,,

Universiteit Leiden
The Netherlands

Opleiding Informatica

Generating Synthetic Liver Tumor CT scans for Medical AI Applications

Sana Asghari (s3677117)

Supervisors:
Dr. D.M. Pelt & André Mesquita Fery Antunes

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

015/07/2025

Abstract

This thesis investigates the use of Generative Adversarial Networks (GANs) to synthesize liver CT images for medical imaging research. A custom preprocessing pipeline was developed using the LiTS dataset to extract and label axial slices containing liver tumors. A GAN model, referred to as TumorGAN, was trained to generate synthetic liver images that mimic the visual characteristics of tumor-containing CT scans. The generated images were evaluated both visually and through quantitative metrics. Results showed that, although the synthetic slices bore a superficial resemblance to real CT images, they were heavily affected by noise, exhibited poor structural coherence, and failed to capture clinically meaningful tumor features.

Contents

1 Introduction

1.1 Background and Motivation

Medical imaging is essential in diagnosing and monitoring disease. Computed tomography (CT) scanning is frequently used to detect liver tumors[1], which are often difficult to identify in early stages. These scans produce detailed cross-sectional views of internal anatomy, allowing clinicians to visually examine the liver and surrounding regions. Such clarity is critical when evaluating abnormal tissue. However, training machine learning models for tasks like tumor detection remains challenging, largely due to the limited availability of annotated medical data[2]. Collecting and labeling CT scans requires domain expertise, strict privacy measures, and considerable time, resulting in datasets that are often small and heterogeneous, restricting model development and performance.

To address this problem, researchers have explored synthetic image generation through machine learning. Among these, Generative Adversarial Networks (GANs) have emerged as a commonly used approach[3]. These models are trained to generate artificial images that resemble real examples from a given dataset. In practice, one model (the generator) creates new images, while another (the discriminator) attempts to distinguish between real and generated samples. Through this competitive training process, the quality of the generated images gradually improves[4]. Although GANs have shown promising results in domains such as natural image synthesis and art generation, applying them to medical imaging introduces specific complications. Because medical images need fine structural details, accurate contrast, and precise anatomy, generating them is much harder than creating ordinary visual content.

Despite these challenges, synthetic CT scans hold potential for expanding training datasets and reducing issues such as class imbalance in tasks like tumor segmentation or classification[5]. By supplementing real data with synthetic samples, researchers can increase the number of examples available to machine learning models without the need for further clinical data collection. This thesis investigates whether a basic GAN, trained on a filtered subset of 2D CT slices that show liver tumors, is capable of producing synthetic images that exhibit both visual and structural similarity to real examples. The focus lies in evaluating the realism and consistency of the generated images, as well as identifying the limitations of using this generative method in a medical imaging context.

1.2 Research Problem

This thesis explores the potential of GANs to generate medically plausible synthetic CT images of the liver with visible tumors. The central research question is whether a vanilla GAN, trained on a filtered set of tumor-containing CT slices, can produce synthetic images that reflect the structural and visual characteristics of real tumorous liver scans. More broadly, the work examines whether such synthetic images hold any potential for supporting medical research or augmenting diagnostic training datasets.

In support of this main objective, a key sub-question of *How can we effectively use existing datasets to create a dataset of healthy and tumor-containing 2D liver slices that can be used to train generative models?* is also addressed. This question guides the pre-processing and data labeling efforts required to prepare meaningful inputs for GAN training.

1.3 Objectives

The primary objective is to evaluate the capability of a basic GAN model in generating 2D axial CT slices that resemble real images of the liver affected by tumors. To achieve this, the study involves constructing a carefully preprocessed dataset from volumetric CT scans, training the GAN under constrained computing conditions, and systematically assessing the quality of the generated outputs.

1.4 Contributions

This thesis contributes a custom data pre-processing pipeline that converts 3D CT volumes from the LiTS dataset[6] into 2D slices, separated into healthy and tumor-containing categories. A vanilla GAN architecture is implemented and trained on the tumor subset to assess its ability to learn and reproduce the visual features of liver tumors. Finally, the quality of the generated images is evaluated using the Fréchet Inception Distance (FID), a widely used metric for assessing the similarity between real and generated image distributions, providing a baseline for comparison in future work[7].

1.5 Thesis Overview

The structure of this thesis is as follows. The next chapter provides medical context regarding liver anatomy, tumor characteristics, and the importance of CT imaging in diagnosis. This is followed by a review of related work on GANs in medical applications, with a particular focus on synthetic image generation. The methodology chapter outlines the dataset preparation, model architecture, and training strategy used in this project. The results chapter presents both qualitative and quantitative evaluations of the generated images, followed by a discussion of the model's limitations and possible directions for improvement. The thesis concludes by reflecting on the findings and proposing future extensions that could enhance the realism and use of synthetic medical images.

2 Medical Background

2.1 Anatomy of the Liver

The liver receives blood from the hepatic artery, which supplies oxygen-rich blood, and the portal vein, which carries nutrient-rich blood from the gut [8]. Blood flows through small vessels called sinusoids, which have thin, porous walls that allow close contact between blood and liver cells. Most of the liver is made up of hepatocytes, responsible for metabolism and detoxification, while other cells such as Kupffer cells and sinusoidal endothelial cells support immune functions.
This structure allows the liver to filter substances efficiently and monitor what enters the body, making it a key organ for both metabolism and immune defense[9].

2.2 Nature of Liver Tumors

Liver tumors are classified based on their site of origin. When a tumor begins in the liver, it is referred to as a *primary cancer*. Tumors that originate elsewhere and spread to the liver are known

as *secondary cancers* or *metastases*.

2.2.1 Primary Liver Tumors

Primary liver tumors develop from different cell types in the liver, such as hepatocytes, bile duct cells, or blood vessel lining cells. Hepatocellular carcinoma (HCC) is the most common, accounting for about 75–85% of cases, and is often associated with chronic conditions such as hepatitis B or C, alcohol-related liver disease, or non-alcoholic fatty liver disease [10].
HCC is highly diverse in its histology and clinical behavior. The 5th edition of the World Health Organization (WHO) Classification of Digestive System Tumors recognizes several subtypes, including steatohepatitic, clear cell, macrotrabecular-massive, and fibrolamellar [11]. These variants differ in morphology, molecular mutations, and prognosis, making subtype identification important for diagnosis and treatment [10].

