           nvidia-cuda     v12.1.105
torch          v2.3.1
```

Parts of our deep-learning experimental architecture are implemented with the help of predefined DL features of PyTorch [Pas+19]. For visualizing the intermediate state of the architecture's learning, Tensorboard [Won+17] is used.

[1] NHR@FAU (https://hpc.fau.de/)



(a)

		PREDICT CLASS	
		Positives	Negatives
TRUE CLASS	Positives	True Positive **(TP)**	False Positive **(FP)**
	Negatives	False Negative **(FN)**	True Negative **(TN)**

(b)

Figure 5.2: Evaluation Methods: (a) Area Under the Curve (AUC) [Rod+21] and (b) Confusion Matrix [Cae17].

5.3 Evaluation Methods & Techniques

In this section, we will discuss two of the most important evaluation metrics used to measure the performance of our PriSwin-Net and PriSwin-Dis architectures. We will also explain how the evaluation pipeline works to measure privacy and utility in quantifiable terms.

5.3.1 Evaluation Metrics

Area Under the Curve (AUC)

The AUC [Hos+15] is a commonly employed measure in classification tasks that quantifies the classifier's capacity to differentiate between classes. It plots the number of actual positives that are accurately identified (Sensitivity) by the model on the y-axis against the number of actual negatives that are wrongly identified as positives (Specificity) by the model on the x-axis (see Figure 5.2a). An AUC value of 1.0 indicates the best classification score (denoted by a black circle in Figure 5.2a), whereas an AUC value of 0.5 indicates accuracy equivalent to random guessing.

Confusion Matrix

Confusion Matrix [Cae17] (see Figure 5.2b) is used for detailed performance analysis of our classification and verification model. It shows the counts of:

(1) True Positives: Instances where the model correctly identified the class of positives.

(2) True Negatives: Instances where the model correctly identified the class of negatives.

(3) False Positives: Instances where the model incorrectly identified cases of the positive class as negative cases.

(4) False Negatives: Instances where the model incorrectly identified cases of the negative class as positive cases.

This detailed breakdown helps us identify specific patterns of errors caused by the anonymization process. By examining the confusion matrix, we can pinpoint areas where the classifier may be misclassifying diseases. In the case of verification, it tells us the number of patients reidentified successfully. This allows us to make targeted improvements to ensure that the anonymized images remain diagnostically useful and keep privacy high.

5.3.2 Evaluation of Anonymization Architecture

After training our PriSwin-Dis architecture, we use the evaluation technique similar to that of used by [Pac+23b] to evaluate how well the U-net generator anonymizes chest X-ray images while preserving essential medical information. This evaluation process involves two main assessments: classification score and verification score. These evaluations help us understand the balance between maintaining data utility and ensuring effective anonymization.

Classification Score: Preserving Disease Information

To determine whether the anonymized images still contain disease information, we calculate the classification score. This score indicates whether the disease information in the chest X-rays is preserved after anonymization. Here's how we approach this:

- Anonymizing Test Images: We use the trained PriSwin-Dis to anonymize the chest X-ray images in our test set. These anonymized images are expected to obscure identifiable information while retaining important medical details.

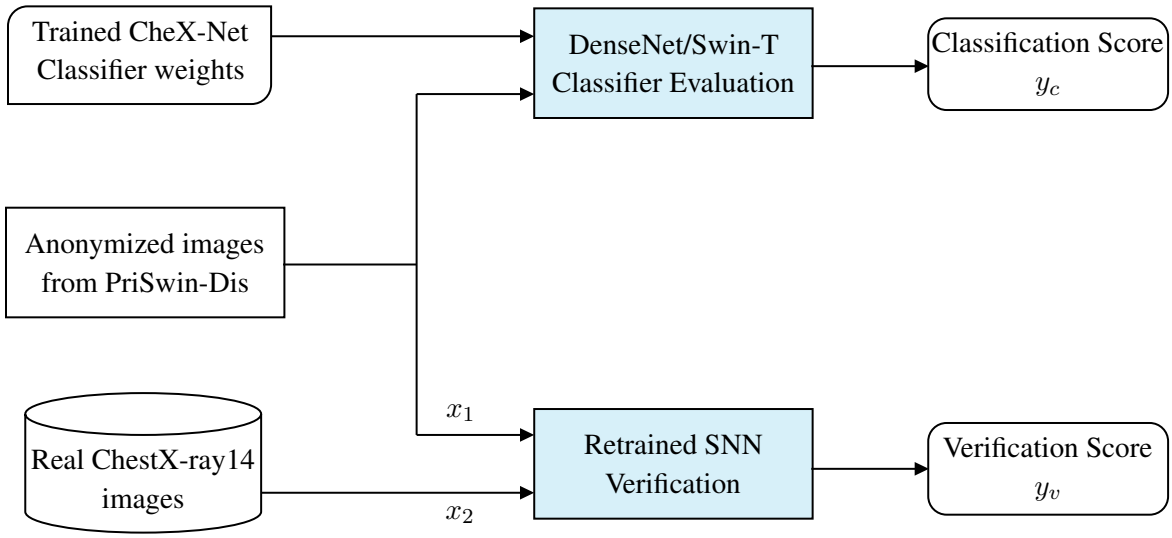


Figure 5.3: PriSwin-Dis evaluation architecture consists of two evaluation techniques: Classification and verification. x_1 (anonymized) and x_2 (original) are the two input images being used for the evaluation.

- Evaluating with Trained Classifier: We utilize a trained classifier specifically trained to detect and classify abnormalities in chest X-rays. This classifier is applied to the anonymized images to identify the occurrence of various diseases. For a fair comparison with PriCheXy-Net [Pac+23b], we use the DenseNet-based classifier used in [Pac+23b]. We have also compared evaluations with stronger classifiers like Swin-T in Chapter 6.

- Calculating AUC Values: The classifier's performance on the anonymized images is measured using the AUC metric for each of the 14 disease classes. We compute the average of the 14 class-specific AUC results to get an overall sense of how well the disease information is preserved in the anonymized images across all diseases.

By evaluating the classification score, we ensure that the anonymization process does not significantly degrade the utility of the Chest X-ray images. A high classification score indicates high utility, i.e., that the anonymized images still contain relevant disease information, making them useful for clinical and research purposes.

Verification Score: Ensuring Effective Anonymization

The verification score assesses whether the anonymized images are sufficiently perturbed to prevent re-identification by any identification network. Here's the step-by-step process:

- Applying re-identification network: We use SNN from [Pac+23b] as a re-identification network to match the anonymized images with their original counterparts. This network is trained on anonymized images to simulate a real-world scenario where a malicious user could use such an identification network to perform a linkage attack to recognize and verify images based on identifiable features.

- Calculating Verification Score: The verification score reflects the network's ability to identify the patient to which the anonymized image belongs correctly. A lower verification score means the identification network struggles to match anonymized images to their original versions, indicating effective anonymization.

By evaluating the verification score, we ensure the anonymization process is robust enough to protect patient privacy. The goal is to make the anonymized images indistinguishable from the original ones in terms of identifiable features.

By combining these two evaluation metrics, classification and verification, we comprehensively assess the performance of our anonymization architecture. This dual evaluation ensures that we achieve a balance between preserving the utility of medical data and protecting patient privacy. Our goal is to develop a system that can effectively anonymize chest X-ray images without compromising the valuable medical information they contain.

5.4 Experimental Configurations

In an attempt to improve the performance of PriCheXy-Net (Section 3.4.3), we experimented with two modifications and therefore proposed the following two architectures: PriSwin-Net (Section 5.4.1) and PriSwin-Dis (Section 5.4.2).

5.4.1 PriSwin-Net

PriSwin-Net represents a modified architecture of the PriCheXy-Net [Pac+23b] model (Section 3.4.3). In this adaptation, the original DenseNet-based classifier, ChexNet developed by [Raj+17], has been substituted with a more robust and powerful classifier, such as Swin-T based SwinCheX [Tas+22] (see Figure 4.2). This advanced classifier is a transformer network developed by Microsoft and has been incorporated as an auxiliary classifier, offering enhanced performance and accuracy in the classification task.

5.4.2 PriSwin-Dis

The PriSwin-Dis architecture is an improved iteration of the above PriSwin-Net architecture. PriSwin-Dis includes an additional auxiliary discriminator model to enforce realism in the images. With this modified architecture we aim to generate more realistic anonymized images with greater utility. PriSwin-Dis architecture (Figure 4.3) is discussed in detail in Section 4.3.

5.5 Hyper-parameter Tuning

In the experiments conducted, various hyperparameters were tuned to optimize the model's performance. The following parameters were carefully adjusted and tested during the training of both PriSwin-Net and PriSwin-Dis architectures:

- *Learning Rate* - Both local and global learning rates were experimented with, specifically values of 10^{-4} and 10^{-5}. These values were chosen to ensure the model was fine-tuned without significant oscillations in the loss function.
- *Batch Size* - Different batch sizes were tested, including 8, 16, 32, and 64. This allowed for balancing between the computational efficiency and the stability of the gradient updates. Given the substantial amount of ChestX-ray images in the dataset and large deep networks, a smaller batch size with lower memory GPUs is recommended.
- μ - The anonymization degree parameter μ was varied in the range of 0.5 to 0.001 to study the effect of anonymization degree on the network learning and types of anonymized images produced.
- *Number of epochs* - The number of training epochs depended on the GPU used. Generally, 80 epochs were sufficient for lower memory GPUs, while some experiments utilized even around 250 epochs for GPUs with higher memory capacity, like the NVIDIA A100.
- *GPU* - The experiments utilized different GPUs, including NVIDIA 2080, 3080, and A100, to determine the impact of computational power and memory on the training process.
- *Patience* - A patience parameter of 5 epochs was set for early stopping, allowing the model to halt training if no improvement was observed over five consecutive epochs.

- *Optimizer* - Various optimizers were tested, including Adam, AdamW, and SGD, to identify the most effective optimization algorithm for the given task.
- *Auxiliary Model Weights* - The weights for the auxiliary models were initially set to a 1:1:1 ratio. Different ratios were also experimented with to find the most effective balance for the auxiliary losses.

Through systematic tuning of these hyperparameters, the model's performance is optimized, which leads to improved accuracy and efficiency in the final results.

Chapter 6

Experiments and Results

In this chapter, we will discuss different types of networks created within this study's scope. We will interpret the outcome of changing the auxiliary classifier and its effect on anonymization performance. We will also show an extensive series of experiments that were performed with the objective of enhancing the anonymization efficiency without hurting the utility too much. Our discussion will include the crucial findings from our results of different experiments. In the later topics of this chapter, these findings are also corroborated by the instigations from lower-dimensional analysis of the internal states of the verification model. Finally, we will see the impact of our modifications and experiments on the privacy-utility trade-off of our anonymization approach.

6.1 Comparative Analysis of Other Models

In our experiments, we compared two auxiliary classifiers, CheXNet [Raj+17] (based on DenseNet) and SwinCheX [Tas+22] (based on Swin-T), to evaluate their performance and utility when integrated into our proposed architecture PriSwin-Dis for handling anonymized images.

6.1.1 ChexNet vs. SwinCheX

One of the primary goals of this thesis is to enhance the utility of anonymized clinical images by improving the accuracy and robustness of disease classification networks. We compared two auxiliary classifiers, CheXNet (based on DenseNet) and SwinCheX (based on Swin Transformer), to identify which network could better support the anonymization architecture in classifying 14 different diseases. CheXNet, a well-established model, has

shown strong performance in medical imaging tasks, while the Swin Transformer, a more recent innovation, promises superior performance due to its advanced architecture.

Pathology	CheXNet	SwinCheX
Atelectasis	0.7677	**0.8256**
Cardiomegaly	0.8740	**0.9096**
Consolidation	0.7451	**0.8145**
Edema	0.8395	**0.8914**
Effusion	0.8254	**0.8825**
Emphysema	0.9023	**0.9158**
Fibrosis	0.8220	**0.8364**
Hernia	0.8816	**0.9422**
Infiltration	0.7002	**0.7179**
Mass	0.8204	**0.8581**
Nodule	0.7593	**0.7866**
Pleural Thickening	0.7683	**0.7802**
Pneumonia	0.7112	**0.7632**
Pneumothorax	0.8527	**0.8769**
Average AUC	0.8050	**0.8430**

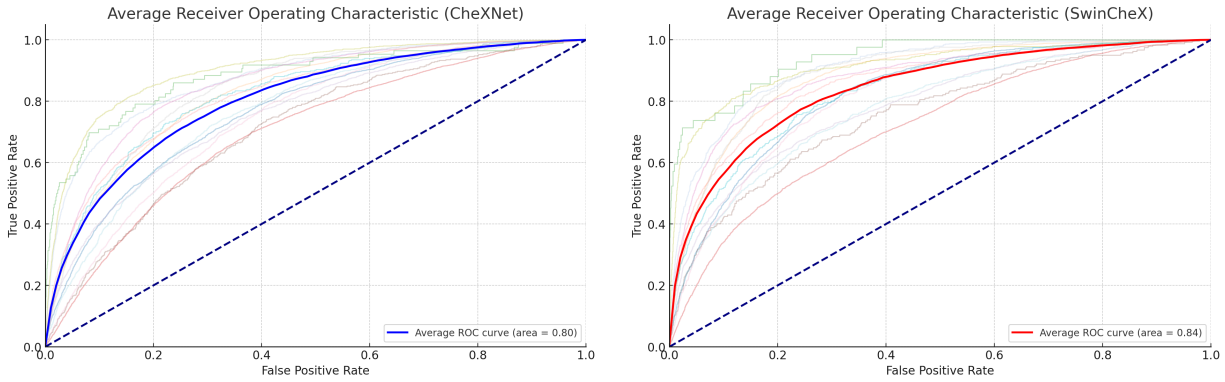
Table 6.1: Comparison of CheXNet & SwinCheX classifier's AUC score where SwinCheX outperforms CheXNet in all the disease categories.

To adapt these networks to the specific needs, both the networks were modified and trained as follows:

- CheXNet (DenseNet): PriCheXy-Net [Pac+23b] utilized the CheXNet model as described by [Raj+17] by training it on ChestX-ray14 [Wan+17] dataset and used it as auxiliary classifier in PriCheXy-Net to optimize it for the classification of 14 diseases.

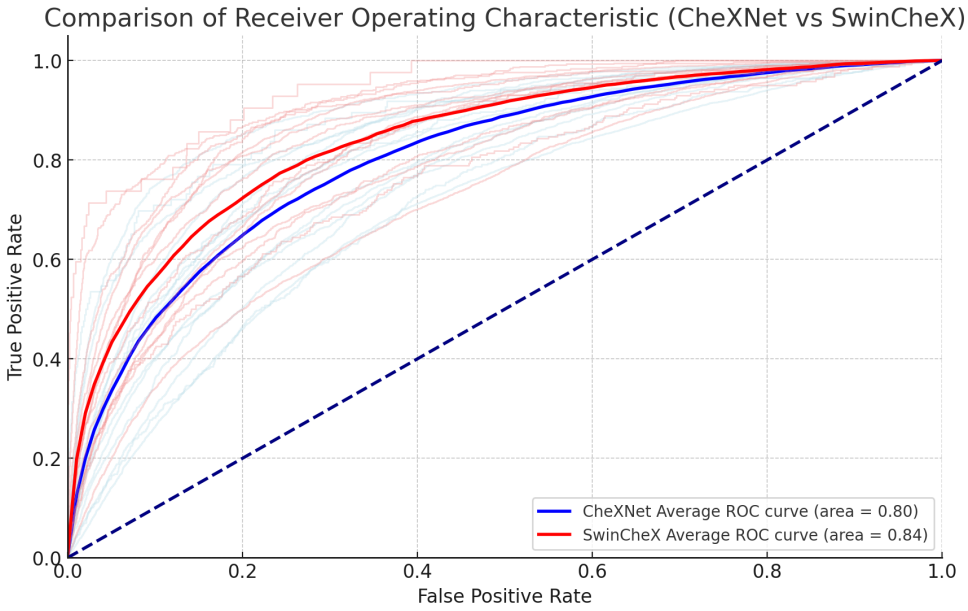
- SwinCheX (Swin Transformer): Built upon the Swin Transformer architecture developed by Microsoft [Liu+21]. We trained SwinCheX from scratch to ensure it could learn features specific to ChestX-ray14 [Wan+17] dataset. We modified the final layer of the Swin Transformer to include 14 Multi Layer Perceptron (MLP) heads, allowing the network to handle multi-label classification tasks for the 14 diseases.

These modifications in both architectures resulted in the performance as follows:



(a)

(b)



(c)

Figure 6.1: Area Under the ROC Curve (AUROC) performances of CheXNet and SwinCheX classifiers. (a) AUROC curve of CheXNet classifier, where the blue line represents the average AUC of all the diseases and lines with light colors represent the individual diseases. (b) AUROC curve of SwinCheX classifier, where the red line represents the average AUC of all the diseases and lines with light colors represent the individual diseases. (c) Comparison of AUROC curves of CheXNet and SwinCheX classifiers, where the blue and red lines represent the average AUC of all the diseases in CheXNet and SwinCheX respectively, and lines with light red color and light blue color represent the individual diseases of CheXNet and SwinCheX.

- CheXNet (DenseNet): [Pac+23b] achieved an AUC of 80.5% on real data (see Figure 6.1a) with CheXNet classifier. Demonstrated robust performance but had limitations in capturing long-range dependencies and contextual information. For a relative comparison of PriChXy-Net and PriSwin-Dis, we use this pre-trained classifier as an evaluation network for utility measurement.

- SwinCheX (Swin Transformer): We achieved an AUC of 84.3% on real data (see Figure 6.1b) with SwinCheX classifier. The hierarchical design and shifted windows of the Swin-T enabled it to capture both local and global contexts within images, resulting in better classification accuracy.

The results of this comparative study between CheXNet and SwinCheX help in drawing the following significant inferences:

- *Performance Improvement*: SwinCheX outperformed CheXNet by achieving a higher AUC score in almost all of the 14 diseases (see Table 6.1), demonstrating a strong classifier and its superior ability to distinguish between different diseases.

- *Architectural Advantages*: The Swin Transformer's ability to capture extensive contextual information and long-range dependencies gave it a clear edge over the DenseNet-based CheXNet.

- *Utility in PriSwin-Dis*: The improvement in AUC from 80.5% to 84.3% (see Figure 6.1) indicates a more reliable and accurate classification, crucial for effective medical diagnosis. By integrating this better-performing SwinCheX classifier as the auxiliary classifier in PriCheXy-Net [Pac+23b] to create PriSwin-Dis architecture, we aim to enhance the model's overall utility in dealing with anonymized medical images.

In conclusion, the idea of transition from CheXNet to SwinCheX[1] as the auxiliary classifier in PriCheXy-Net [Pac+23b] serves as a strong ground for effective anonymization theory. The better classification performance of SwinCheX validates the initial idea of exchanging the auxiliary classifier for a better classifier. The advancements brought by the Swin Transformer architecture make it a more suitable choice for complex multi-label classification tasks in medical imaging.

[1]SwinCheX Classifier Code(https://github.com/rajaatreja/SwinCheX_Classifier).

6.2 Comparisons of PriCheXy-Net and Modified Approaches

As discussed in Section 5.4, we modified PriCheXy-Net [Pac+23b] and proposed two configurations of the modified architecture: PriSwin-Net and PriSwin-Dis. The experiments with these architectures and their results are discussed in this section. In most of our experiments with our proposed anonymization architecture, we kept the deformation degree $\mu = 0.01$, at which [Pac+23b] showed the best performance in terms of privacy and utility. This approach aims to enhance the anonymization performance beyond the current best level.

6.2.1 PriCheXy-Net vs. PriSwin-Net

This comparison between PriCheXy-Net [Pac+23b] and PriSwin-Net aims to identify which anonymization model delivers superior performance and accuracy, thereby enhancing the diagnostic **utility** of anonymized images.

PriCheXy-Net Re-evaluation

To establish a robust baseline for our experiments, we reran PriCheXy-Net from end to end, including the pre-training of the U-Net generator. This comprehensive approach ensured an accurate verification of the model's performance in anonymizing medical images. Utilizing the NVIDIA A100 GPU with 40GB of memory, the training process for PriCheXy-Net was conducted over 250 epochs, taking approximately 10 hours and 30 minutes.

The training (Figure 6.2a) and validation loss (Figure 6.2b) curves for PriCheXy-Net [Pac+23b] displayed an alternating spiking pattern, which generally characterizes the minimax game learning architecture typical in adversarial networks. This spiking indicated the dynamic interplay between the generator and the auxiliary networks, each attempting to optimize against the other. Upon evaluation, PriCheXy-Net [Pac+23b] achieved a classification performance of AUC of 76.2% on CheXNet pre-trained classifier and a verification score of $57.7 \pm 4.0\%$. These results highlighted the model's capacity to anonymize and classify medical images effectively, providing a solid foundation for subsequent comparisons with our proposed architectures.

Finally, the visual difference between the anonymized images from different anonymization architectures can be seen in Figure 6.3 despite having a quantifiable difference in the

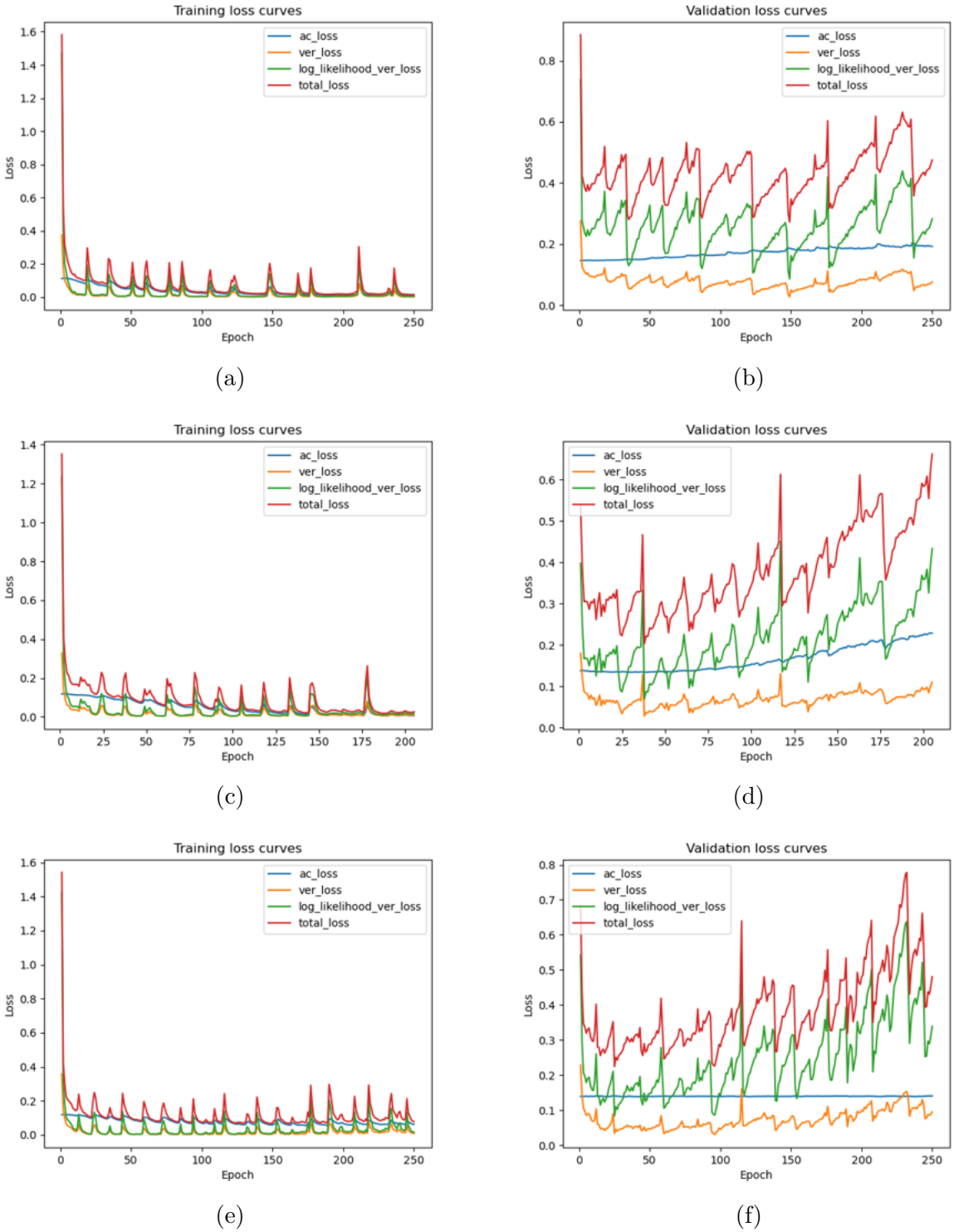


Figure 6.2: Training and validation loss curves of PriCheXy [Pac+23b] and PriSwin-Net. (a)-(b): PriCheXy-Net, (c)-(d): PriSwin-Net, (e)-(f): PriSwin-Net where the auxiliary classifier is not optimized.

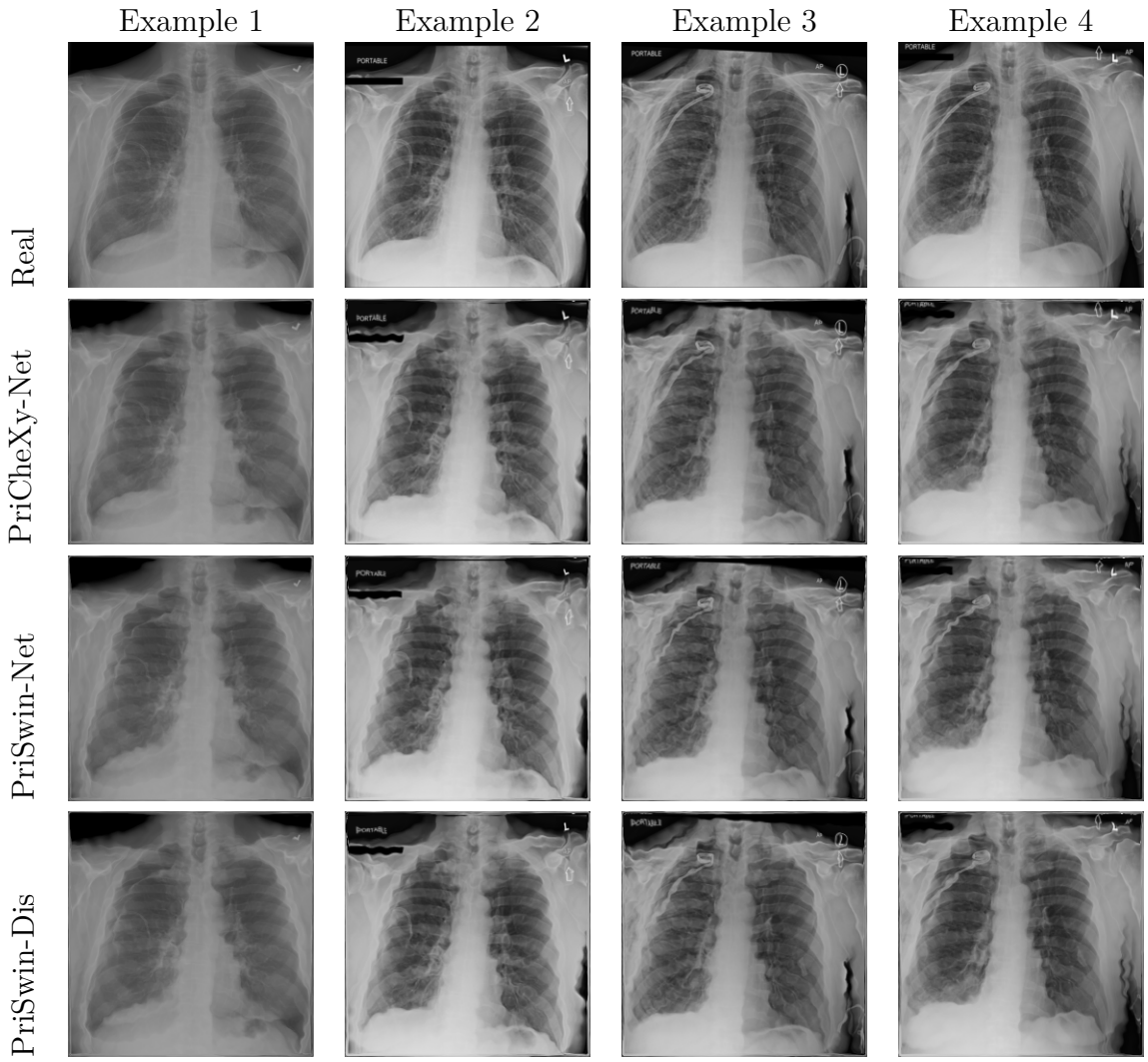


Figure 6.3: Examples of images showing the results of different anonymization techniques at deformation degree $\mu = 0.01$. The first row shows the real images from the ChestX-ray14 [Wan+17] dataset, the second row has the images anonymized by PriCheXy-Net [Pac+23b], the third row - PriSwin-Net anonymized images, and the fourth row - PriSwin-Dis anonymized images with our best performing model.

	Aux. Class. weight	Aux. Ver. weight	Learning Rate	Aux. Class. optimized	Classification Score (↑) DenseNet	Classification Score (↑) Swin-T	Verification Score (↓)
Real Data	-	-	-	-	80.5	84.3	81.8 ± 0.6
PriCheXy-Net	1	1	10^{-4}	✓	76.2	-	57.7 ± 4.0
PriSwin-Net	1	1	10^{-4}	✓	75.4	83.2	62.9 ± 4.9
PriSwin-Net	1	1	10^{-4}	×	77.0	-	66.6 ± 4.0

Table 6.2: Results of PriCheXy [Pac+23b] and PriSwin-Net. PriSwin-Net has two notable configurations: one where the auxiliary classifier is optimized during the architecture training and one without. Classification evaluations have been performed on two classification models: CheXNet and Swin-T. The higher the classification score, the better the utility; the lower the verification score, the better the privacy. All the experiments have $\mu = 0.01$. For the verification score, SNN was retrained 10 times independently, and the average of all scores was recorded.

actual performance of these methods when evaluated on the privacy-utility scale. To the human eye, the visual difference between the images anonymized by different techniques is hard to find. However, on closer inspection, in almost all of the examples shown in Figure 6.3, the following observations are inferred:

- PriSwin-Net tried to distort the boundary around the lungs. This justifies the lower classification performance when a DenseNet-based classifier evaluates the anonymized images. In PriCheXy-Net [Pac+23b], the boundaries of the lungs are preserved during the anonymization process, making it easier for DenseNet to classify the diseases. However, CheXNet does not perform well in PriSwin-Net due to these distortions, which do not affect stronger classifiers like transformer-based SwinCheX.

- Dropping of classification performance on distorting lung boundaries also confirms that the most important areas in chest X-rays are the regions around the heart and lungs, as also found by [Pac+22].

- Due to the effect of a stronger auxiliary classifier (SwinCheX), PriSwin-Net tries to keep the region around the heart sharp and ribs less distorted in comparison to PriCheXy-Net [Pac+23b]. This lower distortion around the heart and inside lungs enables the retrained SNN network to successfully link the anonymized image to the real image, resulting in a higher (worse) verification score.

PriSwin-Net (Frozen Auxiliary Classifier)

With this experiment, we verified our inference of the domain shift issue in the DenseNet-based pre-trained classifier during the evaluation of PriSwin-Net where Swin-T is used as an auxiliary classifier. Domain shift problem usually occurs in DL networks where the network is supplied with images different from the images the network is trained upon [Fon+20]. Therefore, when DenseNet takes images anonymized with PriSwin-Net (transformer-influenced images) to classify, it suffers from the problem of domain shift.

Hence, to tackle this issue of domain shift, the PriSwin-Net architecture is modified. We froze the auxiliary classifier and did not optimize it during the architecture training. This is done to prevent the auxiliary classifier from learning anonymized image features and adapting to classify diseases in anonymized images.

This architecture was also trained on the NVIDIA A100 GPU with 40GB of memory for 24 hours. The model trained with a global learning rate of 10^{-4} until 250 epochs and generated the training and validation loss curves as shown in Figure 6.2e and Figure 6.2f respectively. Unlike the classification loss in the validation loss curve shown in Figure 6.2d, the classification loss in the validation loss curve remains almost constant when the optimization of the auxiliary classifier is frozen, as shown in Figure 6.2f.

When comparing this PriSwin-Net configuration to PriCheXy-Net [Pac+23b], it was found that PriSwin-Net (where the auxiliary classifier is not optimized) achieved a higher AUC score when classifying anonymized images with the same pre-trained DenseNet classifier. The classification score with this configuration was 77.0%, and the verification score was 66.6% with a standard deviation of 4.0%, as illustrated in Table 6.2.

This experiment's better classification performance with a frozen auxiliary classifier helped to explain the domain shift problem that occurs while updating the auxiliary classifier during the architecture's training. When the auxiliary classifier is not optimized with anonymized images during the training, it continues to classify images with lower loss where the deformed image is close to the real image, hence controlling the deformations. As a result, the U-net generator learns to produce less deformed images that are closer to the real images; therefore, the DenseNet-based classifier yields a better classification score for images anonymized with PriSwin-Net anonymization architecture and does not face the domain shift issue.

On the other hand, when the auxiliary classifier is frozen, the verification performance of the anonymization architecture also degrades as the SNN is able to successfully link these anonymized images (with slightly fewer deformations) with their respective real images,

leading to a higher verification score. Therefore, it is important not to freeze the auxiliary classifier and to produce more effective anonymized images. In the next section Section 6.2.2, we show the effects of an additional auxiliary network whose aim is to enforce realism in the images to reduce the domain shift effect and produce realistic-looking anonymized images.

In conclusion, the comparison between PriCheXy-Net [Pac+23b] and PriSwin-Net[2] demonstrated that the latter offered improvements in classification accuracy and verification stability when evaluated by a Dense-Net based classification network. The use of the Swin Transformer in PriSwin-Net provides a more robust framework for handling anonymized medical images, thereby enhancing the overall utility and effectiveness of the model for diagnostic purposes. This comprehensive evaluation underscores the benefits of employing advanced transformer-based models for complex medical imaging tasks.

6.2.2 PriCheXy-Net vs. PriSwin-Dis

As previously discussed, the incorporation of a transformer-based auxiliary classifier in PriSwin-Net (Section 6.1.1) yielded notable insights into the architecture's dynamics. Our attempts to enhance realism in anonymized images by freezing the auxiliary classifier did result in the expected increase in classification scores. However, this approach led to an increase in verification scores as well, which suggested a decrease in image privacy. Therefore, to address this and introduce realism into the images, we integrated an auxiliary discriminator, thereby evolving our architecture into PriSwin-Dis[3] (Section 4.3, Figure 4.3).

A comprehensive series of experiments was conducted using the PriSwin-Dis architecture to examine the influence of various hyper-parameters and the auxiliary discriminator on the privacy and utility of anonymized images. The configuration and results are shown in Table 6.3 with their respective training and validation loss curves in Figure A.1 (in the order of experiments listed in the Table 6.3). The following are the experiments that were performed to tweak the PriSwin-Dis architecture and examine the effect on privacy and utility:

- **Type of model for auxiliary discriminator** - The PriSwin-Dis architecture was embedded with two different discriminator models: ResNet-18 and ResNet-50. The results indicated that ResNet-18 generally provided a better classification score of

[2]PriSwin-Net Code(https://github.com/rajaatreja/PriSwin-Net).
[3]PriSwin-Dis Code(https://github.com/rajaatreja/PriSwin-Dis).

Exp. No.	Experiment Name	GPU	Batch Size	μ	Auxiliary Weights			Learning Rate		Dis. Loss avg.	Gradient Clipping	Aux. Class. optimized	VR Log Likel.	Class. Score (DenseNet)(↑)	Ver. Score (↓)
					AC	VR	DC	Global	Local						
1	PriCheXy-Net	A100	32	0.01	1	1	-	10^{-4}	-	-	×	✓	✓	76.2	57.7 ± 4.0
2	PriSwin-Dis (ResNet-18)	2080	8	0.01	1	1	1	10^{-4}	-	-	×	✓	✓	78.0	81.0 ± 3.7
3	PriSwin-Dis (ResNet-50)	2080	8	0.01	1	1	1	10^{-4}	-	-	×	✓	✓	**74.5 (DenseNet) 82.9 (SwinT)**	**51.6 ± 1.8**
4	PriSwin-Dis	2080	8	0.01	1	1	1	10^{-4}	10^{-5} (DC)	C_1	×	✓	✓	76.5	69.6 ± 3.6
5	PriSwin-Dis	2080	8	0.01	1	1	1	10^{-5}	-	C_1	×	✓	✓	79.1	75.4 ± 3.48
6	PriSwin-Dis	2080	8	0.01	1	1	1	10^{-4}	-	C_1 & C_2	×	✓	✓	73.4	64.6 ± 3.35
7	PriSwin-Dis	A100	8	0.01	1	1	1	10^{-4}	-	-	×	✓	✓	76.0	62.7 ± 7.8
8	PriSwin-Dis	A100	8	0.01	1	1	1	10^{-4}	-	C_1 & C_2	×	✓	✓	75.0	63.9 ± 5.7
9	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-4}	-	C_1 & C_2	×	✓	✓	76.6	72.5 ± 4.23
10	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-5}	-	C_1 & C_2	×	✓	✓	75.0	63.5 ± 7.67
11	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-4}	10^{-5} (DC)	C_1 & C_2	×	✓	✓	79.0	71.7 ± 2.3
12	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-4}	10^{-5} (DC)	C_1 & C_2	×	×	✓	78.0	68.9 ± 3.5
13	PriSwin-Dis	A100	32	0.01	1	1	0.5	10^{-4}	10^{-5} (DC)	-	DC only	✓	✓	79.0	74.1 ± 4.7
14	PriSwin-Dis	A100	32	0.01	1	1	0.5	10^{-4}	10^{-5} (DC)	C_1 & C_2	DC only	×	✓	79.0	73.8 ± 5.3
15	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-5}	-	C_1 & C_2	×	✓	×	**77.0**	**62.2 ± 3.6**
16	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-5}	-	C_1 & C_2	×	×	✓	76.0	63.2 ± 3.9
17	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-5}	-	C_1 & C_2	×	×	×	78.0	71.3 ± 3.6
18	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-7}	-	C_1 & C_2	×	✓	✓	80.0	82.2 ± 1.2
19	PriSwin-Dis	A100	32	0.01	1	1	1	8^{-6}	-	C_1 & C_2	×	✓	×	79.0	74.1 ± 2.7
20	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-5}	-	-	×	✓	×	78.0	67.5 ± 3.4
21	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-4}	-	-	×	✓	✓	77.0	72.0 ± 2.8
22	PriSwin-Dis	A100	32	0.01	1	1	1	10^{-4}	-	C_1 & C_2	×	✓	✓	75.0	65.0 ± 7.2
23	PriSwin-Dis	A100	32	0.01	1	1	0.5	10^{-4}	-	C_1 & C_2	×	✓	✓	79.0	70.2 ± 1.7
24	PriSwin-Dis	A100	32	0.01	1	1	0.5	10^{-4}	-	C_1 & C_2	×	✓	×	78.0	70.0 ± 3.8
25	PriSwin-Dis	A100	32	0.01	1	1	0.5	10^{-4}	10^{-5} (VR)	C_1 & C_2	×	✓	×	77.0	66.84 ± 5.4
26	PriSwin-Dis	A100	32	0.01	1	0.8	0.5	10^{-4}	-	C_1 & C_2	×	✓	×	77.0	66.51 ± 3.3
27	PriSwin-Dis	A100	32	0.01	1	0.8	1	10^{-5}	-	C_1 & C_2	×	✓	×	77.0	69.9 ± 3.4
28	PriSwin-Dis	A100	32	0.01	1	1	0.3	10^{-5}	-	C_1 & C_2	×	✓	×	77.0	69.2 ± 3.4
29	PriSwin-Dis	A100	32	0.5	1	1	1	10^{-5}	-	C_1 & C_2	×	✓	×	79.0	78.2 ± 3.1
30	PriSwin-Net	A100	32	0.5	1	1	-	10^{-4}	-	C_1 & C_2	×	✓	×	80.0	77.6 ± 3.3

Table 6.3: Results of PriCheXy [Pac+23b] and PriSwin-Dis. Classification evaluations have been performed on pre-trained CheXNet classification models with deformation degree μ. The higher the classification score, the better the utility; the lower the verification score, the better the privacy. For the verification score, SNN was retrained 10 times independently, and the average of all scores was recorded. Auxiliary Classifier (AC), Auxiliary Verification Network (VR), and Auxiliary Discriminator (DC) represent auxiliary classifier, verification, and discriminator networks, respectively. C_1 & C_2 represents the computational graph, i.e., in which parts of the architecture, the discriminator loss of real and fakes are averaged.

78% AUC in *Experiment No. 2* of Table 6.3. While architecture with ResNet-18 was faster to train and less computationally intensive, it had a poor verification score of 81 ± 3.7% AUC. The architecture with ResNet-50 in *Experiment No. 3* of Table 6.3 showed an improved verification score of 51.6 ± 1.8% AUC indicating that increased depth and complexity of ResNet-50 allowed it to capture more intricate patterns in the data, leading to enhanced performance. Therefore, ResNet-50 became the choice for an auxiliary discriminator in all of our experiments to improve the overall efficacy of the PriSwin-Dis architecture.

- **Averaging the Discriminator Loss** - Different strategies for averaging discriminator loss were examined to determine their effects on loss stability and overall performance. Following were the results Table 6.3:

  - No Averaging: Interestingly, the verification performance of *experiment no. 2* outperformed every other experiment by having a score of 51.6 ± 1.8% without hurting the classification score too much when compared to PriCheXy-Net (*experiment no. 1*). But, the scale of DC loss in training and validation loss curves (Figures A.1a and A.1b) was much higher as compared to loss curves of AC and VR. Other experiments of this approach (*Experiment No. 2, 3, 7, 13, 20, 21*) also led to less stable loss curves (all the loss curves are attached in Appendix A), with the discriminator's influence potentially overwhelming the training process. To address this issue, we averaged the DC loss first in the computational graph C_1 only and later both in C_1 and C_2 as shown in code snippet 6.1 and 6.2.

  - Averaging of discriminator loss in C_1 computational graph: This method was not continued in any experiment other than *experiment no. 4* & *5* as their loss curves increased continuously, inferencing no learning. Hence, it did not yield the best results with a verification score as high as 75.4 ± 3.48% (in Exp. no. 5).

  - Averaging of discriminator loss in both C_1 & C_2 computational graphs: Provided the most stable loss curves and the best overall performance. This approach balanced the contributions of real and fake losses, preventing the discriminator from dominating the learning process. One of the notable experiments of this approach is *experiment no. 15* where we achieved a 77% classification score (better than PriCheXy-Net) and 62.2 ± 3.6% verification score.

Averaging the discriminator loss with both C_1 and C_2 emerged as the most effective strategy. It facilitated stable training dynamics, resulting in improved performance metrics for both classification and verification. Therefore, this approach was adopted in subsequent experiments to enhance the robustness of the PriSwin-Dis architecture.