2.2.2 Secondary Liver Tumors

Secondary liver tumors originate from cancers such as bowel, breast, or lung cancer and retain the characteristics of their primary tissue [12]. Cancer cells spread to the liver mainly via the bloodstream or lymphatic system. Because the liver filters blood from much of the body, it is a common site for metastases. Treatment is based on the primary cancer type rather than the liver location [12].

3 CT Imaging and Its Importance in Medical Imaging

Computed Tomography (CT) is an imaging technique that produces cross-sectional images of an object or the human body using X-rays and computer-based reconstruction. Unlike conventional X-ray imaging, which results in overlapping shadow projections, CT uses multiple X-ray projections from different angles around the body. In the resulting CT images, air appears dark due to low X-ray absorption, soft tissues are displayed in varying shades of gray, and dense structures such as bone or metal appear bright because of their high absorption [13].
Although CT data are initially two-dimensional (2D), these slices can be digitally stacked and processed to form three-dimensional (3D) reconstructions. This capability enables researchers and clinicians to view internal structures from different angles and in multiple planes. CT is widely used in medicine for diagnosing tumors, strokes, fractures, and cardiovascular or pulmonary diseases, as well as for planning surgery or radiation therapy [13].
CT imaging plays a central role in tumor detection due to its ability to show differences in tissue density [14]. This makes it particularly effective in identifying tumors that have clear structural or density differences compared to surrounding tissues. CT scans are also crucial in studying liver tumors because the liver has a dense and complex network of blood vessels. Correctly identifying tumors is challenging since physicians must distinguish tumors from blood vessels and healthy liver tissue. This differentiation is particularly important for surgical planning, as the tumor's location relative to the hepatic vessels determines whether it can be safely resected [15].

3.1 Dataset Overview

The dataset used in this study is a preprocessed version of the Liver Tumor Segmentation (LiTS) dataset, originally developed for the LiTS17 Challenge held in conjunction with ISBI 2017 and MICCAI 2017 [6]. It is widely recognized within the medical imaging community as a benchmark resource for the development and evaluation of liver and tumor segmentation algorithms in abdominal CT scans. Although the dataset contains real CT scans featuring liver tumors, it does not distinguish between different tumor subtypes. All lesions, whether primary or secondary, are grouped under a single segmentation label. As a result, models trained on this dataset are limited to detecting and segmenting tumors but cannot differentiate between specific pathological types.
The dataset is composed of CT volumes and their corresponding segmentation masks, both stored in the NIfTI file format with the `.nii` extension. The NIfTI format is a standard used in medical imaging for storing volumetric data, allowing for efficient access and manipulation of 3D arrays[16]. In this dataset, each NIfTI file encodes either a grayscale CT volume or a voxel-level segmentation map.
Each patient case is represented by a pair of files:

- `volume-XXX.nii` : The contrast-enhanced CT scan for a given patient,
- `segmentation-XXX.nii` : The manually annotated label map corresponding to the same patient.

The segmentation masks classify each voxel into one of three categories being background (label 0), liver tissue (label 1), and tumor lesion (label 2). In total, the dataset comprises CT scans from 131 patients. These were released in two separate parts("LiTS Dataset Part 1" and "LiTS Dataset Part 2")[6].
The dataset provides substantial clinical variability in terms of tumor size, shape, location, and density. This heterogeneity, along with the high-quality manual annotations and diverse institutional origins of the scans, makes the LiTS dataset a reliable and realistic benchmark for liver and tumor segmentation research.

4 Generative Adversarial Networks (GANs)

4.1 Basic GAN Architecture

GANs are a type of deep learning model used to generate new data that looks similar to a given dataset. They have become especially popular for tasks like creating realistic images, generating audio, and even producing videos. The idea behind GANs is to train two neural networks at the same time being a **generator** and a **discriminator**. These two networks are designed to compete against each other in a learning process.[4].
This interaction can be understood as a two-player **minimax** game, where the generator tries to minimize the ability of the discriminator to correctly classify its outputs, and the discriminator tries to maximize its classification accuracy. The objective function of this game is given by:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right]$$

In this expression, $p_{data}(x)$ represents the real data distribution, $p_z(z)$ represents a prior distribution from which random noise vectors z are sampled, $D(x)$ is the probability assigned by the Discriminator that x is real, and $G(z)$ is the data generated by the Generator from the noise z. The Generator seeks to minimize this value by producing realistic data, while the Discriminator seeks to maximize it by correctly distinguishing real from fake data. Over time, if training is successful, the generator produces outputs that are so realistic that the discriminator cannot tell them apart from the real ones.

4.1.1 The Generator

The Generator G is responsible for producing fake samples that resemble the real data. Its input is a random noise vector z, sampled from a simple distribution such as a Gaussian or uniform distribution, denoted as $p_z(z)$. This noise contains no meaningful information about the real data, it merely acts as a starting point or a seed for the generation process. The output of the Generator, $G(z)$, is a synthetic sample intended to mimic a real data point drawn from $p_{data}(x)$. For instance, if the goal is to generate images, $G(z)$ would output an image with the same dimensions and characteristics as real images in the dataset. The Generator's objective is to fool the Discriminator, making $D(G(z))$ as close to 1 as possible, meaning the Discriminator believes the generated sample is real[17].

4.1.2 The Discriminator

The Discriminator D acts as a binary classifier that distinguishes between real and fake data. Its input is either a real data sample x from the true data distribution $p_{data}(x)$ or a fake sample $G(z)$ generated by the Generator. The output $D(x)$ is a single scalar between 0 and 1, representing the probability that the input is real. If $D(x)$ is close to 1, the Discriminator believes the sample is real, if it is close to 0, it believes the sample is fake. In the minimax game, the Discriminator aims to maximize $V(D, G)$ by assigning high probabilities to real samples ($\log D(x)$ term) and low probabilities to fake samples ($\log(1 - D(G(z)))$ term)[17].