```
def forward(self, real_image, deformed_image):
    real_labels = torch.ones(real_image.size(0), 1).cuda()
    fake_labels = torch.zeros(deformed_image.size(0), 1).cuda()

    # Convert and process images
    real_image = self.transform(real_image)
    deformed_image = self.transform(deformed_image)

    # Compute the discriminator output
    dis_r = self.dis_model(real_image)
    dis_f = self.dis_model(deformed_image)
    loss_r = self.bce_loss(dis_r, real_labels)
    loss_f = self.bce_loss(dis_f, fake_labels)

    # Average loss in C1 computational graph
    # TODO: comment out if average loss not needed
    return (loss_r + loss_f)/2

    # Totoal loss in C1 computational graph
    # TODO: comment out if average loss needed
    return loss_r + loss_f
```

Listing 6.1: Forward function of auxiliary discriminator model part of C_1 computaional graph of PriSwin-Dis architecture.

```
    # Optimize discriminator
    real_labels = torch.ones(inputs1.size(0), 1).cuda()
    fake_labels = torch.zeros(fakes_1.size(0), 1).cuda()

    real_image = transform_dis(inputs1)
    deformed_image = transform_dis(fakes_1.detach())

    dis_r = dis_loss.dis_model(real_image)
    dis_f = dis_loss.dis_model(deformed_image)

    loss_r = criterion_dis(dis_r, real_labels)
```

```
    loss_f = criterion_dis(dis_f, fake_labels)

    total_dis_loss = (loss_r + loss_f)/2
    # Average loss in C2 computational graph
    # TODO: comment out if average loss not needed
    total_dis_loss = (loss_r + loss_f)/2

    # Totoal loss in C2 computational graph
    # TODO: comment out if average loss needed
    total_dis_loss = loss_r + loss_f

    total_dis_loss.backward()
    optimizer_dis.step()
    dis_loss.dis_model.eval()
```

Listing 6.2: Optimization of auxiliary discriminator model part of C_2 computaional graph of PriSwin-Dis architecture.

- **Global and Local Learning Rates** - The term Global Learning Rate (GLR) is used for the experiments where the same learning rate is used for the entire PriSwin-Dis architecture, and Local Learning Rate (LLR) specifies the learning rate for specific auxiliary models, such as the auxiliary discriminator, use a different learning rate than the rest of the architecture. This approach was implemented because the auxiliary discriminator loss was converging very fast (e.g., *exp. no. 2*) within about 40-50 epochs, as shown in its respective training and validation loss curves (Figures A.1a and A.1b). The impact of global and local learning rates on the PriSwin-Dis architecture was thoroughly analyzed to identify the optimal training configurations. The results were as follows:

  - Lower Global Learning Rate (10^{-5}): Resulted in better classification scores (*Experiment Nos. 5, 10, 15-17, 20, 27-29*), indicating enhanced utility. This setting also contributed to more stable training dynamics and smoother loss curves. However, the verification scores in this case were also high, indicating lower privacy. In *exp. no. 15* by tweaking other hyperparameters along with a GLR of 10^{-5} verification score of $62.2 \pm 3.6\%$ was achieved at 77% classification score.
  - Extremely Low Learning Rate (10^{-7} & 8^{-6}): We also tried with very low GLRs to examine the effect of slower auxiliary discriminator learning on the whole

architecture, but it failed to train the model effectively, as evidenced in *exp. 18 & 19* by nearly flat loss curves in Figures A.1ae and A.1af. This setting did not provide the model with sufficient learning capacity, leading to suboptimal performance.

– Local Learning Rate (10^{-5} in DC/VR): After observing the loss curves of exp. no. 3, we slowed down the convergence of DC in *Exp. 4, 11-14, 25* in order to let the DC model learn at the same pace as the whole architecture. However, the resultant models' verification score was almost 70% in every case and could not outperform the experiments with GLR.

These experiments highlighted the necessity of optimizing learning rates to balance the trade-off between training stability and model performance. A lower GLR of 10^{-5} was identified as optimal, offering improved verification scores and stable training. This learning rate configuration was therefore adopted in subsequent experiments to enhance the overall efficacy of the PriSwin-Dis architecture.

- **GPU and Batch Size** - The influence of different GPU models and batch sizes on the training efficiency and performance of the PriSwin-Dis architecture was evaluated. The results of different experiments were as follows:

  – 2080 with Batch Size 8 (*Exp. 2-6*) - Exp. no. 3 showed the best results in terms of classification and verification scores. However, the loss curves were not satisfactory, likely due to the smaller batch size causing oscillations in the loss curve(Figures A.1a and A.1b).

  – A100 with Batch Size 8 (*Exp. 7 & 8*) - To further investigate the loss curve behavior, the same configuration from exp. no. 2 was run on an A100 GPU. The loss curves (Figures A.1i to A.1l) clearly showed that the discriminator was converging very early in the training stages and only increased thereafter, indicating no further effective training.

  – A100 with Batch Size 32 (*Exp. 9-30*) - This configuration resulted in better loss curves. The increased batch size helped stabilize the training process. By tweaking other hyperparameters, this setup achieved better classification performance, particularly evident in experiment number 15.

While the 2080 GPU with a batch size of 8 initially showed promising results, the instability in the loss curves necessitated further investigation. Running the same

configuration on an A100 GPU highlighted the early convergence issue with the discriminator. Ultimately, using an A100 GPU with a batch size of 32 provided more stable loss curves and improved overall performance. This configuration was adopted for subsequent experiments, leading to enhanced classification performance through additional hyperparameter tuning.

- **Gradient Clipping in Discriminator Loss** - As observed in the above experiments, the irregular behavior of auxiliary discriminatory loss curves, we tried applying gradient clipping to the discriminator loss in the *exp. no. 13* & *14*. We investigated their results to determine the effects on training stability and model performance, which are as follows:

  – Without Gradient Clipping: The discriminator loss curves exhibited significant spikes and instability, as observed in experiments *3, 6,* & *11*. These fluctuations (see Figures A.1q and A.1r) hindered the training process, leading to suboptimal performance and inconsistent results.

  – With Gradient Clipping: Implementing gradient clipping resulted in less spiky and more stable DC loss curves, as seen in experiments *13 & 14*. However, the classification and verification performance did not significantly improve despite the improved stability.

  Gradient clipping effectively stabilized the discriminator loss curves, reducing spikes and promoting consistent training dynamics. However, the expected improvements in classification and verification performance were not realized. This suggested that while gradient clipping enhanced training stability, it may need to be combined with other techniques to achieve better overall performance in the PriSwin-Dis architecture. In further exploration of complementary methods, we were able to stabilize the DC loss curve without using gradient clipping, e.g., in Exp. No. *15*, just by tweaking the other parameters.

- **Verification Loss Likelihood Removal** - In the exp. *12-14, 22* & *23* a high log-likelihood verification loss was observed in the training and validation loss curves (Figures A.1s to A.1x). Therefore, the effect of removing the likelihood component from the verification loss function was examined to determine its impact on the loss curve scale, stability, and overall performance. In exp. no. *15, 17, 19, 20,* & *24-30*, we removed the log-likelihood verification loss and used the BCE verification loss to contribute to the total loss of the PriSwin-Dis architecture.

The removal of the likelihood component from the verification loss function proved beneficial for the stability of the training process. It led to smoother loss curves (Appendix A) of the respective experiments and more reliable training dynamics. When compared to the architecture where log-likelihood was used, the experiments with its removal resulted in lower verification scores, indicating higher privacy in the anonymized images generated. For e.g., exp. no. *15* (Table 6.3) produced 62.2 ± 3.6% verification score, which is better than other similar types of experiments mentioned.

- **Auxiliary Model Weights** - Various weights for the auxiliary model components were tested to find the optimal balance between classification accuracy and verification scores. The findings were as follows:
  - Equal Weights: Using equal weights for all auxiliary components (AC, VR, and DC) provided a baseline for comparison. This approach was used in several experiments, including experiments *3* and *15*. The optimal configuration was found when specific adjustments were made to the auxiliary model parameters. By fine-tuning these parameters and keeping the weight of all auxiliary models equally, better performance metrics were achieved, particularly in experiment *15*, which showed improved classification accuracy with a 77% AUC score, and in exp. *3* where we achieved the best verification score of 51.6 ± 1.8% AUC.
  - Adjusted Weight of AC: Experimenting with different weights, particularly reducing the discriminator weight (e.g., setting DC to 0.5), showed improvements in classification scores. For instance, in experiments *23*-*25*, where the DC weight was experimented with, the classification performance improved with scores between 77% to 79%. However, the verification score was reduced.
  - Adjusted Weight of VR: With a similar effect as of AC weight, in exp. no. *26, 27*, a lower VR weight resulted in a better verification score of 66.51 ± 3.3% in comparison to other experiments of the same category.

Adjusting the auxiliary model weights is crucial for optimizing the performance of the PriSwin-Dis architecture for privacy or utility tasks. Fine-tuning these weights, especially reducing the discriminator weight and verification weight, can lead to some improvements in classification accuracy and verification scores. These results were used to conclude that a lower VR weight results in a sensible verification score, and a lower AC weight can increase the classification score. However, weighting these auxiliary models together cannot guarantee to improve the privacy-utility trade-off of

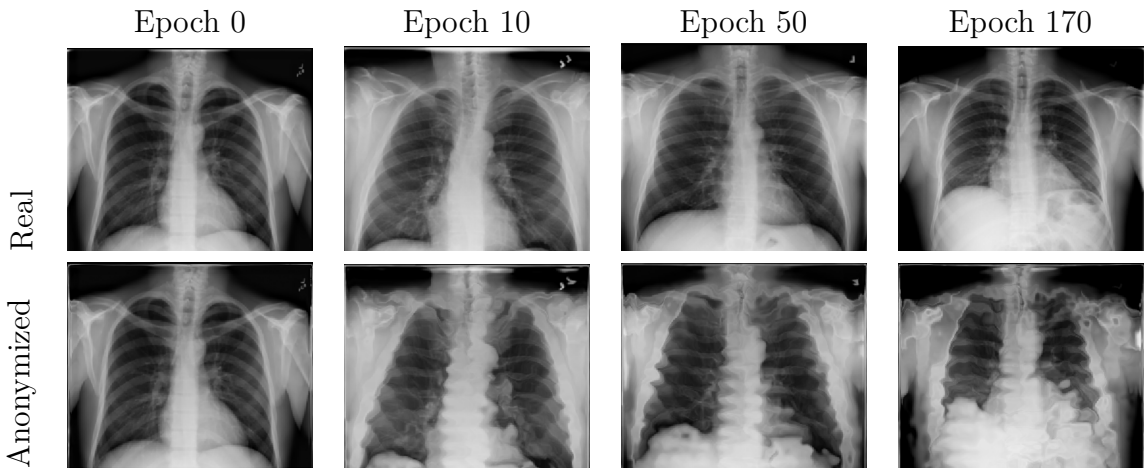


Figure 6.4: Examples of anonymized images from PriCheXy-Dis Exp. No. 29 shows the amount of deformation at different epochs of the training. The first row shows the real images from the ChestX-ray14 [Wan+17] dataset; the second row has the images anonymized by PriSwin-Dis at deformation degree $\mu = 0.5$.

the anonymization architecture, as shown in the results of exp. no. *26*. This suggested keeping the auxiliary weights equal, as the exp. *3* and *15* had the better performance.

- **Deformation Degree (μ)** - The degree of deformation (μ) applied to the images was varied to study its impact on the realism and privacy of the anonymized images. The default deformation degree in our experiments with PriSwin-Dis was $\mu = 0.01$ to have a direct comparison with the best performance of PriCheXy-Net [Pac+23b]. This setting provided a good balance between image realism and privacy, maintaining sufficient detail for utility while anonymizing the data effectively.

  We also tried with a higher deformation degree $\mu = 0.5$ in Exp. No. *29* and *30*, which resulted in more significant alterations to the images, as shown in Figure 6.4. Visually, the images became less recognizable. However, it is interesting to note that transformer-based AC was able to retain important disease details needed for classification, even at higher deformation degrees. Furthermore, it exhibited improved classification performance, achieving approximately 79% AUC. However, there was a compromise on privacy at higher verification scores (worse), which reached around 78% AUC.

  The deformation degree (μ) is a critical parameter for controlling the trade-off between image realism and privacy. A deformation degree of 0.01 was found to provide an optimal balance, ensuring that images remain useful for classification tasks while

effectively anonymizing them to protect privacy. This setting was, therefore, adopted in most experiments to maintain the desired equilibrium between utility and privacy in the PriSwin-Dis architecture.

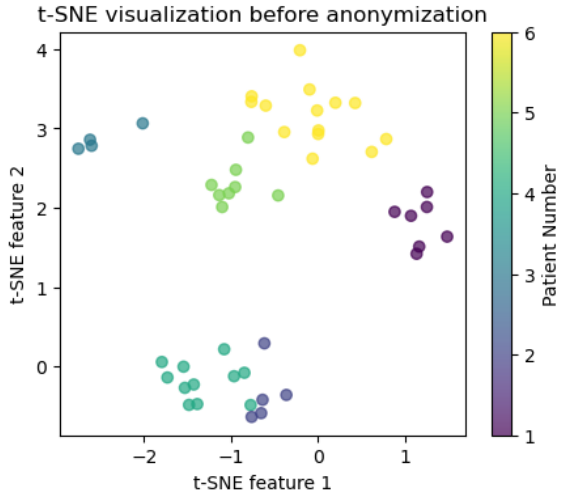
6.3 Investigating Lower-Dimensional Space Transformations

After a thorough investigation of anonymization models, we performed a few experiments to visualize and quantify the difference between the real image and the anonymized image by analyzing their lower dimensional transformations. We also investigated the similarity score of an anonymized image calculated by the SNN.

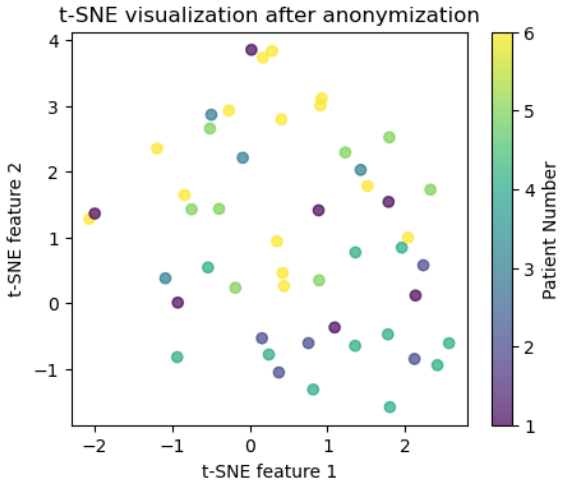
Visual Analysis of Lower-Dimensional Transformations

In this experiment, we explored lower-dimensional space transformations to understand the inner workings of the SNN network and visualize the effect of anonymization on the similarity of images.

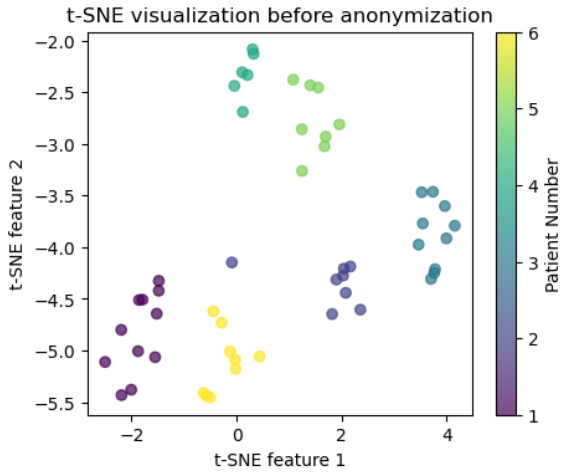
- Dataset Preparation:
  - Dataset Creation: A separate dataset was created from the test set provided used in the research of [Pac+23b]. This dataset was filtered to include images of patients with a single disease.
  - Patient Selection: Patients with more than 10 follow-up images were selected, ensuring their age ranged between 25 and 60 years to focus on fully developed lungs.
- Methodology:
  - Vector Datasets:
    - Before Anonymization: 128-size vectors of original images.
    - After Anonymization: 128-size vectors of anonymized images using the best PriSwin-Dis model (Experiment No. 3).
  - Vector Extraction: The pre-trained SNN network was truncated to extract 128-size vector outputs from ResNet-50 corresponding to each image.



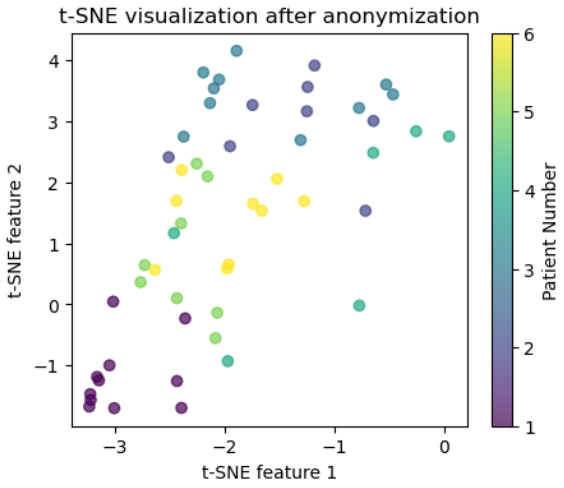
(a) Patients with different diseases before anonymization.



(b) Patients with different diseases after anonymization.



(c) Patients with no diseases before anonymization.



(d) Patients with no diseases after anonymization.

Figure 6.5: Before and after anonymization, t-SNE visualizations of patient images are created through lower-dimensional analysis of a 128-size vector obtained from SNN. Six different patients are shown in different colors, each having multiple images, for each case of different and no disease. In both cases, images from the same patients could not be clustered together anymore after anonymization. This implies that SNN failed to perform a linkage attack after anonymization of images.

  - t-distributed Stochastic Neighbor Embedding (t-SNE) Analysis: The t-SNE (t-distributed Stochastic Neighbor Embedding) analysis was performed on 128-size vectors obtained from the SNN network to visualize the effect of anonymization.

- Experiments:

  - Patients with Different Diseases:

    * Before Anonymization: t-SNE visualization of 128-size vectors showed clear cluster formations as shown in Figure 6.5a. Each cluster represented images from the same patient, indicating that the SNN could reidentify all images belonging to one patient. This clustering demonstrated that images from the same patient were closely located in the lower-dimensional space before anonymization.
    * After Anonymization: t-SNE visualization showed no clear cluster formations (see Figure 6.5b). The data points were scattered equally in a Gaussian space, indicating that the SNN could not reidentify images from the same patient. This dispersion suggested successful anonymization and higher privacy.

  - Patients with No Diseases:

    * Before Anonymization: Similar to the first experiment, t-SNE visualization showed distinct clusters for each patient (see Figure 6.5c), demonstrating the SNN's ability to reidentify all images from the same patient.
    * After Anonymization: t-SNE visualization showed scattered data points without clear cluster formations (see Figure 6.5d), indicating that the anonymization process disrupted the SNN's ability to reidentify images, thus ensuring higher privacy.

The lower-dimensional analysis using t-SNE visualization effectively demonstrated the impact of anonymization on image similarity. The distinct clusters before anonymization indicated that the SNN could reidentify images from the same patient. However, the scattered data points after anonymization confirmed that the PriSwin-Dis model successfully anonymized the images, enhancing privacy. This experiment highlights the efficacy of the PriSwin-Dis model in ensuring the privacy of medical images while maintaining utility.

Disease Types	Patient No.	Avg. Dist. (Euclidean)	
		Before Anon.	After Anon.
Different Diseases	1	16.03	21.65
	2	14.88	13.63
	3	14.00	20.02
	4	15.90	18.00
	5	14.48	17.06
	6	16.40	19.14
No Disease	1	15.74	18.05
	2	16.94	17.82
	3	14.38	19.47
	4	13.83	31.60
	5	16.19	17.85
	6	17.28	15.10

Table 6.4: Analysis of averaged lower dimensional Intra-class distance before and after anonymization between the patients having different diseases and patients having no disease.

Quantitative Analysis of Lower-Dimensional Transformations

In addition to the visual analysis using t-SNE, the results were also quantified by calculating the intra-class distances between images before and after anonymization. This analysis was also performed for both patients with different diseases and those with the same disease (Table 6.4).

- Methodology:
  - Intra-Class Distance Calculation: The intra-class distance was measured as the average Euclidean distance between the 128-size vectors of images belonging to the same patient.
  - Comparison Before and After Anonymization: Intra-class distances were calculated for the original images and the anonymized images to quantify the effect of anonymization.
- Findings:
  - Before Anonymization:
    - Patients with Different Diseases: Images from the same patient had comparatively lower intra-class distances, indicating that they were closely clustered in the lower-dimensional space.
    - Patients with No Diseases: Similar results were observed, with images from the same patient forming tight clusters.
  - After Anonymization:
    - Increased Intra-Class Distances: In almost all the cases, intra-class distances increased significantly after anonymization in both cases. This indicated that images that were originally close together and formed clusters were now dispersed and lay farther apart in the lower-dimensional space.
    - Patients with Different Diseases and No Diseases: Both groups showed a substantial increase in intra-class distances after anonymization (Table 6.4), confirming that the anonymization process disrupted the clusters.
- Implications:
  - Cluster Disruption: The increase in intra-class distances after anonymization demonstrated that the anonymized images no longer formed tight clusters. This

dispersion in the lower-dimensional space meant that the SNN could no longer reidentify images from the same patient, ensuring higher privacy.

– SNN Performance: The failure of the SNN to reidentify images from the same patient after anonymization was directly linked to the increased intra-class distances. Hence, anonymization effectively enhanced the privacy of medical images.

The quantitative analysis corroborated the visual findings from the t-SNE plots. By increasing the intra-class distances between images from the same patient, the anonymization process effectively disrupted the clusters in the lower-dimensional space. This led to the failure of the SNN to reidentify images from the same patient, indicating successful anonymization and higher privacy. The PriSwin-Dis model (Experiment No. 3) was thus validated as an effective method for anonymizing medical images while maintaining their utility.

Image Pair	Image 1 (anonymized)	Image 2 (Real)	Real Label	Similarity Score (SNN)	SNN Predicted Label
Same Patient	00010012_009.png	00010012_025.png	1.0	0.412	0.0
	00010012_009.png	00010012_038.png	1.0	0.265	0.0
	00010012_009.png	00010012_004.png	1.0	0.235	0.0
	00010012_009.png	00010012_016.png	1.0	0.194	0.0
	00010012_009.png	00010012_037.png	1.0	0.151	0.0
	00010012_009.png	00010012_032.png	1.0	0.139	0.0
	00010012_009.png	00010012_003.png	1.0	0.128	0.0
	00010012_009.png	00010012_024.png	1.0	0.119	0.0
	00010012_009.png	00010012_030.png	1.0	0.109	0.0
	00010012_009.png	00010012_008.png	1.0	0.075	0.0
Different Patient	00010012_009.png	00005609_024.png	0.0	0.870	1.0
	00010012_009.png	00021489_014.png	0.0	0.857	1.0
	00010012_009.png	00016807_001.png	0.0	0.855	1.0
	00010012_009.png	00019917_004.png	0.0	0.854	1.0
	00010012_009.png	00007894_001.png	0.0	0.854	1.0
	00010012_009.png	00022993_000.png	0.0	0.853	1.0
	00010012_009.png	00021303_016.png	0.0	0.853	1.0
	00010012_009.png	00022993_011.png	0.0	0.851	1.0
	00010012_009.png	00016094_010.png	0.0	0.851	1.0
	00010012_009.png	00017236_101.png	0.0	0.850	1.0

Table 6.5: Results of top-10 similarity scores of SNN sorted in descending order. These results are for the patient 00010012. The first ten results are compared with all other images of the same patients, and the next ten are results when compared to the images of different patients.

Ranking Images Based on SNN Similarity Score for Anonymized Patients

To further evaluate the effectiveness of the PriSwin-Dis model in anonymizing medical images, we conducted an experiment to rank images based on the similarity scores produced by the SNN network for a given patient. This experiment aimed to assess whether the SNN could correctly identify and rank the original images corresponding to the anonymized images as shown in Tables 6.5 and A.1.

- Dataset Preparation: We created a dataset by selecting 12 random images from the test set and anonymized them using the PriSwin-Dis (Exp. No. 3) model. This dataset was divided into two parts: one with pairs of images from the same patient and another with pairs of different patients. In these pairs, only the selected 12 images were anonymized, while the rest remained unanonymized. Pairs with images from the same patient were labeled as 1, and pairs with images from different patients were labeled as 0.

- Similarity Score Calculation: The SNN network was used to calculate a similarity score (ranging from 0 to 1) for each pair of images. If the similarity score was greater than 0.5, the pair was labeled as a match (1). This calculation was performed for both datasets (same pairs and different pairs).

- Ranking and Analysis: The similarity scores were then sorted from highest to lowest, and the top 10 highest scores were examined to see the top 10 images classified as similar by the SNN to the particular anonymized image. The results showed that in almost all cases (Tables 6.5 and A.1), the SNN did not rank the real images in the top 10 similar images corresponding to the anonymized image. Even pairs with the same image before and after anonymization did not appear in the top 10 similar images.

This experiment demonstrated that the anonymization process was highly effective in fooling the re-identification network. The anonymized images were sufficiently altered, ensuring higher privacy and successfully preventing the SNN from recognizing and ranking the original images. The results confirm that the PriSwin-Dis model enhanced privacy by disrupting the SNN's ability to reidentify images, thereby protecting patient privacy while maintaining the utility of the anonymized images.

Chapter 7

Discussion and Conclusion

7.1 Discussion and Analysis of Classification and Verification Performance

The comparative analysis of PriSwin-Net, PriSwin-Dis, and PriCheXy-Net architectures highlights the improvements and trade-offs in classification and verification performance. In this study, PriCheXy-Net [Pac+23b] serves as the baseline against which the modifications in PriSwin-Net and PriSwin-Dis are evaluated. PriCheXy-Net achieved an AUC of 76.2% classification score with the DenseNet-based CheXNet classifier on rerun and a verification performance of 57.7 ± 4.0%, indicating a reasonable balance between privacy and utility.

PriSwin-Net

The primary modification in PriSwin-Net was the integration of the Swin Transformer (SwinCheX) to replace the DenseNet-based auxiliary classifier. This change aimed to enhance the utility of anonymized images.

- Classification Performance:
  - With Optimized Auxiliary Classifier: PriSwin-Net showed an AUC of 75.4% with DenseNet and 83.2% with SwinCheX, indicating improved utility when evaluated with SwinCheX. However, this improvement was accompanied by domain shift issues when the model was evaluated with DenseNet.
  - Without Optimized Auxiliary Classifier: Achieving a higher classification score of 77.0% with DenseNet, this approach addressed the domain shift issue by

not updating the auxiliary classifier during training. This result suggests that freezing the auxiliary classifier during training helps mitigate domain shift and maintain better classification performance.

- Verification Performance:
  - With Optimized Auxiliary Classifier: The verification score was 62.9 ± 4.9%, which was worse than PriCheXy-Net. This result indicates that the improved utility of the Swin Transformer came at the cost of reduced privacy.
  - Without Optimized Auxiliary Classifier: Achieved a verification score of 66.6 ± 4.0%, slightly higher but comparable to PriCheXy-Net. This indicates that while freezing the auxiliary classifier improved classification performance, it slightly degraded verification scores.

Hence, the integration of the Swin-T in PriSwin-Net significantly enhanced the utility of anonymized images, particularly when evaluated with the SwinCheX classifier. However, the domain shift issue, when evaluated with DenseNet, indicates that the improvements in utility were not universal across all classifiers. Freezing the auxiliary classifier during training mitigated this issue, suggesting that model stability and consistency across different evaluation methods are crucial. The slight degradation in verification scores when freezing the auxiliary classifier indicates a trade-off between maintaining classification performance and achieving optimal privacy.

PriSwin-Dis

PriSwin-Dis incorporated an auxiliary discriminator to introduce realism into the anonymized images, aiming to improve both privacy and utility through careful tuning of hyperparameters.

- Classification Performance:
  - With ResNet-50 Discriminator (Exp. 3 Table 6.3): Achieved an AUC of 74.5% with DenseNet and 82.9% with SwinCheX, showing substantial improvements in utility.
  - With Averaged Discriminator Loss (Exp. 15): Achieved an AUC of 77.0% with a verification score of 62.2 ± 3.6%, indicating a better balance between privacy and utility.

- Verification Performance:
  - Best Performance (Exp. 3): Achieved the lowest verification score of 51.6 ± 1.8% with ResNet-50, indicating enhanced privacy.
  - Stable Loss Curves (Exp. 15): With a lower global learning rate and averaged discriminator loss, PriSwin-Dis achieved more stable loss curves and improved verification scores.

Hence, the introduction of an auxiliary discriminator in PriSwin-Dis, along with the careful tuning of hyperparameters such as learning rates and averaging discriminator loss, significantly improved the privacy-utility trade-off. The best configuration (Experiment 3) demonstrated both enhanced classification accuracy and lower verification scores, surpassing the performance of PriCheXy-Net. The stable loss curves observed in Experiment 15 further highlight the importance of optimizing training dynamics to achieve consistent and reliable results. These findings suggest that the auxiliary discriminator plays a crucial role in improving the realism of anonymized images, thereby enhancing both utility and privacy.

The advanced model architectures and hyperparameter optimization in PriSwin-Net and PriSwin-Dis significantly improved the effectiveness of anonymization techniques in medical imaging. PriSwin-Dis, in particular, demonstrated that integrating an auxiliary discriminator and optimizing training dynamics can achieve a superior balance between classification performance and privacy. These findings underscore the importance of continuous refinement and innovation in developing robust anonymization models that protect patient privacy while maintaining the utility of medical images.

Classification and Verification Performance in Lower-Dimensional Space

The results obtained from lower-dimensional analysis provided additional insights into the impact of anonymization on image similarity and re-identification risk.

- Visual Analysis with t-SNE:
  - Patients with Different Diseases: Before anonymization, t-SNE visualizations showed clear cluster formations for each patient, indicating that the SNN could reidentify images from the same patient. After anonymization, these clusters were disrupted, and data points were scattered, indicating successful anonymization.

  - Patients with No Diseases: Similar results were observed, with clear clusters before anonymization and dispersed data points after anonymization, ensuring higher privacy.

- Quantitative Analysis of Intra-Class Distances: The intra-class distances increased significantly after anonymization, confirming that images that were originally close together were now dispersed in the lower-dimensional space. This disruption ensured that the SNN could not reidentify images from the same patient, enhancing privacy.
- Ranking Images Based on SNN Similarity Score: An experiment was conducted to rank images based on SNN similarity scores for a given patient. The results showed that in almost all cases, the SNN did not rank the real images in the top 10 similar images corresponding to the anonymized image, indicating effective anonymization.

These experiments highlight the effectiveness of the PriSwin-Dis model in ensuring the privacy of medical images while maintaining their utility. The visual and quantitative analysis demonstrated that the anonymization process effectively disrupted the SNN's ability to re-identify images, ensuring higher privacy.

7.2 Impact of Optimization Changes on Privacy-Utility Trade-off

The impact of various optimization changes on the privacy-utility trade-off was evaluated by analyzing the classification and verification performance of PriSwin-Net and PriSwin-Dis in comparison to the baseline PriCheXy-Net. The results are visually represented in Figure 7.1, where the AUC scores are plotted against the SNN verification scores, with the y-axis inverted to reflect higher privacy at lower values.

CheXNet [Pac+23b] and SwinCheX (ours) serve as pre-trained classifiers (blue dots in Figure 7.1), showing the classification and verification performance on real, unanonymized data. In comparison with CheXNet's 80.5% and 81.8%, SwinCheX showed an AUC of 84.3% with the same verification score of 81.8%. These benchmarks highlight the performance on real data, providing a context for evaluating our proposed models.

PriCheXy-Net on re-evaluation, serving as the baseline, achieved an AUC of 76.2% and a verification score of 57.7 $\pm$ 4.0% (green square in Figure 7.1). This model balanced privacy and utility reasonably well, setting a standard against which PriSwin-Net and PriSwin-Dis were evaluated.

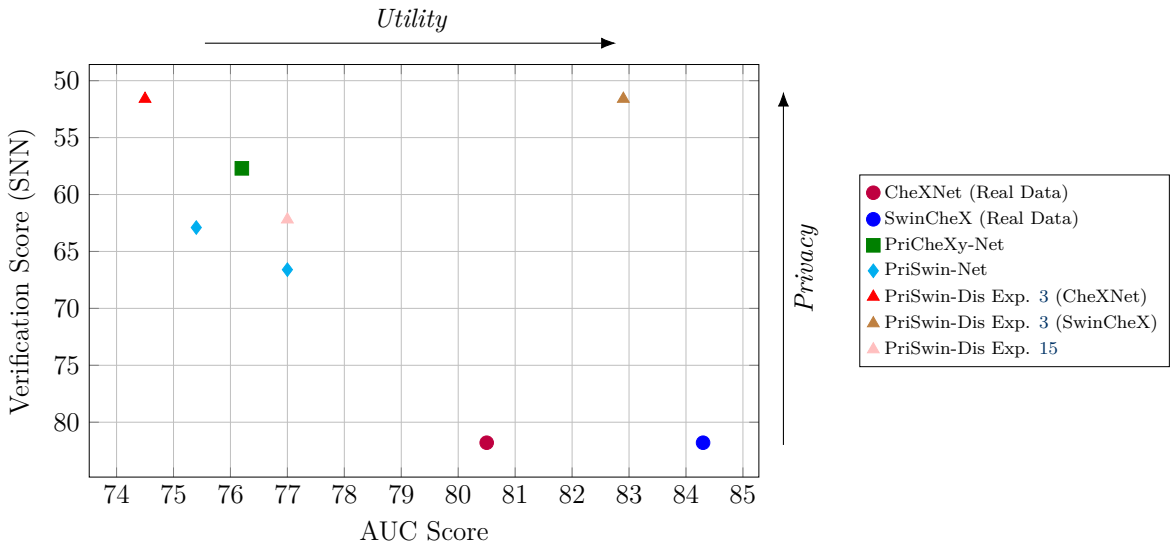


Figure 7.1: Graphical representation of the results in terms of the privacy-utility trade-off. The y-axis indicates patient verification performance evaluated by retrained SNN network, reflecting privacy levels, while the x-axis shows the classification performance of anonymized images, representing data utility. The legend shows the two classifiers used: CheXNet and SwinCheX, and PriCheXy-Net for baseline comparison with our proposed models PriSwin-Net and PriSwin-Dis (with Exp. no. referring to experiments in Table 6.3).

PriSwin-Net was modified by integrating the Swin-T to replace the DenseNet-based auxiliary classifier. With the optimized auxiliary classifier, PriSwin-Net (shown as a brown diamond point in Figure 7.1) achieved an AUC of 75.4% with DenseNet and 83.2% with SwinCheX but had a verification score of 62.9 ± 4.9%, indicating improved utility but reduced privacy. Without optimizing the auxiliary classifier, the model achieved a higher AUC of 77.0% and a verification score of 66.6 ± 4.0%, addressing domain shift issues but slightly degrading privacy.

PriSwin-Dis (red triangle in Figure 7.1) incorporated an auxiliary discriminator to introduce realism into the anonymized images. In Experiment 3 of PriSwin-Dis, it achieved an AUC of 74.5% with DenseNet and 82.9% with SwinCheX, with a significantly lower verification score of 51.6 ± 1.8%, indicating enhanced privacy. In experiment 15, it showed an AUC of 77.0% and a verification score of 62.2 ± 3.6%, providing a balanced trade-off with stable loss curves.

The optimal balance between privacy and utility was found in the PriSwin-Dis Experiment 3 with SwinCheX, located in the top-right corner of the plot. This configuration

demonstrated that integrating an auxiliary discriminator and careful hyperparameter tuning could achieve substantial improvements in both classification accuracy and privacy.

In conclusion, the advanced model architectures and optimization strategies in PriSwin-Dis, particularly in Experiment 3 with SwinCheX, significantly improved the privacy-utility trade-off compared to the baseline PriCheXy-Net. These findings underscore the effectiveness of integrating an auxiliary discriminator and optimizing training parameters to achieve superior anonymization while maintaining high utility in medical images.

7.3 Conclusion

In this thesis, we have explored the integration of advanced model architectures and optimization strategies to enhance the utility and privacy of anonymized medical images, focusing on the PriSwin-Net and PriSwin-Dis models. Our primary objective was to improve the balance between data utility and privacy, particularly in the context of medical imaging data used for diagnostic purposes.

The modifications implemented in PriSwin-Net, including the incorporation of the Swin Transformer (SwinCheX) as an auxiliary classifier, resulted in significant improvements in the classification performance of anonymized images. The model achieved an AUC of 75.4% with DenseNet and 83.2% with SwinCheX, indicating enhanced utility when evaluated with the SwinCheX classifier. However, the domain shift issue observed when using DenseNet highlights the importance of ensuring model stability and consistency across different evaluation methods.

In PriSwin-Dis, the integration of an auxiliary discriminator aimed to introduce realism into anonymized images, thereby improving both privacy and utility through careful hyperparameter tuning. This model demonstrated substantial improvements, achieving an AUC of 74.5% with DenseNet and 82.9% with SwinCheX, with significantly lower verification scores, indicating enhanced privacy. The optimal configuration, as seen in Experiment 3 (Table 6.3) of PriSwin-Dis, achieved both high classification accuracy and low verification scores, underscoring the effectiveness of this approach.

Overall, the findings from this study highlight the critical role of advanced model architectures, such as the Swin-T, and optimization strategies, including the use of auxiliary discriminators, in achieving an improved and better privacy-utility trade-off. These results underscore the importance of continuous refinement and innovation in developing robust

anonymization models that can protect patient privacy while maintaining the utility of medical images for diagnostic purposes.

7.4 Suggestions for Future Improvements in X-ray Data Anonymization

Based on this thesis's findings, quite a few open-ended areas for future research and improvement in X-ray anonymization have been identified.

Addressing domain shift problems is crucial for ensuring model stability and consistency across different evaluation techniques. Future research should focus on advanced methods in domain adaptation and transfer learning to mitigate these issues. Enhancing transformer-based models beyond the Swin Transformer could yield further improvements in both privacy and utility. Exploring variations of transformers and their applications in medical image anonymization can provide new insights and enhance model performance.

Stabilizing training dynamics is essential for reliable anonymization model performance. Developing techniques to stabilize these dynamics, such as adaptive learning rates, advanced regularization methods, or more robust optimization algorithms, should be a priority. Additionally, optimizing hyperparameters for anonymization models will be vital. This includes fine-tuning parameters such as learning rates, batch sizes, and the weighting of loss functions to achieve the best balance between privacy and utility.

Integrating data from multiple applications, such as combining chest X-rays with brain X-rays, could help in building a robust generic anonymization technique. Leveraging the strengths of different modalities may improve overall performance. Evaluating the real-world applicability of anonymization models is also important. Conducting extensive evaluations on diverse datasets, considering different patient demographics, and analyzing the impact of anonymization on clinical decision-making will provide valuable insights into practical applicability.

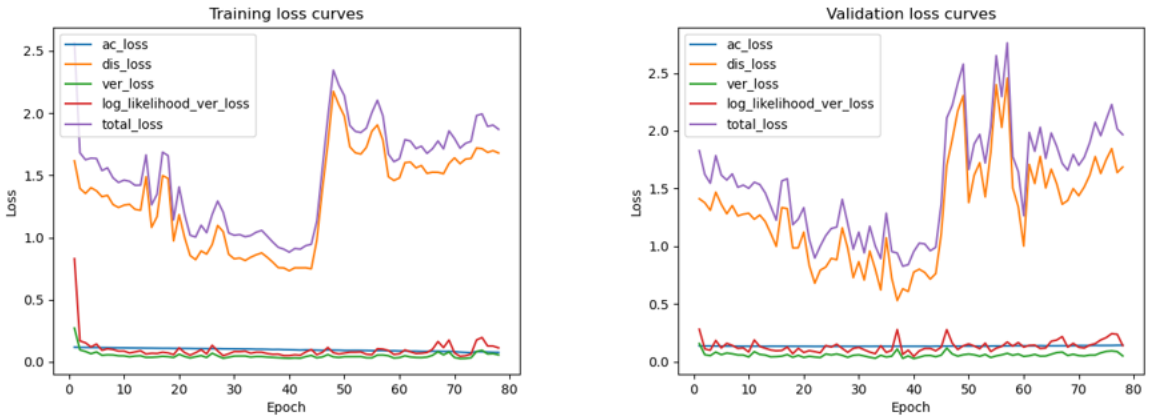
Exploring advanced adversarial techniques in anonymization is another promising area. Future work could investigate more sophisticated adversarial networks and their effectiveness in preserving privacy while maintaining high data utility. Developing user-friendly implementations and tools for medical practitioners and researchers to easily anonymize X-ray data is crucial. This could involve creating open-source software packages or integrating anonymization techniques into existing medical imaging platforms.

Techniques involving generating synthetic medical data using generative models like Latent Diffusion Models (LDMs) [Pac+23a] in combination with dedicated privacy presents a promising solution concerning anonymizing medical images. This method could be a potential solution for creating large public medical datasets without compromising privacy.

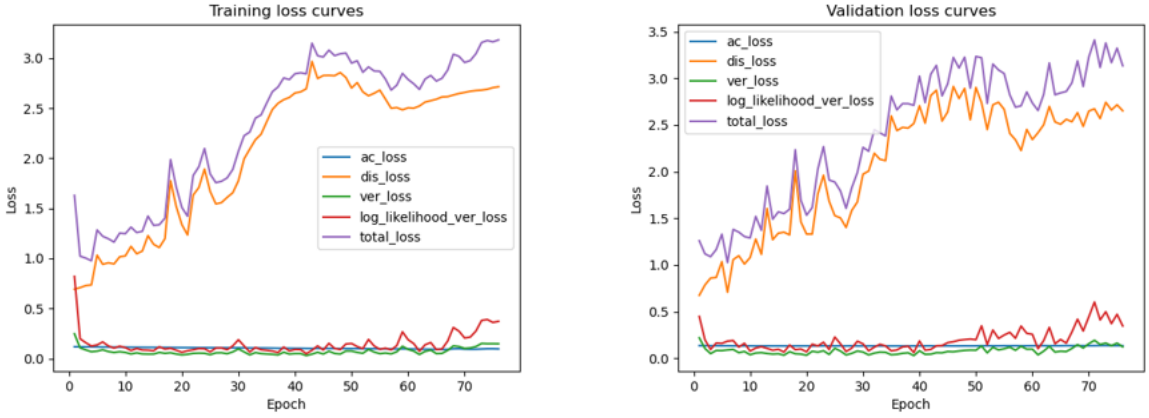
Finally, by focusing on these areas, future research can significantly advance the field of X-ray anonymization, contributing to the development of more secure, effective, and practical methods for protecting patient privacy while maintaining the utility of medical images.