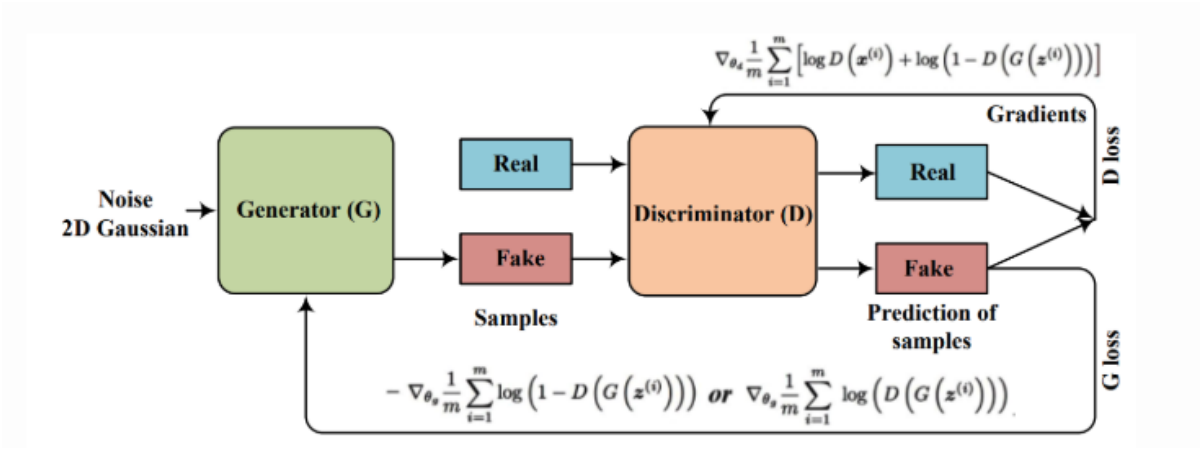


Figure 1: Schematic of a Generative Adversarial Network [17]. The generator G takes in a random noise vector z and outputs a fake sample $G(z)$. Both $G(z)$ and real data X are given to the discriminator D, which predicts whether each input is real or fake. The system trains both networks in a loop to improve performance over time.

Figure 1 illustrates the basic architecture of a GAN. On the left side, a noise vector z is fed into the generator G, which tries to produce data that resembles the training data. The output $G(z)$, along with real data X, is passed to the discriminator D. The discriminator outputs a prediction indicating whether each sample is real or fake. This feedback is used to update both networks. The discriminator is trained to improve its classification accuracy, while the generator is trained to produce more convincing fake data that can fool the discriminator.

4.1.3 Training Procedure

The training process of GANs is what makes them unique and powerful. Unlike traditional neural networks, GANs involve two models learning simultaneously through a competitive setup.
Training is alternated between the two networks, meaning they take turns updating their parameters. At the beginning of training, the generator's outputs are mostly random and not convincing, so the discriminator can easily detect that they are fake. However, the generator gradually improves by receiving feedback from the discriminator. When the generator is trained, it uses the discriminator's feedback to adjust itself and make its fake outputs more realistic. This causes the discriminator to be less confident in its predictions. As a result, the discriminator also has to improve. When the discriminator is trained, it learns to better identify fake observations, which in turn pushes the generator to come up with new and better fakes[4].
This back-and-forth training dynamic continues over many iterations. As the generator gets better at fooling the discriminator, and the discriminator becomes more skilled at spotting fakes, both networks evolve and improve. Ideally, this process continues until the generator creates samples that are so realistic that the discriminator cannot distinguish them from real data.

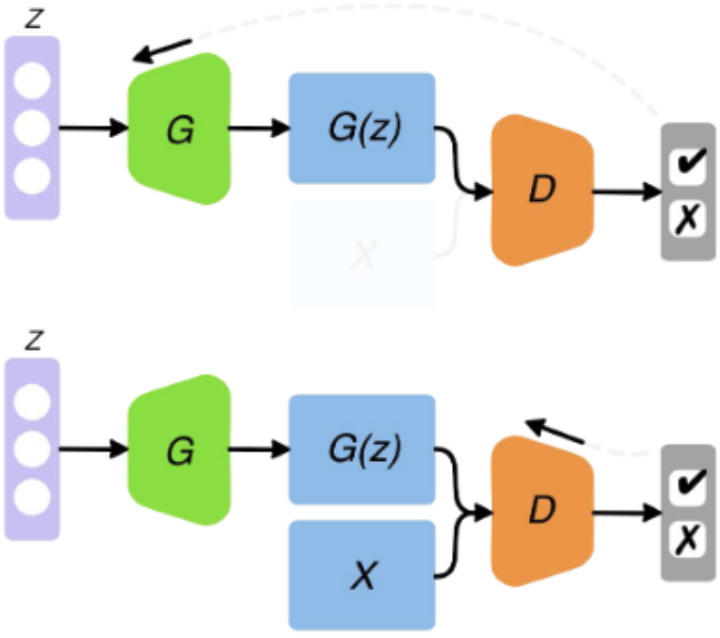


Figure 2: Schematic of the GAN training process[18]. In each training step, the generator G produces fake data from random noise z, and the discriminator D attempts to distinguish this from real data X. Feedback from the discriminator is used to improve the generator. The networks are trained in alternation to push each other to improve.

Figure 2 [18]shows how training is alternated between the generator and discriminator. In the top part of the diagram, the generator creates fake data $G(z)$ from random input z, which is judged by the discriminator D. In the bottom part, both the real data X and the generator's output $G(z)$ are evaluated by the discriminator. Based on the discriminator's performance, each network updates its parameters. This loop continues until the generator produces highly realistic outputs that are difficult to classify as fake.

4.1.4 Advantages and Challenges

GANs come with a mix of strengths that set them apart from older generative models. Unlike traditional generative models, GANs do not rely on Markov chains. A Markov chain is a stochastic process that generates samples by iteratively transitioning between states, where each state depends only on the previous one, often used in models like Boltzmann machines to approximate data distributions. However, Markov chains can be computationally expensive and slow to converge, requiring many iterations to produce high-quality samples and introducing complexity in tuning transition parameters. By avoiding Markov chains, GANs simplify the training process, as they rely entirely on backpropagation through the generator and discriminator networks, which is more straightforward and computationally efficient [4].
However, GANs also present some significant challenges. Training can be unstable due to the adversarial nature of the process, where the generator and discriminator must be carefully balanced. If this balance is not maintained, the generator may suffer from mode collapse, a situation in which it produces a limited variety of outputs regardless of input. Additionally, GANs do not offer an explicit likelihood function, which, in statistical modeling, is a mapping from parameter values to the probability of observing the data given those parameters. The likelihood quantifies how well a model with specific parameters explains the observed data and is central to maximum likelihood estimation methods[19]. The lack of such a function makes it difficult to evaluate the performance of the model quantitatively[4].
Despite these challenges, GANs form the basis for a wide range of successful applications in computer vision and beyond.