Appendix A

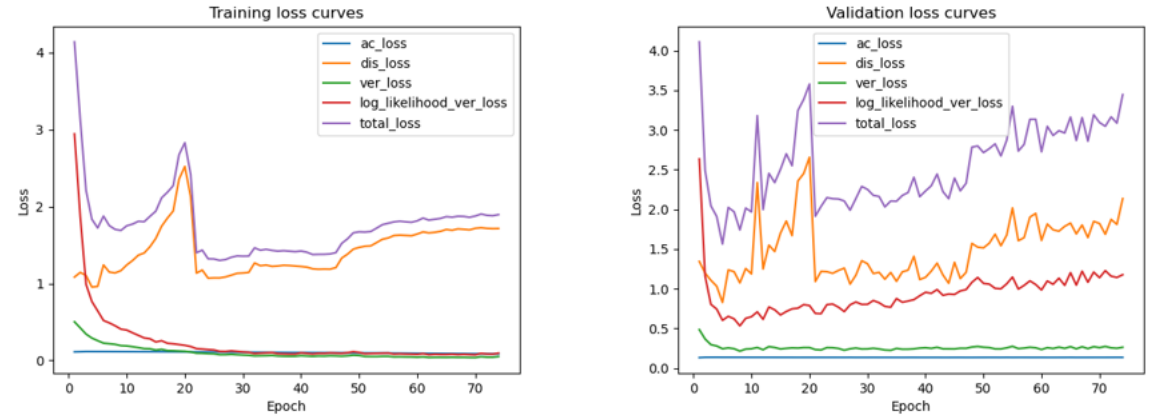
Appendix



(a) (b)

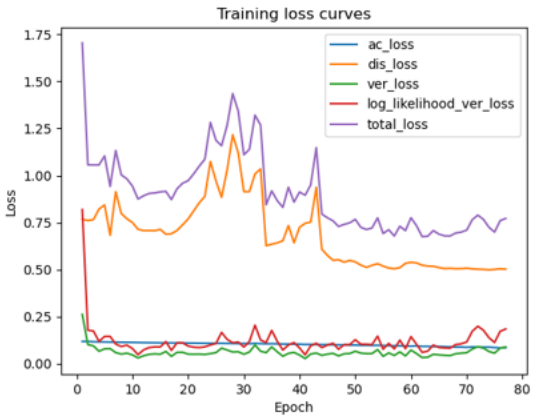


(c) (d)

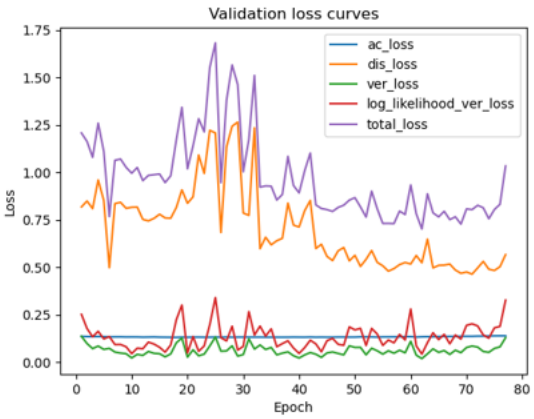


(e) (f)

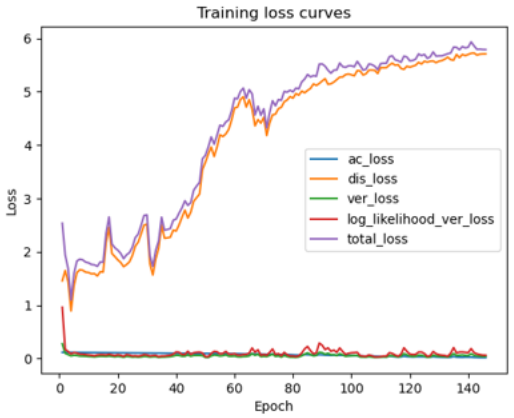
Figure A.1: Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 3 to 5 in Table 6.3.



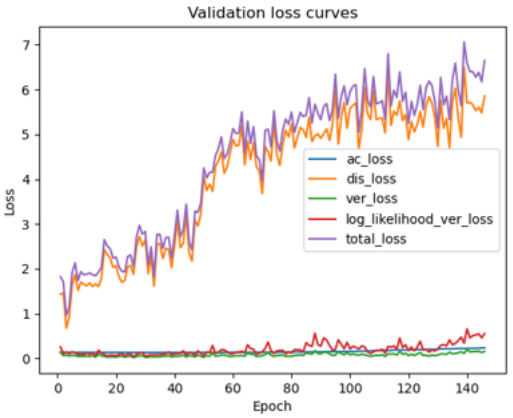
(g)



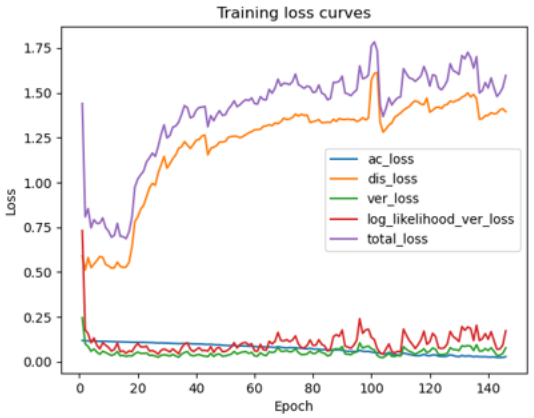
(h)



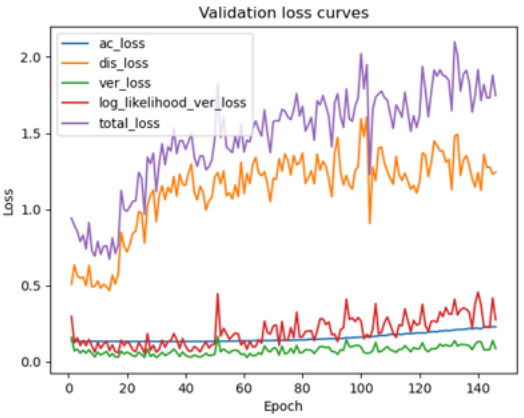
(i)



(j)

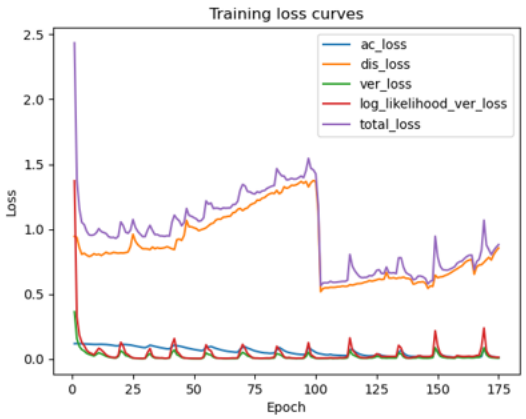


(k)

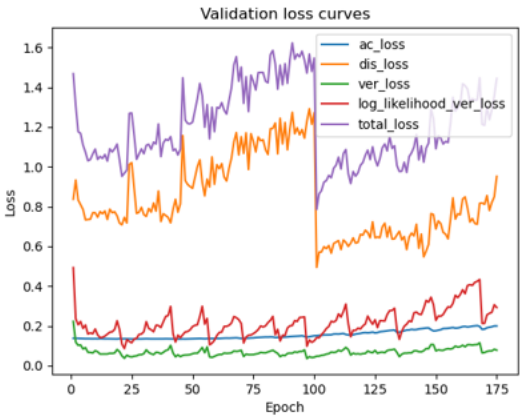


(l)

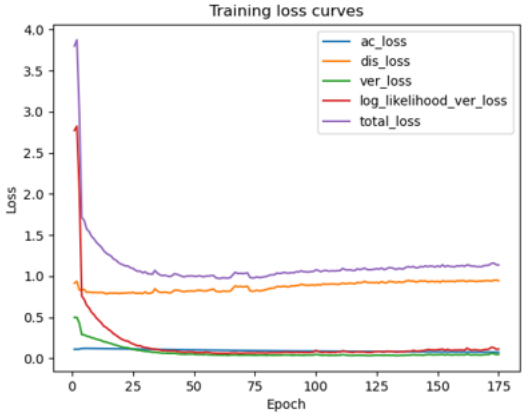
Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 6 to 8 in Table 6.3.



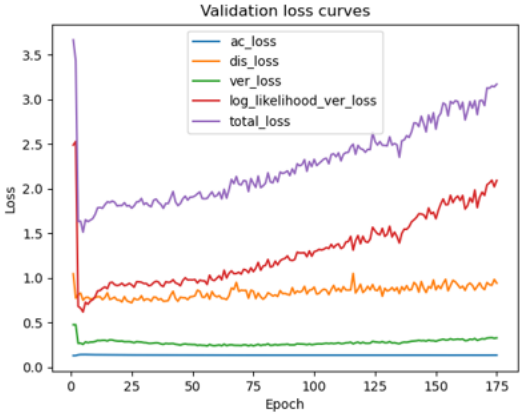
(m)



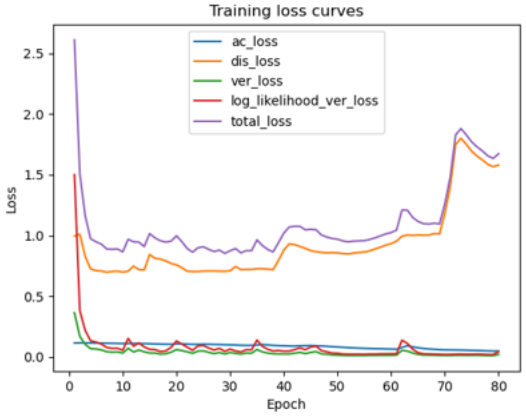
(n)



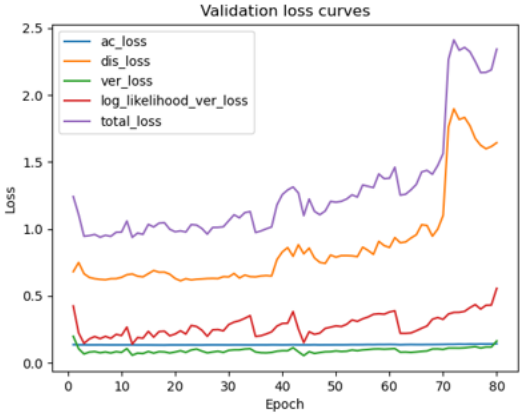
(o)



(p)

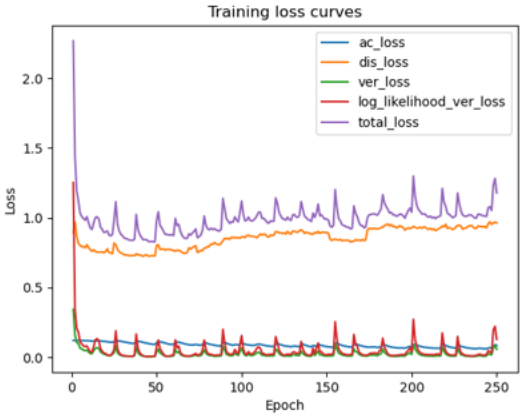


(q)

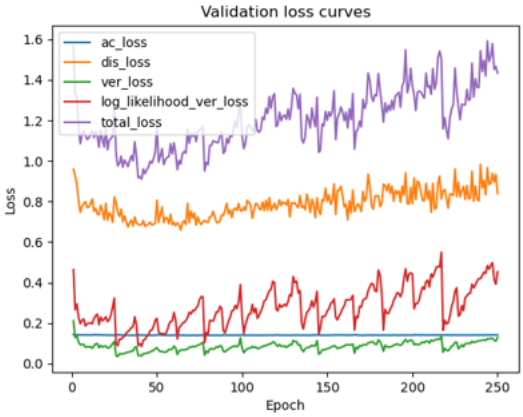


(r)

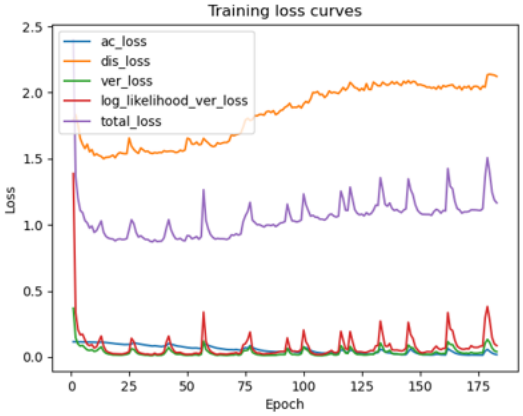
Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 9 to 11 in Table 6.3.



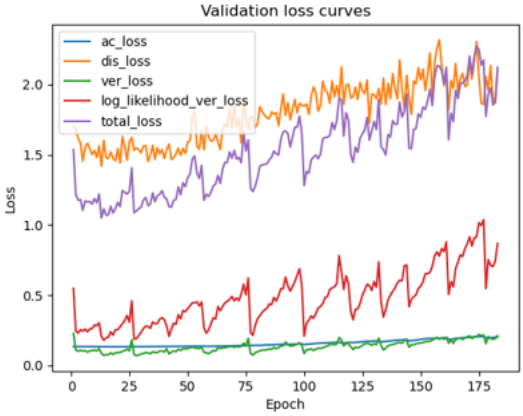
(s)



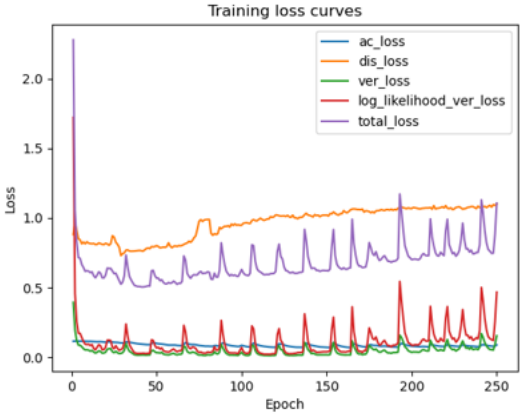
(t)



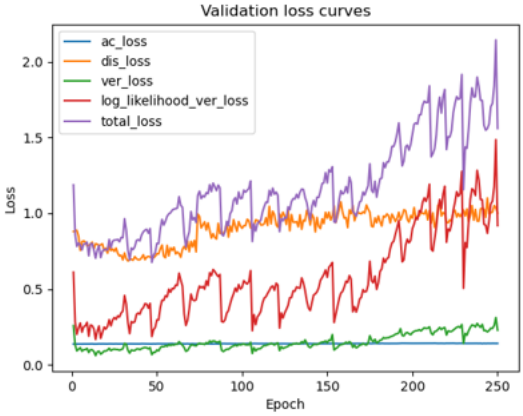
(u)



(v)

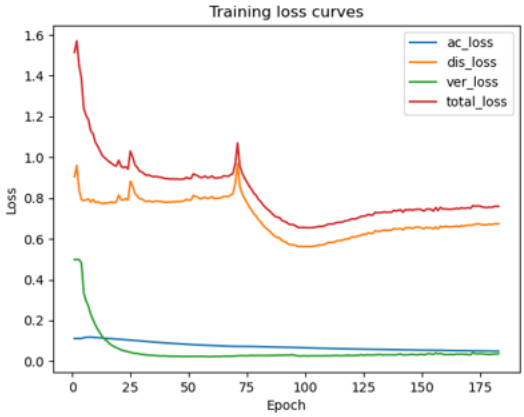


(w)

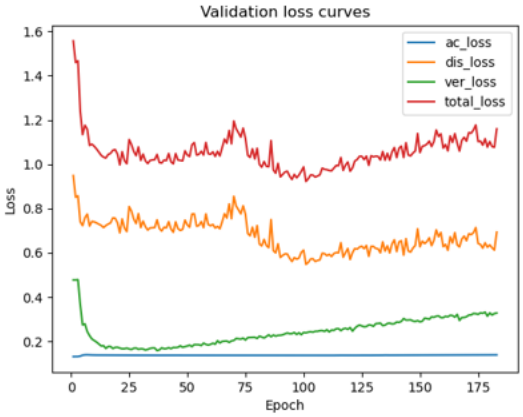


(x)

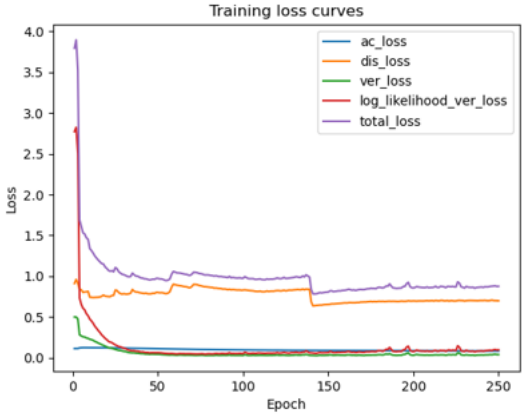
Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 12 to 14 in Table 6.3.



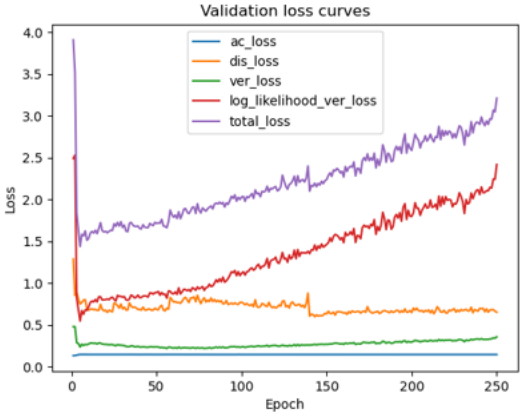
(y)



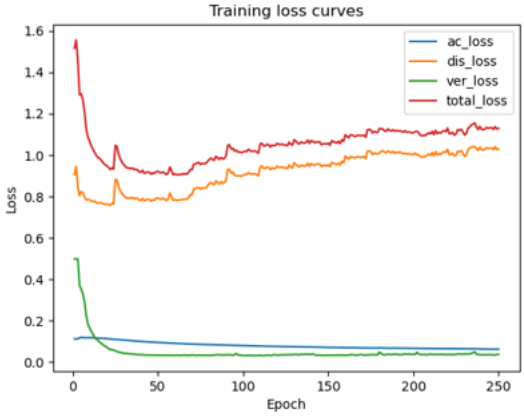
(z)



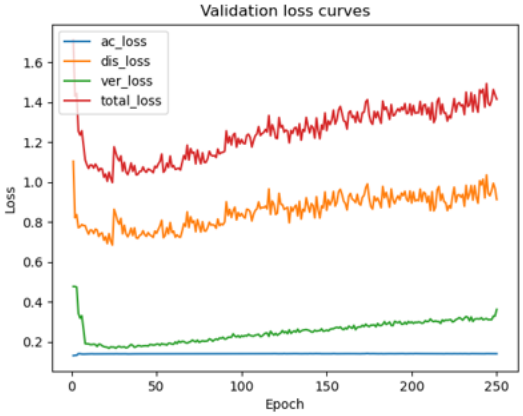
(aa)



(ab)

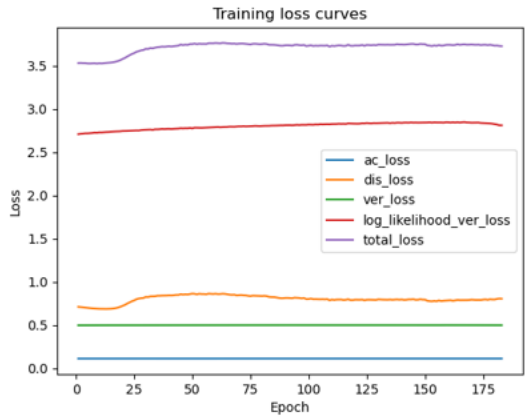


(ac)

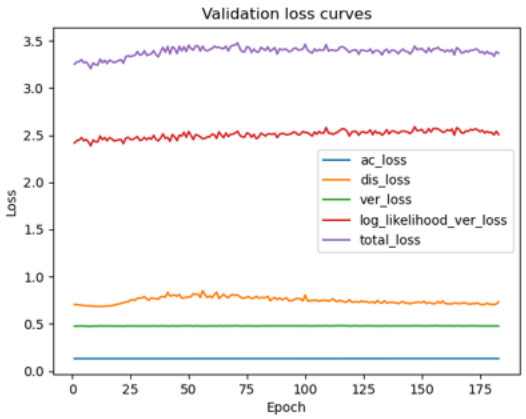


(ad)

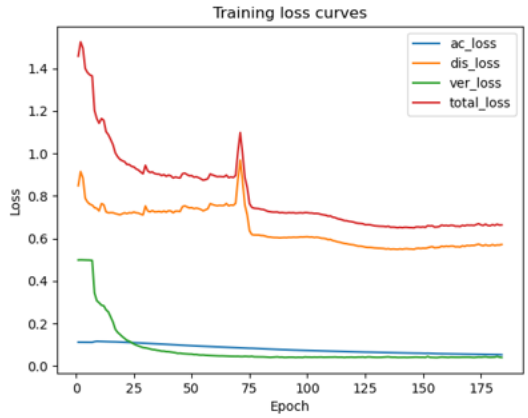
Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 15 to 17 in Table 6.3.



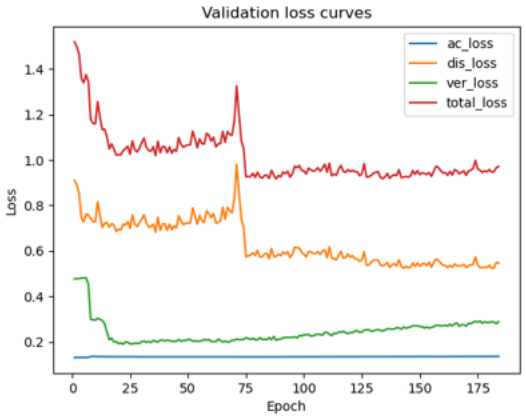
(ae)



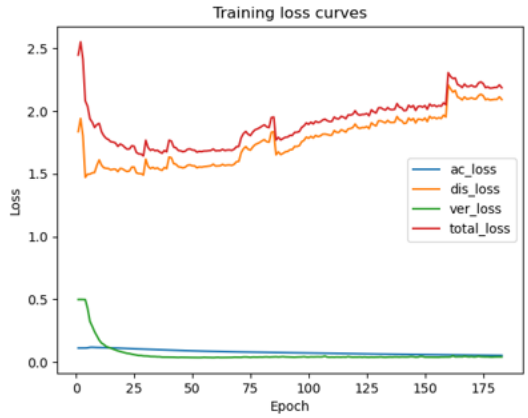
(af)



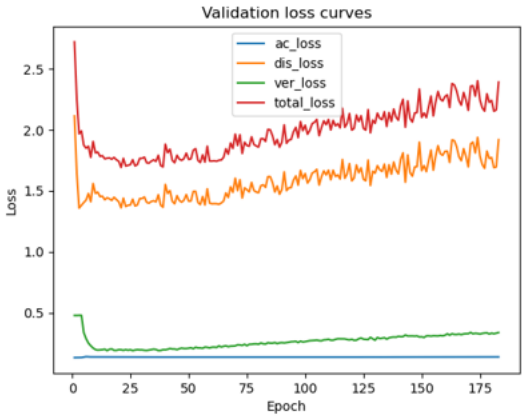
(ag)



(ah)

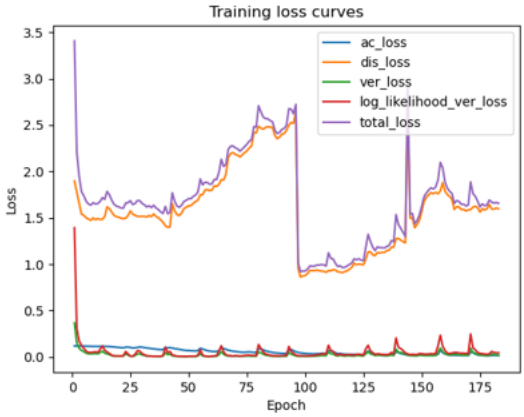


(ai)

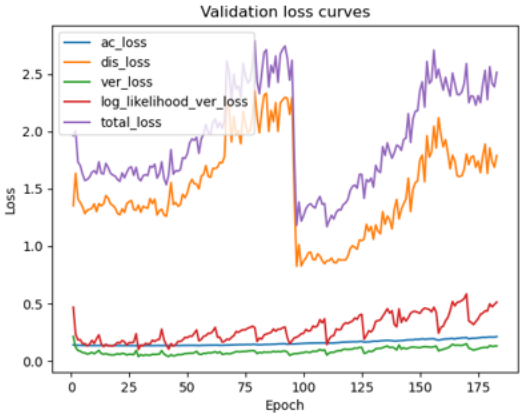


(aj)

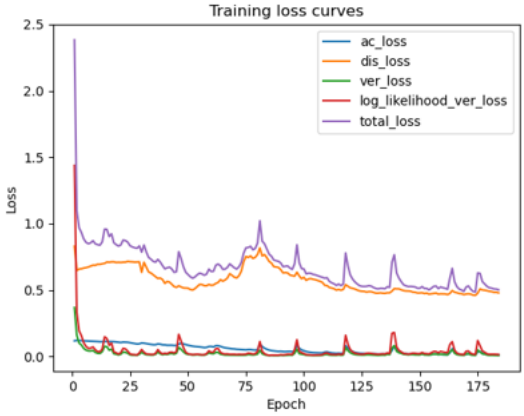
Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 18 to 20 in Table 6.3.



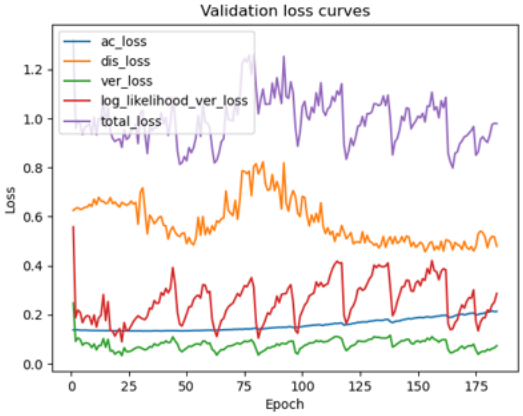
(ak)



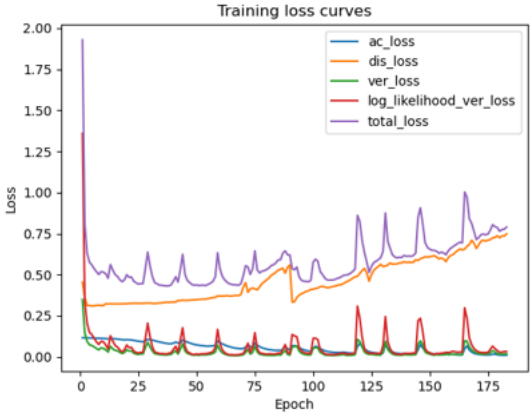
(al)



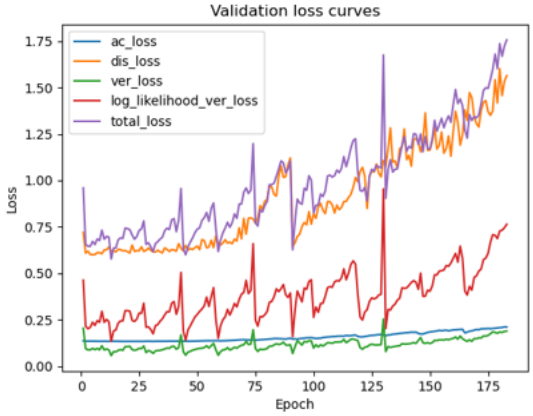
(am)



(an)

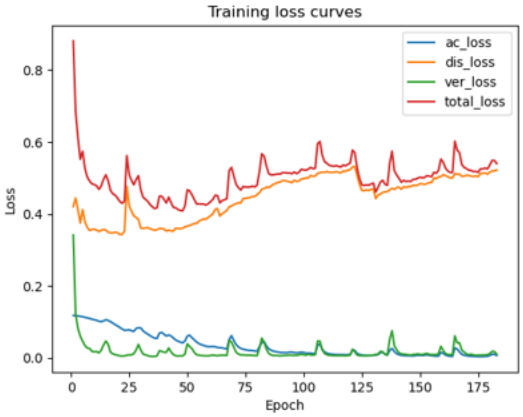


(ao)

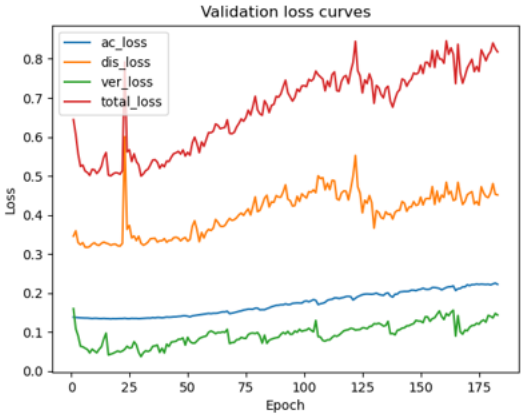


(ap)

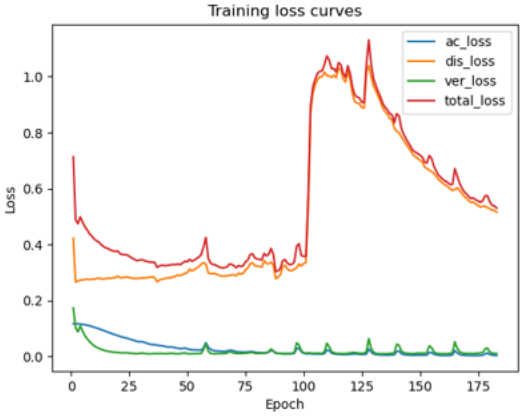
Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 21 to 23 in Table 6.3.



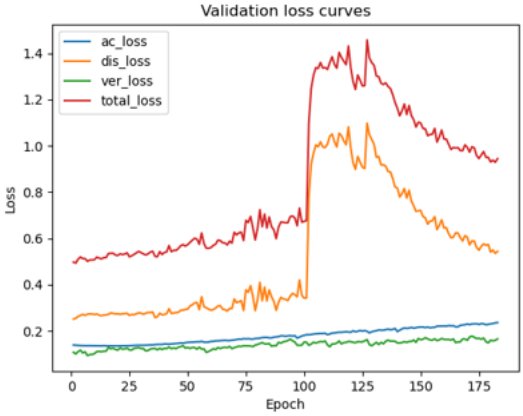
(aq)



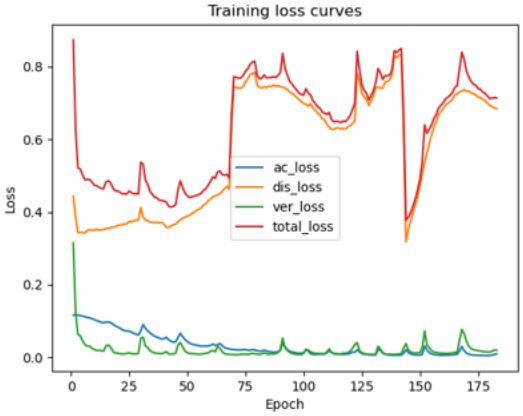
(ar)



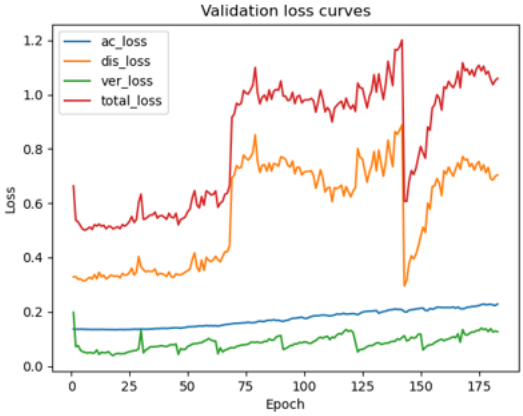
(as)



(at)

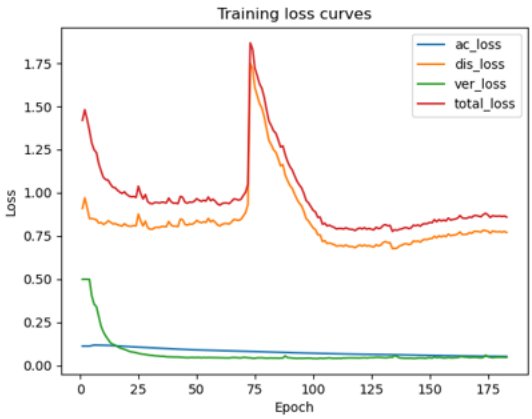


(au)

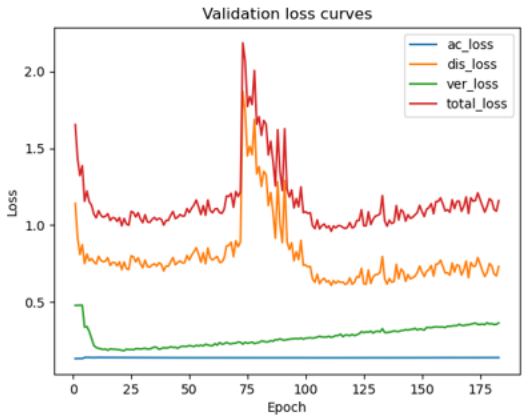


(av)

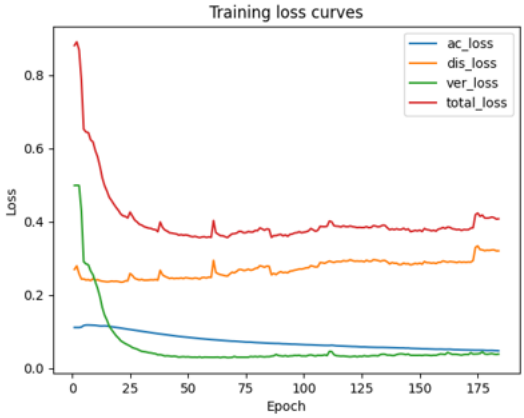
Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 24 to 26 in Table 6.3.



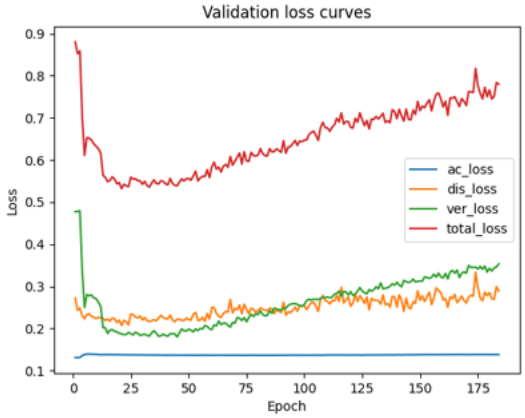
(aw)



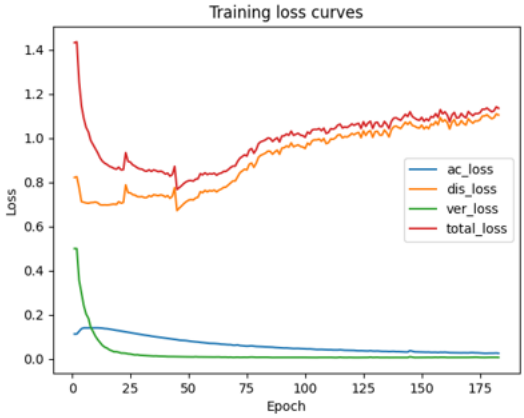
(ax)



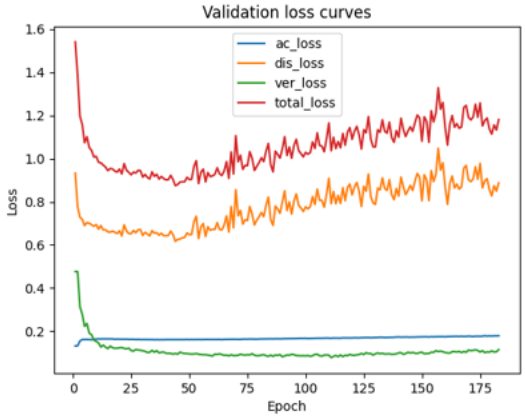
(ay)



(az)

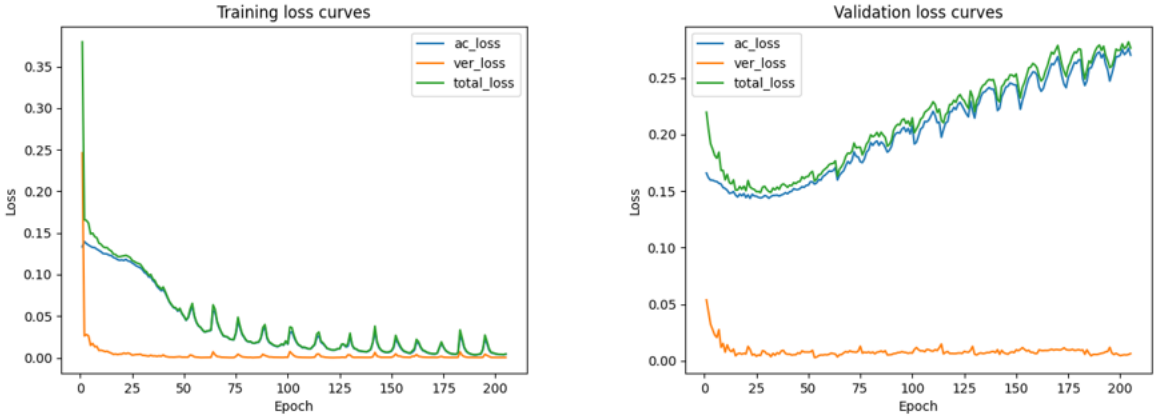


(ba)



(bb)

Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves of experiments from number 27 to 29 in Table 6.3.



(bc) (bd)

Figure A.1: (Continued...) Training and validation loss curves of PriSwin-Dis. Each row of the subfigures shows the loss curves from experiment 30 in Table 6.3.

Patient	Image Pair	Image 1 (anonymized)	Image 2 (Real)	Real Label	Similarity Score (SNN)	SNN Predicted Label
00012021	Same Patient	00012021_013.png	00012021_018.png	1.0	0.695	1.0
		00012021_013.png	00012021_029.png	1.0	0.475	0.0
		00012021_013.png	00012021_039.png	1.0	0.467	0.0
		00012021_013.png	00012021_066.png	1.0	0.456	0.0
		00012021_013.png	00012021_042.png	1.0	0.369	0.0
		00012021_013.png	00012021_037.png	1.0	0.359	0.0
		00012021_013.png	00012021_016.png	1.0	0.352	0.0
		00012021_013.png	00012021_013.png	1.0	0.338	0.0
		00012021_013.png	00012021_028.png	1.0	0.322	0.0
	Different Patient	00012021_013.png	00012973_012.png	0.0	0.849	1.0
		00012021_013.png	00015646_008.png	0.0	0.845	1.0
		00012021_013.png	00012576_013.png	0.0	0.842	1.0
		00012021_013.png	00012294_015.png	0.0	0.838	1.0
		00012021_013.png	00020656_001.png	0.0	0.836	1.0
		00012021_013.png	00014663_000.png	0.0	0.834	1.0
		00012021_013.png	00005532_008.png	0.0	0.833	1.0
		00012021_013.png	00021818_010.png	0.0	0.832	1.0
		00012021_013.png	00001736_032.png	0.0	0.831	1.0
		00012021_013.png	00028190_000.png	0.0	0.829	1.0
00018055	Same Patient	00018055_005.png	00018055_040.png	1.0	0.291	0.0
		00018055_005.png	00018055_046.png	1.0	0.257	0.0
		00018055_005.png	00018055_043.png	1.0	0.249	0.0
		00018055_005.png	00018055_030.png	1.0	0.182	0.0
		00018055_005.png	00018055_005.png	1.0	0.159	0.0
		00018055_005.png	00018055_049.png	1.0	0.145	0.0
		00018055_005.png	00018055_047.png	1.0	0.142	0.0
		00018055_005.png	00018055_044.png	1.0	0.128	0.0
		00018055_005.png	00018055_004.png	1.0	0.126	0.0
		00018055_005.png	00018055_002.png	1.0	0.107	0.0
	Different Patient	00018055_005.png	00029254_000.png	0.0	0.888	1.0
		00018055_005.png	00000948_000.png	0.0	0.888	1.0
		00018055_005.png	00004808_066.png	0.0	0.885	1.0
		00018055_005.png	00016699_000.png	0.0	0.878	1.0
		00018055_005.png	00010871_003.png	0.0	0.873	1.0
		00018055_005.png	00010871_009.png	0.0	0.873	1.0
		00018055_005.png	00019750_023.png	0.0	0.873	1.0
		00018055_005.png	00027726_029.png	0.0	0.870	1.0
		00018055_005.png	00011355_005.png	0.0	0.867	1.0
		00018055_005.png	00019373_002.png	0.0	0.866	1.0

Table A.1: Results of top-10 similarity scores of SNN sorted in descending order. These results are for 12 different patients. The first ten results in each section of a patient in the table are compared with all other images of the same patients, and the next ten are results when compared to the images of different patients.

Patient	Image Pair	Image 1 (anonymized)	Image 2 (Real)	Real Label	Similarity Score (SNN)	SNN Predicted Label
00022416	Same Patient	00022416_007.png	00022416_022.png	1.0	0.485	0.0
		00022416_007.png	00022416_004.png	1.0	0.292	0.0
		00022416_007.png	00022416_034.png	1.0	0.254	0.0
		00022416_007.png	00022416_027.png	1.0	0.207	0.0
		00022416_007.png	00022416_017.png	1.0	0.186	0.0
		00022416_007.png	00022416_064.png	1.0	0.164	0.0
		00022416_007.png	00022416_058.png	1.0	0.151	0.0
		00022416_007.png	00022416_054.png	1.0	0.142	0.0
		00022416_007.png	00022416_070.png	1.0	0.138	0.0
		00022416_007.png	00022416_003.png	1.0	0.131	0.0
	Different Patient	00022416_007.png	00020213_023.png	0.0	0.834	1.0
		00022416_007.png	00020318_008.png	0.0	0.831	1.0
		00022416_007.png	00020318_002.png	0.0	0.830	1.0
		00022416_007.png	00019373_055.png	0.0	0.830	1.0
		00022416_007.png	00004808_078.png	0.0	0.820	1.0
		00022416_007.png	00011402_000.png	0.0	0.816	1.0
		00022416_007.png	00009845_001.png	0.0	0.815	1.0
		00022416_007.png	00011973_012.png	0.0	0.814	1.0
		00022416_007.png	00013625_002.png	0.0	0.813	1.0
		00022416_007.png	00020318_032.png	0.0	0.808	1.0
00000583	Same Patient	00000583_046.png	00000583_012.png	1.0	0.805	1.0
		00000583_046.png	00000583_008.png	1.0	0.746	1.0
		00000583_046.png	00000583_040.png	1.0	0.737	1.0
		00000583_046.png	00000583_018.png	1.0	0.710	1.0
		00000583_046.png	00000583_011.png	1.0	0.699	1.0
		00000583_046.png	00000583_010.png	1.0	0.639	1.0
		00000583_046.png	00000583_036.png	1.0	0.624	1.0
		00000583_046.png	00000583_028.png	1.0	0.615	1.0
		00000583_046.png	00000583_042.png	1.0	0.596	1.0
		00000583_046.png	00000583_007.png	1.0	0.587	1.0
	Different Patient	00000583_046.png	00004808_012.png	0.0	0.841	1.0
		00000583_046.png	00007569_012.png	0.0	0.834	1.0
		00000583_046.png	00007569_014.png	0.0	0.824	1.0
		00000583_046.png	00004808_035.png	0.0	0.823	1.0
		00000583_046.png	00010092_011.png	0.0	0.823	1.0
		00000583_046.png	00004808_027.png	0.0	0.821	1.0
		00000583_046.png	00001736_017.png	0.0	0.819	1.0
		00000583_046.png	00006821_000.png	0.0	0.816	1.0
		00000583_046.png	00001736_020.png	0.0	0.814	1.0
		00000583_046.png	00004808_014.png	0.0	0.808	1.0
00003064	Same Patient	00003064_042.png	00003064_030.png	1.0	0.711	1.0
		00003064_042.png	00003064_004.png	1.0	0.525	1.0
		00003064_042.png	00003064_045.png	1.0	0.482	0.0
		00003064_042.png	00003064_034.png	1.0	0.433	0.0
		00003064_042.png	00003064_008.png	1.0	0.407	0.0
		00003064_042.png	00003064_035.png	1.0	0.386	0.0
		00003064_042.png	00003064_003.png	1.0	0.342	0.0
		00003064_042.png	00003064_040.png	1.0	0.339	0.0
		00003064_042.png	00003064_012.png	1.0	0.315	0.0
		00003064_042.png	00003064_006.png	1.0	0.303	0.0
	Different Patient	00003064_042.png	00020405_003.png	0.0	0.844	1.0
		00003064_042.png	00016807_009.png	0.0	0.842	1.0
		00003064_042.png	00018496_001.png	0.0	0.840	1.0
		00003064_042.png	00026810_002.png	0.0	0.840	1.0
		00003064_042.png	00005567_023.png	0.0	0.840	1.0
		00003064_042.png	00009892_002.png	0.0	0.837	1.0
		00003064_042.png	00019767_000.png	0.0	0.837	1.0
		00003064_042.png	00020751_009.png	0.0	0.835	1.0
		00003064_042.png	00022572_060.png	0.0	0.833	1.0
		00003064_042.png	00023075_004.png	0.0	0.833	1.0

Table A.1: (Continued...) Results of top-10 similarity scores of SNN sorted in descending order. These results are for 12 different patients. The first ten results in each section of a patient in the table are compared with all other images of the same patients, and the next ten are results when compared to the images of different patients.