4.2 Other Variants of GANs

Beyond the standard GAN framework, several specialized variants have been developed to suit different generative tasks. **Conditional GANs** (cGANs) incorporate additional input such as class labels, images, or text to control the output, making them suitable for tasks where guidance or specificity is needed [20]. For fine-grained control and high-quality image synthesis, **StyleGAN** introduces a mapping network and style-based modulation, enabling precise manipulation of image attributes and highly realistic outputs [21]. On the other hand, if the goal is to translate between image domains without paired data such as converting paintings to photos or healthy to diseased organs, **CycleGAN** offers an effective solution using cycle consistency loss to preserve content structure [22].

5 Related Work

5.1 GANs in Medical Imaging

Medical image synthesis is a vital research area in biomedical engineering, addressing the challenge of limited annotated datasets for deep learning applications like medical image analysis [23]. This thesis uses GANs to generate synthetic healthy and tumor-containing liver CT scans, enabling comparative evaluation of their quality.

Recent studies have applied GANs to medical imaging, particularly for CT scan synthesis. A patch-based GAN was used for brain CT-to-MRI translation, demonstrating high-fidelity cross-modal synthesis but not focusing on direct CT generation [24]. Similarly, Costa employed GANs for retinal image synthesis, enhancing segmentation tasks, which highlights GANs' versatility but differs from liver-specific applications [25]. These works provide a foundation for synthetic image generation, though they address different modalities or tasks compared to our focus on liver CT scans.

A highly relevant study by Frid used DCGANs to generate synthetic liver lesion CT images (cysts, metastases, hemangiomas) from a dataset of 182 scans [26]. They trained separate DCGANs per lesion class, augmenting a CNN classifier to improve sensitivity from 78.6% to 85.7% and specificity from 88.4% to 92.4%[26]. This approach aligns with the use of separate GANs for healthy and tumor-containing liver datasets in this thesis. However, their focus on classification contrasts with this papers goal of generating and comparing synthetic images for quality assessment.

Current research often prioritizes classification or cross-modal tasks, with limited emphasis on comparing synthetic healthy and tumor-containing images using quantitative metrics like PSNR or SSIM. Additionally, most studies use 2D data, overlooking 3D CT context, and face challenges in capturing subtle tumor features. This thesis addresses these gaps by training basic GANs on separate healthy and tumor-containing liver CT datasets to generate 2D slices, comparing their realism against a baseline of real scans using visual and quantitative evaluations, potentially informing future 3D extensions.

5.2 Evaluation Metrics for Synthetic Images

Evaluating the quality of GAN-generated medical images is not straightforward because standard quantitative measures do not always match how humans judge realism. Higaki et al. [27] explored this issue in myocardial perfusion imaging (MPI) by using both objective metrics and a Visual Turing Test (VTT).

For the objective evaluation, they reported a Fréchet Inception Distance (FID) of 100.6 between real and generated images, showing that the two distributions were still quite different. Interestingly, the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) score was lower (better) for generated images (32.3 ± 7.5) than for real ones (49.9 ± 8.0), suggesting that synthetic images sometimes looked smoother and visually cleaner.

The VTT was used to check how realistic the images looked to experts. Nine cardiologists were asked to classify real and fake MPI images. Their first Correct Answer Rate (CAR) was only 61.1%, close to random guessing. After being told some clues about typical GAN artifacts, the CAR improved to 80.0%.

This work highlights that quantitative scores alone can be misleading in medical image synthesis. Human evaluation, especially by trained clinicians, is important when judging whether GAN-

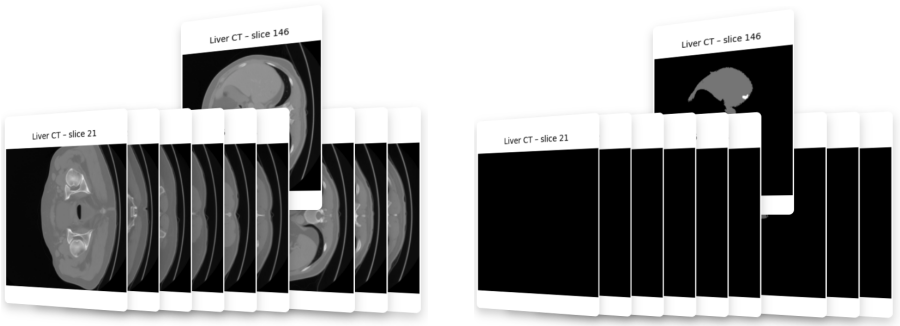
generated images are realistic enough for medical use.

6 Methodology

6.1 Data Pre-processing

6.1.1 Volumetric Slice Extraction and Initial Labeling

The LiTS dataset contains 3D CT scans of the abdomen along with segmentation masks that label different parts of the body. These masks help identify which slices show the liver and which ones contain tumors, making it possible to find the exact locations of tumors within the scan.



(a) CT scan slices **(b)** Corresponding segmentation masks

Figure 3: Decomposition of a 3D abdominal CT scan from Patient 61 into 2D axial slices with the corresponding segmentation mask for each slice.

To enable the use of 2D GANs, each 3D CT scan was decomposed into a stack of 2D axial slices, as illustrated in Figure 3. For each axial CT slice, the corresponding segmentation slice was extracted, forming image-mask pairs. This was done using a custom Python pre-processing pipeline that iterated through each patient volume slice by slice.
Since most GAN architectures are designed for 2D inputs, converting 3D volumes into 2D slices allows the model to focus on spatial features within each plane while significantly reducing computational load. It also increases the number of training samples, which is beneficial for model learning.
Each slice pair was temporarily cached and analyzed to assess tumor presence before being assigned to the appropriate training category.

6.1.2 Binary Tumor Classification and Dataset Separation

Each axial slice was then classified into one of two categories based on its segmentation mask:

- **Healthy Slice:** If the segmentation mask associated with a slice contained **no pixels labeled as** 2, the slice was considered free of visible tumor and classified as *healthy*.
- **Tumor Slice:** If the segmentation mask included **any non-zero count of voxels labeled as** 2, the corresponding slice was labeled as *tumor-containing*.

This binary classification scheme was applied across the full dataset, effectively splitting it into two clearly delineated subsets: `healthy` and `tumored`. Each subset was saved to a separate directory

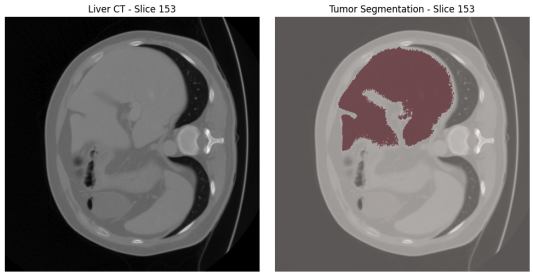


Figure 4: Slice 153 from Patient 61 showing a liver tumor. The left image is the original CT scan, and the right image shows the corresponding segmentation mask highlighting the tumor region in red.