Patient	Image Pair	Image 1 (anonymized)	Image 2 (Real)	Real Label	Similarity Score (SNN)	SNN Predicted Label
00004342	Same Patient	00004342_059.png	00004342_025.png	1.0	0.886	1.0
		00004342_059.png	00004342_013.png	1.0	0.803	1.0
		00004342_059.png	00004342_028.png	1.0	0.765	1.0
		00004342_059.png	00004342_004.png	1.0	0.764	1.0
		00004342_059.png	00004342_019.png	1.0	0.737	1.0
		00004342_059.png	00004342_024.png	1.0	0.723	1.0
		00004342_059.png	00004342_021.png	1.0	0.661	1.0
		00004342_059.png	00004342_018.png	1.0	0.633	1.0
		00004342_059.png	00004342_055.png	1.0	0.583	1.0
		00004342_059.png	00004342_058.png	1.0	0.565	1.0
	Different Patient	00004342_059.png	00020945_050.png	0.0	0.890	1.0
		00004342_059.png	00016191_011.png	0.0	0.890	1.0
		00004342_059.png	00028139_000.png	0.0	0.888	1.0
		00004342_059.png	00004893_083.png	0.0	0.886	1.0
		00004342_059.png	00019767_011.png	0.0	0.886	1.0
		00004342_059.png	00020405_012.png	0.0	0.886	1.0
		00004342_059.png	00019919_000.png	0.0	0.884	1.0
		00004342_059.png	00023075_016.png	0.0	0.883	1.0
		00004342_059.png	00020405_015.png	0.0	0.881	1.0
00005066	Same Patient	00005066_018.png	00005066_012.png	1.0	0.242	0.0
		00005066_018.png	00005066_010.png	1.0	0.183	0.0
		00005066_018.png	00005066_000.png	1.0	0.181	0.0
		00005066_018.png	00005066_055.png	1.0	0.147	0.0
		00005066_018.png	00005066_057.png	1.0	0.124	0.0
		00005066_018.png	00005066_030.png	1.0	0.117	0.0
		00005066_018.png	00005066_013.png	1.0	0.111	0.0
		00005066_018.png	00005066_041.png	1.0	0.086	0.0
		00005066_018.png	00005066_003.png	1.0	0.084	0.0
		00005066_018.png	00005066_021.png	1.0	0.078	0.0
	Different Patient	00005066_018.png	00008554_010.png	0.0	0.891	1.0
		00005066_018.png	00009574_024.png	0.0	0.870	1.0
		00005066_018.png	00022572_089.png	0.0	0.867	1.0
		00005066_018.png	00014525_023.png	0.0	0.862	1.0
		00005066_018.png	00011583_020.png	0.0	0.860	1.0
		00005066_018.png	00009574_019.png	0.0	0.850	1.0
		00005066_018.png	00022572_042.png	0.0	0.844	1.0
		00005066_018.png	00014058_006.png	0.0	0.844	1.0
		00005066_018.png	00011355_049.png	0.0	0.842	1.0
		00005066_018.png	00013625_016.png	0.0	0.840	1.0
00005681	Same Patient	00005681_039.png	00005681_046.png	1.0	0.721	1.0
		00005681_039.png	00005681_014.png	1.0	0.702	1.0
		00005681_039.png	00005681_047.png	1.0	0.701	1.0
		00005681_039.png	00005681_015.png	1.0	0.626	1.0
		00005681_039.png	00005681_054.png	1.0	0.615	1.0
		00005681_039.png	00005681_004.png	1.0	0.564	1.0
		00005681_039.png	00005681_030.png	1.0	0.562	1.0
		00005681_039.png	00005681_024.png	1.0	0.443	0.0
		00005681_039.png	00005681_034.png	1.0	0.430	0.0
		00005681_039.png	00005681_031.png	1.0	0.400	0.0
	Different Patient	00005681_039.png	00022899_022.png	0.0	0.848	1.0
		00005681_039.png	00027213_057.png	0.0	0.846	1.0
		00005681_039.png	00021670_021.png	0.0	0.839	1.0
		00005681_039.png	00006808_035.png	0.0	0.839	1.0
		00005681_039.png	00013627_002.png	0.0	0.838	1.0
		00005681_039.png	00027213_050.png	0.0	0.838	1.0
		00005681_039.png	00014251_010.png	0.0	0.834	1.0
		00005681_039.png	00025612_009.png	0.0	0.834	1.0
		00005681_039.png	00009237_021.png	0.0	0.830	1.0
		00005681_039.png	00013670_000.png	0.0	0.829	1.0

Table A.1: (Continued...) Results of top-10 similarity scores of SNN sorted in descending order. These results are for 12 different patients. The first ten results in each section of a patient in the table are compared with all other images of the same patients, and the next ten are results when compared to the images of different patients.

Patient	Image Pair	Image 1 (anonymized)	Image 2 (Real)	Real Label	Similarity Score (SNN)	SNN Predicted Label
00006304	Same Patient	00006304_039.png	00006304_005.png	1.0	0.364	0.0
		00006304_039.png	00006304_000.png	1.0	0.294	0.0
		00006304_039.png	00006304_009.png	1.0	0.285	0.0
		00006304_039.png	00006304_013.png	1.0	0.190	0.0
		00006304_039.png	00006304_002.png	1.0	0.176	0.0
		00006304_039.png	00006304_008.png	1.0	0.114	0.0
		00006304_039.png	00006304_015.png	1.0	0.102	0.0
		00006304_039.png	00006304_058.png	1.0	0.095	0.0
		00006304_039.png	00006304_059.png	1.0	0.079	0.0
		00006304_039.png	00006304_010.png	1.0	0.062	0.0
	Different Patient	00006304_039.png	00014525_023.png	0.0	0.895	1.0
		00006304_039.png	00022572_087.png	0.0	0.892	1.0
		00006304_039.png	00017893_001.png	0.0	0.892	1.0
		00006304_039.png	00019373_014.png	0.0	0.881	1.0
		00006304_039.png	00013750_014.png	0.0	0.878	1.0
		00006304_039.png	00019373_053.png	0.0	0.875	1.0
		00006304_039.png	00008508_001.png	0.0	0.874	1.0
		00006304_039.png	00020318_025.png	0.0	0.872	1.0
		00006304_039.png	00013625_016.png	0.0	0.869	1.0
		00006304_039.png	00014996_011.png	0.0	0.865	1.0
00000181	Same Patient	00000181_049.png	00000181_064.png	1.0	0.553	1.0
		00000181_049.png	00000181_033.png	1.0	0.362	0.0
		00000181_049.png	00000181_003.png	1.0	0.314	0.0
		00000181_049.png	00000181_019.png	1.0	0.270	0.0
		00000181_049.png	00000181_060.png	1.0	0.242	0.0
		00000181_049.png	00000181_050.png	1.0	0.202	0.0
		00000181_049.png	00000181_021.png	1.0	0.117	0.0
		00000181_049.png	00000181_058.png	1.0	0.089	0.0
		00000181_049.png	00000181_022.png	1.0	0.087	0.0
		00000181_049.png	00000181_044.png	1.0	0.074	0.0
	Different Patient	00000181_049.png	00013625_016.png	0.0	0.844	1.0
		00000181_049.png	00014525_023.png	0.0	0.829	1.0
		00000181_049.png	00020318_029.png	0.0	0.827	1.0
		00000181_049.png	00011402_002.png	0.0	0.825	1.0
		00000181_049.png	00019238_000.png	0.0	0.818	1.0
		00000181_049.png	00019750_031.png	0.0	0.812	1.0
		00000181_049.png	00013625_022.png	0.0	0.812	1.0
		00000181_049.png	00017236_041.png	0.0	0.812	1.0
		00000181_049.png	00019495_002.png	0.0	0.807	1.0
		00000181_049.png	00008554_010.png	0.0	0.806	1.0
00011355	Same Patient	00011355_057.png	00011355_005.png	1.0	0.843	1.0
		00011355_057.png	00011355_016.png	1.0	0.783	1.0
		00011355_057.png	00011355_044.png	1.0	0.766	1.0
		00011355_057.png	00011355_013.png	1.0	0.755	1.0
		00011355_057.png	00011355_025.png	1.0	0.749	1.0
		00011355_057.png	00011355_022.png	1.0	0.718	1.0
		00011355_057.png	00011355_028.png	1.0	0.714	1.0
		00011355_057.png	00011355_052.png	1.0	0.714	1.0
		00011355_057.png	00011355_029.png	1.0	0.669	1.0
		00011355_057.png	00011355_020.png	1.0	0.660	1.0
	Different Patient	00011355_057.png	00001247_002.png	0.0	0.884	1.0
		00011355_057.png	00017236_000.png	0.0	0.878	1.0
		00011355_057.png	00013089_000.png	0.0	0.876	1.0
		00011355_057.png	00011583_009.png	0.0	0.876	1.0
		00011355_057.png	00010871_003.png	0.0	0.874	1.0
		00011355_057.png	00007557_025.png	0.0	0.872	1.0
		00011355_057.png	00029051_000.png	0.0	0.872	1.0
		00011355_057.png	00014116_001.png	0.0	0.871	1.0
		00011355_057.png	00012609_001.png	0.0	0.867	1.0
		00011355_057.png	00020318_023.png	0.0	0.867	1.0

Table A.1: (Continued...) Results of top-10 similarity scores of SNN sorted in descending order. These results are for 12 different patients. The first ten results in each section of a patient in the table are compared with all other images of the same patients, and the next ten are results when compared to the images of different patients.

List of Abbreviations

AC Auxiliary Classifier 83, 84, 89, 90

AdaGrad Adaptive Gradient 26

ADAM Adaptive Moment Estimation 26

ADS Anonymization through Data Synthesis 46, 47, 127

AI Artificial Intelligence 16, 17

ANN Artificial Neural Network 19–23, 25

AUC Area Under the Curve 50, 61, 66, 68, 74–77, 81, 84, 89, 90, 99, 100, 102–104, 129

AUROC Area Under the ROC Curve 75, 128

BCE Binary Cross Entropy 27, 50, 61, 62, 88

CAD Computer Aided Diagnostics 16, 17, 53

CNN Convolutional Neural Network 5, 18, 19, 28, 46, 47, 127

CoAD Coronary Artery Disease 14

ConvNet Convolutional Neural Network 27

COPD Chronic Obstructive Pulmonary Disease 14

CT Computed Tomography 1, 7, 11

CV Computer Vision 16, 34

CXR Chest X-ray Radiography 1–3, 5

DC Auxiliary Discriminator 83, 84, 87–89

DL Deep Learning 3, 16, 18, 19, 26, 30, 33, 36, 53, 63, 65, 81

DP Differential Privacy 40, 43, 44, 46

DP-Pix Differential Privacy - Pixelization 3, 44, 45, 127

DP-SGD Differential Privacy - Stochastic Gradient Descent 44

DP-VAE Differential Privacy - Variational Auto-Encoders 44

EU European Union 2

FC Fully-Connected 27–29, 35

GAN Generative Adversarial Network 29, 30, 40, 46, 47, 127

GDP Global Differential Privacy 43

GDPR General Data Protection Regulation 2, 39

GLR Global Learning Rate 86, 87

GPU Graphical Processing Unit 65, 70, 83, 87, 88

HIPAA Health Insurance Portability and Accountability Act 2

LDM Latent Diffusion Model 106

LDP Local Differential Privacy 43

LLR Local Learning Rate 86

LR Learning Rate 29

MAE Mean Absolute Error 27

MIT Medical Imaging Technique 1, 39

ML Machine Learning 16–19, 36

MLP Multi Layer Perceptron 61, 74

MRI Magnetic Resonance Imaging 1, 7, 8

MSE Mean Squared Error 27

NHR@FAU Erlangen National High Performance Computing Center 65

NHS National Health Service 1, 2, 127

NIH National Institutes of Health 53

NLP Natural Language Processing 26, 53

NN Neural Network 19, 26, 27, 30, 32, 33, 35–37, 44, 45, 49

PET Positron Emission Tomography 1, 7

PLCO Prostate, Lung, Colorectal, and Ovarian Cancer 54

ReLU Rectified Linear Unit 23–25, 28, 33

ResNet Residual Neural Network 31, 33–36, 50, 127

SGD Stochastic Gradient Descent 26, 71

SNN Siamese Neural Network 3, 32, 35, 36, 47, 50, 61, 62, 69, 80, 81, 83, 91–93, 96, 97, 101–103, 118–121, 127, 129

SPECT Single Photon Emission Computed Tomography 1, 7

Swin-T Swin Transformer 34–36, 60, 61, 68, 69, 73, 76, 80, 81, 100, 103, 104

t-SNE t-distributed Stochastic Neighbor Embedding 92, 93, 95, 96, 101

USA United States of America 2

ViT Vision Transformer 34

VR Auxiliary Verification Network 83, 84, 87, 89

List of Figures

List of Tables

List of Algorithms

Bibliography

[Abr05] A. Abraham. Artificial neural networks. *Handbook of measuring system design*, 2005 (cited on pp. 19–23).

[Ach+03] R. Acharya, P. S. Bhat, S. Kumar, and L. C. Min. Transmission and storage of medical images with patient information. *Computers in Biology and Medicine*, 33(4):303–310, 2003 (cited on p. 15).

[Ada+89] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989 (cited on p. 43).

[Aer+14] H. J. Aerts, E. R. Velazquez, R. T. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nature communications*, 5(1):4006, 2014 (cited on p. 16).

[Aki+12] O. Akin, S. B. Brennan, D. D. Dershaw, M. S. Ginsberg, M. J. Gollub, H. Schöder, D. M. Panicek, and H. Hricak. Advances in oncologic imaging: update on 5 common cancers. *CA: a cancer journal for clinicians*, 62(6):364–393, 2012 (cited on p. 19).

[Aks+16] A. Akselrod-Ballin, L. Karlinsky, S. Alpert, S. Hasoul, R. Ben-Ari, and E. Barkan. A region based convolutional network for tumor detection and classification in breast mammography. In *Deep Learning and Data Labeling for Medical Applications: First International Workshop, LABELS 2016, and Second International Workshop, DLMIA 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 21, 2016, Proceedings 2*, pages 197–205. Springer, 2016 (cited on p. 1).

[Ali+16] O. Ali and A. Ouda. A classification module in data masking framework for business intelligence platform in healthcare. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–8. IEEE, 2016 (cited on p. 17).

[Ali23] A. Ali. A perfect guide to understand encoder decoders in depth with visuals. 2023. URL: https://medium.com/@ahmadsabry678/a-perfect-guide-to-understand-encoder-decoders-in-depth-with-visuals-30805c23659b. accessed: 01.05.2024 (cited on p. 31).

[Ana+12] M. Analoui, J. D. Bronzino, and D. R. Peterson. *Medical imaging: principles and practices*. CRC Press, 2012 (cited on p. 7).

[Ana+21] A. Anaya-Isaza, L. Mera-Jiménez, and M. Zequera-Diaz. An overview of deep learning in medical imaging. *Informatics in medicine unlocked*, 26:100723, 2021 (cited on p. 18).

[And12] G. Andriole. Prostate cancer screening in the randomized prostate, lung, colorectal, and ovarian cancer screening trial: mortality results after 13years of follow-up: andriole gl, for the plco project team (washington univ school of medicine, st louis, mo; et al) j natl cancer inst 104: 125-132, 2012 §. *Yearbook of Urology*, 2012:39–40, 2012 (cited on p. 54).

[Anw+18] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018 (cited on p. 28).

[Bae+19] H. Bae, D. Jung, H.-S. Choi, and S. Yoon. Anomigan: generative adversarial networks for anonymizing private medical data. In *Pacific Symposium on Biocomputing 2020*, pages 563–574. World Scientific, 2019 (cited on p. 46).

[Bas+00] I. A. Basheer and M. Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1):3–31, 2000 (cited on p. 21).

[Bel+22] A. G. Belyaev and P.-A. Fayolle. Black-box image deblurring and defiltering. *Signal Processing: Image Communication*, 108:116833, 2022 (cited on p. 42).

[Ben22] S. Benhur. A Friendly Introduction to Siamese Networks | Built In — builtin.com. https://builtin.com/machine-learning/siamese-network, 2022. [Accessed 19-06-2024] (cited on p. 36).

[Bis95] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995 (cited on p. 33).

[Bra08] W. G. Bradley. History of medical imaging. *Proceedings of the American Philosophical Society*, 152(3):349–361, 2008 (cited on p. 7).

[Bro+93] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using aßiamesetime delay neural network. *Advances in neural information processing systems*, 6, 1993 (cited on p. 35).

[Bud+19] M. Buda, A. Saha, and M. A. Mazurowski. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. *Computers in biology and medicine*, 109:218–225, 2019 (cited on p. 60).

[Bus+20] A. Bustos, A. Pertusa, J.-M. Salinas, and M. De La Iglesia-Vaya. Padchest: a large chest x-ray image dataset with multi-label annotated reports. *Medical image analysis*, 66:101797, 2020 (cited on p. 54).

[Cae17] O. Caelen. A bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, 81(3):429–450, 2017 (cited on pp. 66, 67).

[Cao+03] F. Cao, H. K. Huang, and X. Zhou. Medical image security in a hipaa mandated pacs environment. *Computerized medical imaging and graphics*, 27(2-3):185–196, 2003 (cited on p. 15).

[Cho+14] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014 (cited on p. 30).

[Cho+18] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018 (cited on p. 46).

[Coa+00] G. Coatrieux, H. Maitre, B. Sankur, Y. Rolland, and R. Collorec. Relevance of watermarking in medical imaging. In *Proceedings 2000 IEEE EMBS international conference on information technology applications in biomedicine. ITAB-ITIS 2000. Joint Meeting Third IEEE EMBS international conference on information technol*, pages 250–255. IEEE, 2000 (cited on p. 15).

[Col18] M. Collins. Computational graphs, and backpropagation. *Lecture Notes, Columbia University*:11–23, 2018 (cited on p. 63).

[De 16] M. De Bruijne. Machine learning approaches in medical image analysis: from detection to diagnosis, 2016 (cited on p. 17).

[Don+22] J. Dong, A. Roth, and W. J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(1):3–37, 2022 (cited on p. 43).

[Duc+13] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th annual symposium on foundations of computer science*, pages 429–438. IEEE, 2013 (cited on p. 43).

[Dwo+06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006 (cited on pp. 3, 4, 43, 44).

[Dwo08] C. Dwork. Differential privacy: a survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008 (cited on pp. 3, 43).

[Eur16] E. Europa. Eur-lex-32016r0679-en-eur-lex, 2016 (cited on p. 39).

[Fan18] L. Fan. Image pixelization with differential privacy. In *Data and Applications Security and Privacy XXXII: 32nd Annual IFIP WG 11.3 Conference, DBSec 2018, Bergamo, Italy, July 16–18, 2018, Proceedings 32*, pages 148–162. Springer, 2018 (cited on pp. 3, 42–44).

[Fan19] L. Fan. Differential privacy for image publication. In *Theory and Practice of Differential Privacy (TPDP) Workshop*, volume 1 of number 2, page 6, 2019 (cited on pp. 3, 44).

[Fon+20] A. Fontanella, E. Pead, T. MacGillivray, M. O. Bernabeu, and A. Storkey. Classification with a domain shift in medical imaging. In *Medical Imaging Meets NeurIPS Workshop*, 2020 (cited on p. 81).

[Fro23] Frontiers Media SA. Emerging advances in exploiting pulmonary administration for treatment of thoracic diseases. 2023. URL: https://www.frontiersin.org/research-topics/45436/emerging-advances-in-exploiting-pulmonary-administration-for-treatment-of-thoracic-diseases/overview. accessed: 01.05.2024 (cited on p. 14).

[Fu+00] M. S. Fu and O. C. Au. Data hiding by smart pair toggling for halftone images. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 4, pages 2318–2321. IEEE, 2000 (cited on p. 41).

[Gao22] Z. Gao. *NEAT-based Multiclass Classification with Class Binarization*. PhD thesis, June 2022. DOI: 10.13140/RG.2.2.15066.41922 (cited on p. 37).

[Gaz+06] T. Gaziano, K. S. Reddy, F. Paccaud, S. Horton, and V. Chaturvedi. Cardiovascular disease. *Disease Control Priorities in Developing Countries. 2nd edition*, 2006 (cited on p. 14).

[Gel+05] J. Geleijns and J. Wondergem. X-ray imaging and the skin: radiation biology, patient dosimetry and observed effects. *Radiation protection dosimetry*, 114(1-3):121–125, 2005 (cited on p. 14).

[Gha03] Z. Ghahramani. Unsupervised learning. In *Summer school on machine learning*, pages 72–112. Springer, 2003 (cited on p. 18).

[Goo+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014 (cited on p. 46).

[Gu+18] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. Recent advances in convolutional neural networks. *Pattern recognition*, 77:354–377, 2018 (cited on p. 28).

[Gue+19a] S. Guendel, F. C. Ghesu, S. Grbic, E. Gibson, B. Georgescu, A. Maier, and D. Comaniciu. Multi-task learning for chest x-ray abnormality classification on noisy labels. *arXiv preprint arXiv:1905.06362*, 2019 (cited on p. 1).

[Gue+19b] S. Guendel, S. Grbic, B. Georgescu, S. Liu, A. Maier, and D. Comaniciu. Learning to recognize abnormalities in chest x-rays with location-aware dense networks. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain,*

November 19-22, 2018, Proceedings 23, pages 757–765. Springer, 2019 (cited on p. 1).

[Har+21] D. Harrison, F. C. De Leo, W. J. Gallin, F. Mir, S. Marini, and S. P. Leys. Machine learning applications of convolutional neural networks and unet architecture to predict and classify demosponge behavior. *Water*, 13(18):2512, 2021 (cited on p. 32).

[Has+09] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani, and J. Friedman. Overview of supervised learning. *The elements of statistical learning: Data mining, inference, and prediction*:9–41, 2009 (cited on p. 18).

[He+16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016 (cited on pp. 28, 33).

[Hel+24] F. Hellmann, S. Mertes, M. Benouis, A. Hustinx, T.-C. Hsieh, C. Conati, P. Krawitz, and E. André. Ganonymization: a gan-based face anonymization framework for preserving emotional expressions. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024 (cited on p. 46).

[Ho+19] Y. Ho and S. Wookey. The real-world-weight cross-entropy loss function: modeling the costs of mislabeling. *IEEE access*, 8:4806–4813, 2019 (cited on p. 27).

[Hos+15] M. Hossin and M. N. Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015 (cited on p. 66).

[Hua+17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017 (cited on pp. 5, 50, 60).

[Hug22] A. Hughes. Swin Transformer supports 3-billion-parameter vision models that can train with higher-resolution images for greater task applicability - Microsoft Research — microsoft.com. https://www.microsoft.com/en-us/research/blog/swin-transformer-supports-3-billion-parameter-vision-models-that-can-train-with-higher-resolution-images-for-greater-task-applicability/, 2022. [Accessed 19-06-2024] (cited on p. 34).

[Ike+21] A. V. Ikechukwu, S. Murali, R. Deepu, and R. Shivamurthy. Resnet-50 vs vgg-19 vs training from scratch: a comparative analysis of the segmentation and classification of pneumonia from chest x-ray images. *Global Transitions Proceedings*, 2(2):375–381, 2021 (cited on pp. 33, 34).

[Irv+19] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, et al. Chexpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33 of number 01, pages 590–597, 2019 (cited on p. 54).

[Iso+17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017 (cited on p. 46).

[Jac+20] A. Jacobi, M. Chung, A. Bernheim, and C. Eber. Portable chest x-ray in coronavirus disease-19 (covid-19): a pictorial review. *Clinical imaging*, 64:35–42, 2020 (cited on p. 1).

[Jan+13] S. Jana, A. Narayanan, and V. Shmatikov. A scanner darkly: protecting user privacy from perceptual applications. In *2013 IEEE symposium on security and privacy*, pages 349–363. IEEE, 2013 (cited on p. 43).

[Ji+19] Q. Ji, J. Huang, W. He, and Y. Sun. Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. *Algorithms*, 12(3):51, 2019 (cited on p. 34).

[Jin+12] C. Jin, L.-l. Zhou, and X. Wang. Digital signature based on dicom standards of medical image by java (dsmi). *International Journal of Computer and Communication Engineering*, 1(4):454, 2012 (cited on p. 15).

[Joh+19] A. E. Johnson, T. J. Pollard, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, Y. Peng, Z. Lu, R. G. Mark, S. J. Berkowitz, and S. Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019 (cited on p. 54).

[Kai+20] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020 (cited on pp. 3, 5, 16).

[Kar+12] V. S. Karkhanis and J. M. Joshi. Pleural effusion: diagnosis, treatment, and management. *Open access emergency medicine: OAEM*, 4:31, 2012 (cited on p. 15).

[Kas+15] H. Kasban, M. El-Bendary, and D. Salama. A comparative study of medical imaging techniques. *International Journal of Information Science and Intelligent System*, 4(2):37–58, 2015 (cited on pp. 1, 7).

[Kim+17] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*, pages 1857–1865. PMLR, 2017 (cited on p. 46).

[Kim+18] J. Kim, J. Hong, and H. Park. Prospects of deep learning for medical imaging. *Precision and Future Medicine*, 2(2):37–52, 2018 (cited on p. 18).

[Kim+21] B. N. Kim, J. Dolz, P.-M. Jodoin, and C. Desrosiers. Privacy-net: an adversarial approach for identity-obfuscated segmentation of medical images. *IEEE Transactions on Medical Imaging*, 40(7):1737–1749, 2021 (cited on pp. 46–48).

[Kin+14] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014 (cited on p. 27).

[Kor+13] P. Korshunov, A. Melle, J.-L. Dugelay, and T. Ebrahimi. Framework for objective evaluation of privacy filters. In *Applications of Digital Image Processing XXXVI*, volume 8856, pages 265–276. SPIE, 2013 (cited on p. 41).

[Kri+12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012 (cited on pp. 18, 28).

[Kum+20] A. Kumar, S. Sarkar, and C. Pradhan. Malaria disease detection using cnn technique with sgd, rmsprop and adam optimizers. *Deep learning techniques for biomedical and health informatics*:211–230, 2020 (cited on p. 26).

[Kum24] A. Kumar. Demystifying encoder decoder architecture & neural network. 2024. URL: https://vitalflux.com/encoder-decoder-architecture-neural-network/. accessed: 01.05.2024 (cited on pp. 30, 31).

[Lam+17] P. Lambin, R. T. Leijenaar, T. M. Deist, J. Peerlings, E. E. De Jong, J. Van Timmeren, S. Sanduleanu, R. T. Larue, A. J. Even, A. Jochems, et al. Radiomics: the bridge between medical imaging and personalized medicine. *Nature reviews Clinical oncology*, 14(12):749–762, 2017 (cited on p. 16).

[Lax+18] B. Laxmi Sree and M. Vijaya. A weighted mean square error technique to train deep belief networks for imbalanced data, 2018 (cited on p. 27).

[LeC+02] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002 (cited on p. 57).

[LeC+15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015 (cited on p. 18).

[LeC+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989 (cited on p. 27).

[Lee+17] H. Lee, S. Kim, J. W. Kim, and Y. D. Chung. Utility-preserving anonymization for health data publishing. *BMC medical informatics and decision making*, 17:1–12, 2017 (cited on pp. 16, 17).

[Lef+83] E. Lefons, A. Silvestri, F. Tangorra, et al. An analytic approach to statistical databases. In *VLDB*, pages 260–274, 1983 (cited on p. 43).

[Li+05] M. Li, R. Poovendran, and S. Narayanan. Protecting patient privacy against unauthorized release of medical images in a group communication environment. *Computerized medical imaging and graphics*, 29(5):367–383, 2005 (cited on p. 15).

[Li+21] S. Li, J. Wu, X. Xiao, F. Chao, X. Mao, and R. Ji. Revisiting discriminator in gan compression: a generator-discriminator cooperative compression scheme. *Advances in Neural Information Processing Systems*, 34:28560–28572, 2021 (cited on p. 29).

[Lia+23] D. Liao and C. Hargreaves. Classification of dental teeth x-ray images using a deep learning cnn model, May 2023. DOI: 10.20944/preprints202305.0513.v1 (cited on p. 13).

[Lie+85] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems (TODS)*, 10(3):395–411, 1985 (cited on p. 43).

[Lig02] R. W. Light. Pleural effusion. *New England Journal of Medicine*, 346(25):1971–1977, 2002 (cited on p. 14).

[Lin+16] C. Lin, Z. Song, H. Song, Y. Zhou, Y. Wang, and G. Wu. Differential privacy preserving in big data analytics for connected health. *Journal of medical systems*, 40:1–9, 2016 (cited on p. 17).

[Liu+05] K. Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on knowledge and Data Engineering*, 18(1):92–106, 2005 (cited on p. 43).

[Liu+19] X. Liu and C.-J. Hsieh. Rob-gan: generator, discriminator, and adversarial attacker. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11234–11243, 2019 (cited on p. 29).

[Liu+20] P. Liu, Y. Xu, Q. Jiang, Y. Tang, Y. Guo, L.-e. Wang, and X. Li. Local differential privacy for social network publishing. *Neurocomputing*, 391:273–279, 2020 (cited on p. 43).

[Liu+21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021 (cited on pp. 5, 34, 35, 60, 74).

[Los+10] A. Lostumbo, C. Wanamaker, J. Tsai, K. Suzuki, and A. H. Dachman. Comparison of 2d and 3d views for evaluation of flat lesions in ct colonography. *Academic Radiology*, 17(1):39–47, 2010 (cited on p. 17).

[Lun+18] N. B. Lunsford, K. F. Sapsis, B. Smither, J. Reynolds, B. Wilburn, and T. Fairley. Young women's perceptions regarding communication with healthcare providers about breast cancer, risk, and prevention. *Journal of Women's Health*, 27(2):162–170, 2018 (cited on p. 19).

[Mah20] B. Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9(1):381–386, 2020 (cited on p. 18).

[Mai+18] A. Maier, S. Steidl, V. Christlein, and J. Hornegger. *Medical imaging systems: An introductory guide*. Springer, 2018 (cited on pp. 10–12).

[Max+20] M. Maximov, I. Elezi, and L. Leal-Taixé. Ciagan: conditional identity anonymization generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5447–5456, 2020 (cited on p. 39).

[Muk22] S. Mukherjee. The Annotated ResNet-50 — towardsdatascience.com. https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758, 2022. [Accessed 18-06-2024] (cited on p. 33).

[Mur12] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012 (cited on p. 17).

[Nar+22] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1):291–322, 2022 (cited on p. 25).

[Nat+21] P. Natu, S. Natu, and U. Agrawal. Privacy issues in medical image analysis. In *Data protection and privacy in healthcare*, pages 51–64. CRC Press, 2021 (cited on pp. 16, 17).

[Ngu+22] H. Q. Nguyen, K. Lam, L. T. Le, H. H. Pham, D. Q. Tran, D. B. Nguyen, D. D. Le, C. M. Pham, H. T. Tong, D. H. Dinh, et al. Vindr-cxr: an open dataset of chest x-rays with radiologist's annotations. *Scientific Data*, 9(1):429, 2022 (cited on p. 54).

[NHS24] NHS United Kingdom. Diagnostic imaging dataset statistical release. 2024. URL: https://www.england.nhs.uk/statistics/wp-content/uploads/sites/2/2024/04/Statistical-Release-18th-April-2024-PDF-308KB.pdf. accessed: 01.05.2024 (cited on pp. 1, 2).

[Nix+19] M. Nixon and A. Aguado. *Feature extraction and image processing for computer vision*. Academic press, 2019 (cited on p. 42).

[Osh+15] K. O'shea and R. Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015 (cited on p. 28).

[Pac+22] K. Packhäuser, S. Gündel, N. Münster, C. Syben, V. Christlein, and A. Maier. Deep learning-based patient re-identification is able to exploit the biometric nature of medical chest x-ray data. *Scientific Reports*, 12(1):14851, 2022 (cited on pp. 2, 3, 5, 16, 32, 35, 36, 42, 50, 61, 62, 80).

[Pac+23a] K. Packhäuser, L. Folle, F. Thamm, and A. Maier. Generation of anonymous chest radiographs using latent diffusion models for training thoracic abnormality classification systems. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2023 (cited on pp. 45, 106).

[Pac+23b] K. Packhäuser, S. Gündel, F. Thamm, F. Denzinger, and A. Maier. Deep learning-based anonymization of chest radiographs: a utility-preserving measure for patient privacy. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 262–272. Springer, 2023 (cited on pp. 4, 5, 31, 32, 41, 42, 44, 45, 49–51, 58, 60, 62, 67–69, 74, 76–83, 90, 91, 99, 102).

[Pas+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: an imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019 (cited on p. 65).

[Pul19] E.-M. Pulfer. Different approaches to blurring digital images and their effect on facial detection, 2019 (cited on p. 42).

[Qi+19] J. Qi, J. Du, S. M. Siniscalchi, and C.-H. Lee. A theory on deep neural network based vector-to-vector regression with an illustration of its expressive power in speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):1932–1943, 2019 (cited on p. 26).

[Rad+15] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015 (cited on p. 46).

[Rah+20] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam, R. Mazhar, T. Hamid, M. T. Islam, S. Kashem, Z. B. Mahbub, et al. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *Ieee Access*, 8:191586–191601, 2020 (cited on p. 4).

[Rah+22] A. M. Rahmani, E. Azhir, M. Naserbakht, M. Mohammadi, A. H. M. Aldalwie, M. K. Majeed, S. H. Taher Karim, and M. Hosseinzadeh. Automatic covid-19 detection mechanisms and approaches from medical images: a systematic review. *Multimedia tools and applications*, 81(20):28779–28798, 2022 (cited on p. 4).

[Raj+12] A. Rajkumar and S. Agarwal. A differentially private stochastic gradient descent algorithm for multiparty classification. In *Artificial Intelligence and Statistics*, pages 933–941. PMLR, 2012 (cited on p. 44).

[Raj+17] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, et al. Chexnet: radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017 (cited on pp. 50, 60, 69, 73, 74).

[Rat+18] P. Rathore, R. Rastogi, V. Singh, and S. Soni. Comparative role of x-ray mammography and sonography with sonoelastography in palpable breast lesions. *Annals of International medical and Dental Research*, 4, May 2018. DOI: 10.21276/aimdr.2018.4.3.RD2 (cited on p. 13).

[Rif08] R. Rifkin. Multiclass classification. *Lecture Notes, Spring08. MIT, USA*, 59, 2008 (cited on p. 38).

[Rod+21] M. M. Rodríguez-Hernández, R. E. Pruneda, and J. M. Rodríguez-Díaz. Statistical analysis of the evolutive effects of language development in the resolution of mathematical problems in primary school education. *Mathematics*, 9(10), 2021. ISSN: 2227-7390. DOI: 10.3390/math9101081. URL: https://www.mdpi.com/2227-7390/9/10/1081 (cited on p. 66).

[Ron+15] O. Ronneberger, P. Fischer, and T. Brox. U-net: convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015 (cited on pp. 32, 33, 46, 60).

[Ros58] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958 (cited on p. 20).

[Ruc+16] N. Ruchaud and J.-L. Dugelay. Automatic face anonymization in visual data: are we really well protected? In *Electronic Imaging*, 2016 (cited on pp. 39–42).

[Sea+79] J. A. Sears and C. J. Grieve. The crookes tube. *School Science and Mathematics*, 79(6):493–501, 1979 (cited on p. 7).

[Sha+17] S. Sharma, S. Sharma, and A. Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017 (cited on p. 23).

[She+10] J. Shepherd, B. Fan, Y. Lu, L. Marquez, K. Salama, J. Hwang, and E. Fung. Dual-energy x-ray absorptiometry with serum ferritin predicts liver iron concentration and changes in concentration better than ferritin alone. *Journal of clinical densitometry : the official journal of the International Society for Clinical Densitometry*, 13:399–406, October 2010. DOI: 10.1016/j.jocd.2010.05.003 (cited on p. 13).

[Shi+18] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *Simulation and Synthesis in Medical Imaging: Third International Workshop, SASHIMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3*, pages 1–11. Springer, 2018 (cited on p. 46).

[Shr+06] C. P. Shrivastava, S. Devgarha, and V. Ahlawat. Mediastinal tumors: a clinicopathological analysis. *Asian Cardiovascular and Thoracic Annals*, 14(2):102–104, 2006 (cited on pp. 14, 15).

[Shu23] D. Shulman. Optimization methods in deep learning: a comprehensive overview. *arXiv preprint arXiv:2302.09566*, 2023 (cited on p. 26).

[Sim+14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014 (cited on pp. 5, 28).

[Sue17] P. Suetens. *Fundamentals of medical imaging*. Cambridge university press, 2017 (cited on pp. 8, 9, 14).

[Suz17] K. Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017 (cited on pp. 17, 18).

[Sze+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015 (cited on p. 28).

[Tas+22] S. Taslimi, S. Taslimi, N. Fathi, M. Salehi, and M. H. Rohban. Swinchex: multi-label classification on chest x-ray images with transformers. *arXiv preprint arXiv:2206.04246*, 2022 (cited on pp. 36, 61, 69, 73).

[The13] The University of Texas System. Health insurance portability and accountability act. 2013. URL: https://utsystem.edu/sites/default/files/documents/publication/2013/400-health-insurance-portability-and-accountability-act-hipaa/hipaa400.pdf. accessed: 01.05.2024 (cited on p. 2).

[Tho+18] R. E. Thomas, S. K. Banu, and B. Tripathy. Image anonymization using clustering with pixelization. *Int. J. Eng. Technol*, 7:990–993, 2018 (cited on p. 42).

[Vas+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017 (cited on p. 30).

[Wan+12] S. Wang and R. M. Summers. Machine learning and radiology. *Medical image analysis*, 16(5):933–951, 2012 (cited on p. 17).

[Wan+17] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017 (cited on pp. 5, 13, 15, 36, 53, 61, 74, 79, 90).

[Wan+20] H. Wang, Q. Zhao, Q. Wu, S. Chopra, A. Khaitan, and H. Wang. Global and local differential privacy for collaborative bandits. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 150–159, 2020 (cited on p. 43).

[Weg+22] B. Weggenmann, V. Rublack, M. Andrejczuk, J. Mattern, and F. Kerschbaum. Dp-vae: human-readable text anonymization for online reviews with differentially private variational autoencoders. In *Proceedings of the ACM Web Conference 2022*, pages 721–731, 2022 (cited on p. 44).

[Wil+20] M. J. Willemink, W. A. Koszek, C. Hardell, J. Wu, D. Fleischmann, H. Harvey, L. R. Folio, R. M. Summers, D. L. Rubin, and M. P. Lungren. Preparing medical imaging data for machine learning. *Radiology*, 295(1):4–15, 2020 (cited on p. 2).

[Wol23] A. B. Wollek. *Deep Learning for Clinical Decision Support Systems in Chest Radiography*. PhD thesis, Technische Universität München, 2023 (cited on p. 39).

[Won+17] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE transactions on visualization and computer graphics*, 24(1):1–12, 2017 (cited on p. 65).

[Wor16] World Health Organization. To x-ray or not to x-ray? 2016. URL: https://www.who.int/news-room/feature-stories/detail/to-x-ray-or-not-to-x-ray-. accessed: 01.05.2024 (cited on p. 1).

[Wor23] World Health Organization. Chronic respiratory diseases. 2023. URL: https://www.who.int/health-topics/chronic-respiratory-diseases. accessed: 01.05.2024 (cited on p. 14).

[Xia+19] X. Xiao, L. Wang, K. Ding, S. Xiang, and C. Pan. Deep hierarchical encoder–decoder network for image captioning. *IEEE Transactions on Multimedia*, 21(11):2942–2956, 2019 (cited on p. 30).

[Yoo+19] J. Yoon, D. Jarrett, and M. Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019 (cited on p. 46).

[Yoo+20] J. Yoon, L. N. Drumright, and M. Van Der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24(8):2378–2388, 2020 (cited on pp. 46, 47).

[Zei+14] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014 (cited on p. 28).

[Zha+17a] L. Zhang, L. Lu, I. Nogues, R. M. Summers, S. Liu, and J. Yao. Deeppap: deep convolutional networks for cervical cell classification. *IEEE journal of biomedical and health informatics*, 21(6):1633–1643, 2017 (cited on pp. 28, 29).

[Zha+17b] L. Zhang, L. Lu, I. Nogues, R. M. Summers, S. Liu, and J. Yao. Deeppap: deep convolutional networks for cervical cell classification. *IEEE Journal of Biomedical and Health Informatics*, 21(6):1633–1643, 2017. DOI: 10.1109/JBHI.2017.2705583 (cited on p. 28).

[Zho+22] H. Zhong and K. Bu. Privacy-utility trade-off. *arXiv preprint arXiv:2204.12057*, 2022 (cited on p. 4).

[Zhu+17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017 (cited on p. 46).