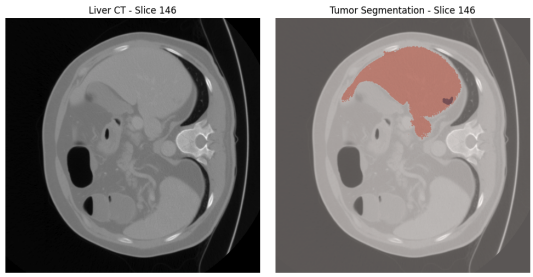


Figure 5: Slice 146 from Patient 61 showing both liver and tumor regions. The left image is the original CT scan, and the right image displays the segmentation mask, with the liver and tumor regions separately annotated.

structure, further organized into training and validation splits using a consistent directory format required by PyTorch's `ImageFolder` class.

6.1.3 Preprocessing Pipeline

All extracted slices were preprocessed using a consistent transformation pipeline to prepare them for training the GAN models. Since CT scans are originally grayscale, each image was explicitly converted to a single grayscale channel to avoid introducing any unintended color information from RGB representations. Afterward, the images were resized to a standardized resolution of 64×64 pixels using bilinear interpolation. This resolution was selected to strike a balance between retaining enough anatomical detail and ensuring efficient model training, as higher resolutions can be computationally expensive. Following resizing, the pixel intensity values were normalized to fall within the range $[-1, 1]$. This normalization step is important because the GAN's generator network uses a `tanh` activation function at its final layer, which naturally outputs values in this same range. Matching the input data to this output range helps the model train more effectively by ensuring the real and generated images are on the same scale. These transformations were implemented using the `torchvision.transforms` library in Python and were applied dynamically during training using a PyTorch `DataLoader`. This approach allowed for scalable and consistent preprocessing across both the healthy and tumor datasets.

6.1.4 Resulting Dataset Summary

The original dataset that got organized into two distinct subsets for further GAN training data was seprated into two distinct subsets being the **HealthyGAN dataset**, which consisted of CT slices containing liver tissue without any tumor presence, and the **TumorGAN dataset**, which included

slices with visible tumor regions alongside liver tissue. The HealthyGAN dataset was used to train a model capable of generating anatomically plausible liver images free of pathological features, while the TumorGAN dataset was used to train a model focused on synthesizing realistic tumor-bearing liver CT slices.
The complete dataset comprised 19,156 axial slices derived from 131 patient scans. A detailed breakdown of the dataset composition is provided in Table 1.

Table 1: Dataset composition

Split	**Healthy**	**Tumored**	**Total**
Train	10,716	6,410	17,126
Val	1,257	773	2,030
Total	11,973	7,183	19,156

6.1.5 Resolution Tradeoffs and Downscaling Rationale

As part of the preprocessing , all CT images were downscaled from their original resolution of 512×512 pixels to 64×64. This reduction was motivated by the need to train the model efficiently within the constraints of limited computational resources, including restricted GPU access and modest local hardware. Lower-resolution images reduce memory consumption and training time, enabling the entire pipeline to run on accessible hardware while supporting iterative development and experimentation.
While this approach significantly lowered the computational burden, it came at the cost of visual fidelity. Medical CT scans rely on fine anatomical details, such as soft-tissue gradients, tumor boundaries, and localized contrast which are lost at lower resolutions. At 64×64, small tumors or subtle lesions may become indistinct or entirely imperceptible, limiting the model's ability to learn clinically meaningful patterns. This tradeoff likely contributed to the observed limitations in image realism and evaluation metrics, such as SSIM and PSNR, discussed in later chapters.
Despite these drawbacks, downscaling was a necessary compromise that allowed to focus on the broader feasibility of GAN-based medical image generation.

6.2 Model Architecture

The GAN architecture developed in this study consists of two fully connected neural networks: a Generator (G) and a Discriminator (D). Both networks are designed to operate on grayscale axial CT slices of size 64×64, consistent with the image format described in the preprocessing pipeline. The Generator transforms a 100-dimensional latent vector $z \sim \mathcal{N}(0, I)$ into a synthetic image. This process is implemented using a sequence of four linear layers with progressively increasing dimensionality, followed by non-linear activations. ReLU activations are applied after each layer except the final one, which uses a `tanh` activation to align the output range with that of the normalized input data. The final vector is reshaped into a single-channel image of resolution 64×64. Conversely, the Discriminator receives an image input and determines whether it is real or generated. The image is first flattened into a one-dimensional vector and then passed through a series of linear layers with decreasing size. LeakyReLU activations are used between layers to preserve gradient

flow. A final sigmoid activation produces a scalar output representing the probability that the input image is real.
An overview of the architecture for both networks is provided in Table 2.

Table 2: GAN model architecture

Component	Layer	Output Shape	Activation
Generator	Linear	256	ReLU
	Linear	512	ReLU
	Linear	1024	ReLU
	Linear	4096	Tanh
	Reshape	$1 \times 64 \times 64$	–
Discriminator	Flatten	4096	–
	Linear	512	LeakyReLU (0.2)
	Linear	256	LeakyReLU (0.2)
	Linear	1	Sigmoid

6.3 Design Choices

The architecture was designed with simplicity and efficiency in mind, making it suitable for grayscale CT images. Fully connected layers were used instead of convolutional layers to keep the model lightweight and easy to train. Since the images are small, this design choice allowed for quick experimentation without heavy computational cost, while still capturing important spatial features in medical images.
The activation functions were chosen to support stable training. ReLU was used in the Generator to promote efficient learning and avoid vanishing gradients. In the Discriminator, LeakyReLU was applied to help maintain gradient flow, especially during early stages of training when the generator may produce poor-quality images. The final layer of the Generator uses a `tanh` activation to match the normalized pixel range of the input data.
A 100-dimensional latent vector was used as input to the Generator. This size is a common choice in GANs, providing enough variation for the model to generate diverse outputs, without making training more complex than necessary[28].
Adam was selected as the optimizer for both networks because of its adaptive learning rate and momentum features, which help stabilize GAN training[29]. A learning rate of 2×10^{-4} was used, following common practice in similar image generation tasks.
Overall, this minimal design offers a good balance between simplicity and performance. While more advanced architectures like convolutional or residual networks could be explored in the future, the current setup provides a strong baseline for generating medical images in a controlled and interpretable way.

6.4 Training Setup

6.4.1 Training Parameters

The GAN model was trained for a total of 1000 epochs. During this training process, the model learned to generate synthetic liver tumor CT images based on adversarial feedback. At regular

intervals, generated image samples and model weights were saved to monitor progress and preserve outputs.
The key hyperparameters and configuration details used during training are shown below:

Parameter	**Value**
Total Epochs	1000
Image Size	64×64 pixels (grayscale)
Batch Size	64
Latent Vector Dimension	100
Learning Rate	0.0002
Noise Distribution	Standard Normal $\mathcal{N}(0, 1)$
Checkpoint Frequency	Every 20 epochs
Image Sample Frequency	Every 30 epochs

Table 3: GAN training parameters

6.4.2 Loss Functions and Optimization

The generator maps random noise vectors to synthetic CT images, while the discriminator learns to differentiate between real and the fake generated images.
The training objective uses binary cross-entropy (BCE) loss:

$$\mathcal{L}_{BCE}(x, y) = -\left[y \log(x) + (1 - y) \log(1 - x)\right]$$

During each iteration:

- The discriminator is trained using BCE loss on both real images (label 1) and fake images (label 0). The total loss is the sum of both.
- The generator is trained to maximize the discriminator's classification error, by producing images that are labeled as real (label 1).

6.4.3 Training Progress and Convergence

Throughout training, the generator and discriminator losses were recorded for each epoch. Initially, the discriminator had a much lower loss, indicating it could easily distinguish real from fake images. However, as training progressed, the generator improved and the gap between the generator and discriminator losses gradually narrowed.
This convergence suggests that both networks were learning effectively. By the end of training, the losses had approached each other, indicating a balanced adversarial process.
This balance is desirable in GAN training, as it typically corresponds to the generator producing images that are difficult for the discriminator to distinguish from real samples.

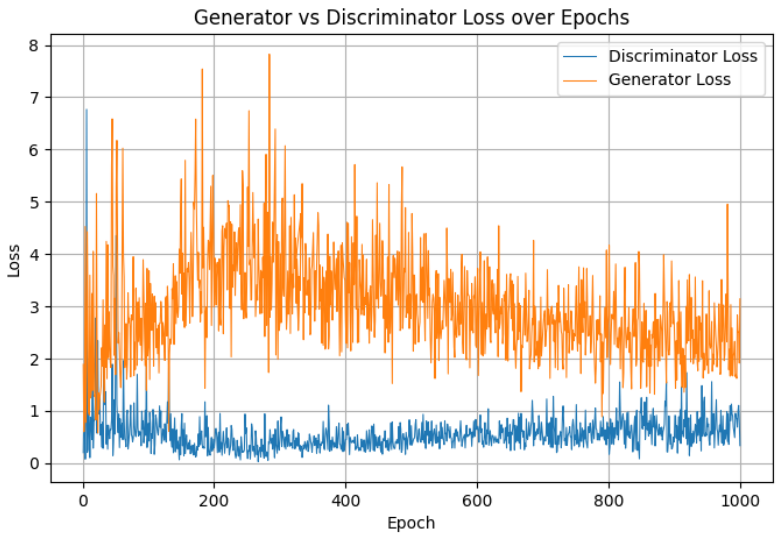


Figure 6: Generator and discriminator losses over training epochs

6.5 Evaluation Using Fréchet Inception Distance (FID)

The Fréchet Inception Distance (FID),is a metrics to quantitatively evaluate the quality of images generated by GANs[7]. It compares real and generated images by checking how similar their important features such as edges, textures, and patterns are. These features are extracted using a pre-trained Inception V3 model, which is a deep learning model originally trained on a huge image dataset (ImageNet). Because it has already learned to recognize thousands of objects, it can pick out meaningful and high-level details from images, making it very effective for analyzing and comparing them.[30].
During the 1000 epochs of GAN training, a total of 29 tumor-containing synthetic CT images were saved as sample checkpoints. To properly evaluate the GAN, it was necessary to compare these generated images with the same number of real CT images. Therefore, 29 random real CT images were selected from the tumor-containing CT dataset to create a separate evaluation dataset.

The *pytorch_fid* function is the core method used to compare these two sets of images. This function operates on the statistical representations of the two datasets rather than directly comparing pixel values. Moreover, in the process of computing the FID, all images in both datasets were internally rescaled to 299x299 [31] and they got converted to 3-channel RGB since the Inception-v3 network expects 3-channel input. For each dataset, the Inception-v3 network processes the images and produces 2048-dimensional feature vectors from one of its final pooling layers. The empirical mean and covariance matrix of these feature vectors are then computed for both the real and generated images, and these statistical quantities are used to calculate the FID.
Mathematically, the FID is defined as[31]:

$$FID = \|\mu_1 - \mu_2\|^2 + Tr\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1\Sigma_2\right)^{1/2}\right)$$

where μ_1 denotes the mean feature vector of the real images, and μ_2 denotes the mean feature vector of the generated (synthetic) images. The terms Σ_1 and Σ_2 represent the covariance matrices

of the feature vectors of the real and generated images, respectively. The term $\|\mu_1 - \mu_2\|^2$ is the squared Euclidean distance between the two mean vectors, indicating how far apart the average features of the two datasets are. The Tr refers to the trace of a matrix, which is the sum of its diagonal elements and corresponds to the total variance in this context. The expression $(\Sigma_1\Sigma_2)^{1/2}$ represents the matrix square root of the product of the two covariance matrices and measures the similarity of the feature distributions of the two datasets[32].

6.5.1 Interpretation of FID Values

The general interpretation of the FID value is summarized in table 4 [31]:

Table 4: FID value Ranges

FID Score	Interpretation
<10	Excellent, almost indistinguishable from real images
10 – 30	Good, realistic images with minor artifacts
30 – 50	Moderate, visible differences compared to real images
> 50	Poor, generated images are far from real distribution

The FID score calculated for the generated tumor containing CT images was found to be **239.55**. This value is much higher than the commonly accepted threshold of 50 for reasonable image quality, whoch shows that GAN has done a poor job in generating realistic images.

7 Results and Discussion

7.1 Results of TumorGAN

The TumorGAN model, based on a vanilla GAN architecture, was trained for 1000 epochs on the tumor-containing subset of the LiTS dataset. Despite this extended training duration, the resulting synthetic images were far from satisfactory in terms of medical realism and diagnostic quality. Visual inspection of the generated samples revealed that most outputs were extremely noisy, lacked anatomical coherence, and failed to capture the complex structural and textural features of real tumorous liver CT slices.

This outcome reflects a broader challenge inherent in applying vanilla GANs to high-stakes domains such as medical imaging. Vanilla GANs rely solely on fully connected layers and basic loss functions like binary cross-entropy. While they are theoretically capable of modeling complex data distributions, in practice they often suffer from unstable training, mode collapse, and poor convergence, especially when the data distribution is high-dimensional and subtle, as is the case in liver tumor CT scans.

Tumorous liver slices contain intricate anatomical details, low contrast boundaries, and complex noise patterns typical of CT imaging. A vanilla GAN, which lacks architectural innovations such as convolutional layers, attention mechanisms, or progressive growing, is inherently limited in its ability to model such fine-grained visual structure. As a result, the generated outputs often resemble coarse, grainy blobs rather than coherent anatomical patterns. Even at epoch 1000, the samples retained significant noise artifacts and failed to convincingly replicate the appearance of actual tumor tissue or liver parenchyma.

Another critical factor limiting the model's performance was the **training time**. Vanilla GANs are notoriously data-hungry [33] and sensitive to hyperparameters, and in many documented cases, successful convergence requires tens of thousands of epochs, especially on complex datasets. The 1000-epoch training, although computationally intensive, was insufficient to fully learn the highly nuanced distribution of real tumorous CT slices. Additionally, the lack of architectural specialization like convolutional feature extraction, meant that the model had no spatial inductive bias, making the learning process even slower and less stable.
It is also important to highlight that vanilla GANs provide no mechanism for controlling the characteristics of the tumors being generated such as their size, shape, anatomical location, or contrast levels. This lack of conditional control significantly limits their applicability in medical image synthesis, where precise and clinically meaningful variation is often required. Additionally, the discriminator network, being relatively shallow and composed solely of fully connected layers, was insufficiently expressive to enforce high-resolution realism. As a result, the feedback it provided to the generator was weak and imprecise, further contributing to the noisy and anatomically implausible outputs observed during training.

7.1.1 Visual Evolution of Generated CT Slices

To better understand the model's learning progression, 10 samples are selected from different epochs (from epoch 100 to 1000) and are visualized below. As shown in Figure 7, the early outputs (epochs 100 to 400) are dominated by high-frequency noise, blurred shapes, and random textures. Over time, particularly by epochs 500 and 800, the structure becomes slightly more defined. However, even at epoch 800, the tumor regions are not clearly distinguishable, and the anatomical consistency remains weak.

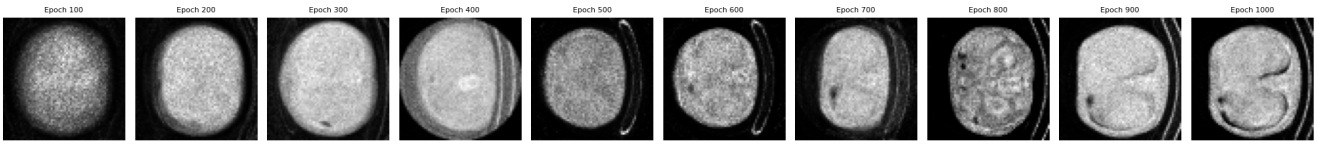


Figure 7: Progression of TumorGAN generated CT slices across training epochs. Samples are shown from 5 different epochs including the first synthetic image and last one. While image structure gradually improves, even later-stage outputs remain noisy and lack sufficient clinical realism.

The noticeable change in the shape and structural composition of the images at epochs 900 and 1000 may indicate a shift in the generator's strategy to minimize adversarial loss. In earlier epochs, the generator predominantly produced smooth, averaged shapes, which served as a stable and relatively effective method to reduce discriminator rejection, as such generic forms are statistically "safe" approximations of the training distribution. However, in the later stages of training, the generator appears to have changed its approach by introducing new and more varied structural patterns, which, while visually distinct, do not necessarily correspond to realistic anatomical features. This behavior is consistent with GAN dynamics, where the generator can exploit weaknesses in the discriminator by focusing on producing outputs that are harder to classify as fake, even if they deviate from the true data distribution. Such a shift may indicate partial mode collapse or a

biased exploration of specific regions in the latent space, resulting in images with different but not necessarily better structural representations.

7.2 Comparison and Analysis

Figure 8 shows a side-by-side comparison between synthetic CT image generated by TumorGAN at epoch 800 and a real tumor-containing CT slice which initially had a scale of 512x512 but is been down scaled to 64x64 to match the size of the generated CT images . Although the overall structure appears similar, with a central liver region and a darker tumor-like mass, the generated image is noticeably noisier. Fine details, such as tumor edges and soft tissue gradients, are blurred or lost entirely.
This visual similarity in layout does not translate into diagnostic realism. The noise artifacts and lack of sharp anatomical boundaries significantly reduce the clinical value of the generated image.

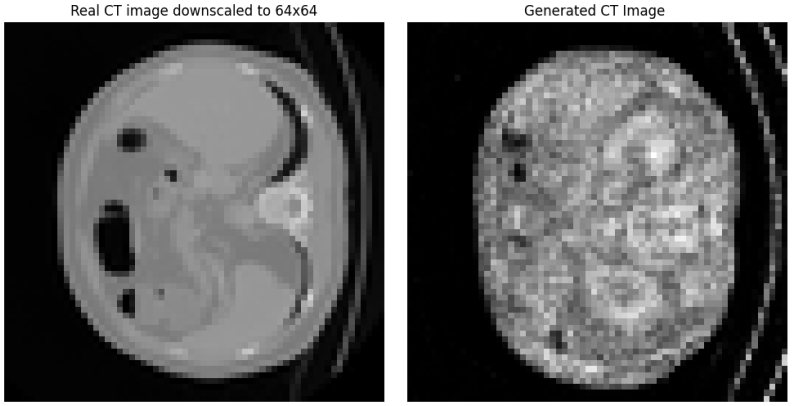


Figure 8: Real liver CT image containing tumor Versus Fake CT image of liver containing tumor generated by Vanilla GAN

7.3 Discussion of Limitations

Despite the potential of generative models in medical image synthesis, this project encountered several practical limitations that constrained both model choice and experimental depth. The most critical constraint was the combination of limited computational resources and dataset-specific requirements, which ultimately led to the use of a vanilla GAN architecture.

Given the resource constraints, specifically the use of Google Colab and a mid-range personal laptop with limited GPU availability, it was not feasible to implement and train more sophisticated GAN variants such as DCGAN, StyleGAN, or GANs with convolutional layers and attention mechanisms. These architectures often require significantly more memory, prolonged training time, and optimized infrastructure, which were not accessible within the scope of this bachelor's project.
Additionally, the dataset posed unique challenges. While the LiTS dataset is clinically rich, its volumetric 3D nature necessitated extensive preprocessing, including slice extraction, labeling, and normalization. This added overhead made the pipeline more sensitive to training runtime and storage availability, especially when dealing with high-resolution inputs or multiple architectures. Moreover, even within the relatively lightweight vanilla GAN framework, training for 1000 epochs

already demanded several hours per run, making iterative tuning and experimentation difficult. Attempts to increase image resolution or add complexity to the model resulted in crashes, slowdowns, or failed checkpoints due to memory exhaustion.

A significant limitation in this study was the use of 64×64 resolution images, which restricted the model's ability to capture fine-grained anatomical details essential for medical imaging. One potential direction for future work is to generate high-resolution outputs using more advanced generative models. Deep learning architectures such as Progressive Growing GANs (ProGAN) or StyleGAN[34] have been developed specifically to address this challenge, enabling the synthesis of detailed, high-resolution images in stages or by separating style and structure. Another option is to train a separate super-resolution neural network, such as ESRGAN (Enhanced Super-Resolution GAN) [35], which can upsample low-resolution outputs from the generator into higher-resolution versions while preserving structural details. These techniques could be integrated into the current pipeline either by replacing the vanilla GAN architecture entirely or by appending an upsampling network to the generator. Exploring these methods may help overcome the resolution bottleneck while keeping training tractable on limited hardware.

The combined factors of dataset dimensionality, hardware limits, and long training durations, prevents a full investigation of more advanced generative techniques or architectural improvements. While vanilla GANs provided a baseline for exploring synthetic tumor generation, their inability to capture fine-grained anatomical detail further exposed the limitations imposed by the constrained environment.
In summary, the exclusive use of a simple GAN model was not a design preference but a necessity imposed by limited hardware, time, and data handling capabilities. Future work with better access to compute power and memory could explore more expressive GAN variants that are better suited for the complexity of medical image synthesis.

7.4 Ethical and Clinical Considerations

The generation of synthetic medical images, particularly those representing pathological conditions like liver tumors, brings forward significant ethical and clinical concerns. While the use of generative models such as GANs presents exciting possibilities for data augmentation, model pretraining, and research acceleration, their application must be approached with caution, especially when the generated outputs may influence real-world clinical workflows or decisions.
As demonstrated in the results, the GAN model struggled to produce anatomically accurate or diagnostically useful outputs even after 1000 epochs of training. The generated images were consistently noisy and lacked the detailed structure required for any form of clinical application. Given this, the synthetic outputs are not only unreliable but could potentially mislead or misinform if introduced into sensitive medical contexts without appropriate safeguards.
Moreover, the use of simplistic architectures like vanilla GANs for medical imaging raises questions about the ethical responsibility of researchers. Unlike applications in art or entertainment, mistakes in healthcare data modeling can have direct consequences on human health. Generating images that mimic disease without a high standard of quality, validation, and oversight could contribute to bias, misdiagnosis, or flawed decision-support systems if used improperly.
In conclusion, while synthetic data generation holds potential for supporting medical research,

deploying models like vanilla GANs without rigorous validation and clinical oversight is both ethically irresponsible and clinically unsafe. These tools should never be used in real-world diagnosis or treatment settings without comprehensive testing, expert evaluation, and adherence to regulatory standards.

8 Conclusions and Future Work

8.1 Answering the Research Questions

Can GANs generate realistic and medically accurate liver CT images?
Based on the experiments conducted in this thesis, the answer is partially no, at least not with a basic GAN architecture. While the synthetic CT images generated by TumorGAN showed rough structural resemblance to real images such as similar grayscale patterns and general liver positioning, they lacked anatomical clarity, exhibited high noise, and failed to reproduce the subtle contrast and texture seen in true medical CT scans.

How can we effectively use existing datasets to create a dataset of healthy and tumor-containing 2D liver slices that can be used to train generative models?
existing datasets such as the LiTS dataset can be effectively used to create a specialized dataset of healthy and tumor-containing 2D liver slices suitable for training generative models. By using the provided segmentation masks, it is possible to accurately separate slices into healthy and tumor-containing categories, as demonstrated in this study. The pre-processing pipeline which involved volumetric slice extraction, binary tumor classification based on segmentation labels, and normalization to a consistent resolution, proved to be a reliable method for producing clean, labeled 2D datasets.
This approach is particularly effective in scenarios where no task-specific dataset is available, as it allows researchers to repurpose general clinical imaging datasets for generative modeling tasks. Moreover, because the LiTS dataset contains high-quality annotations and diverse tumor presentations, the resulting 2D dataset works perfectly for training baseline generative models, such as GANs, by providing sufficient structural variability and clear labeling.

Are tumors well-generated?
No. The generated images were too noisy to clearly identify or locate tumor regions. The synthetic outputs lacked any distinct tumor boundaries or consistent structure.

Should GANs be considered reliable tools for medical research?
Not in their basic form. This thesis used a vanilla GAN with fully connected layers due to resource limitations. However, this does not discredit GANs as a concept. Advanced variants like DCGAN or StyleGAN with convolutional layers, attention mechanisms, and conditional training have shown more promising results in other medical imaging tasks.

In conclusion, while vanilla GANs are not reliable for generating diagnostically useful synthetic medical images, this study highlights their potential as a baseline method and underscores the promise of more advanced generative architectures.

8.2 Recommendations for Future Work

For future work, two directions can be pursued to improve the results and clinical relevance.
First, future studies should consider using more advanced GAN architectures such as StyleGAN or DCGAN, which are better equipped to capture detailed anatomical features. These models use convolutional layers and style-based modulation, which offers stronger performance in generating high-resolution and structurally coherent images.
Second, further experiments with vanilla GANs may still be useful, but they would require a much longer training schedule, potentially exceeding 5,000 epochs, along with access to more powerful hardware to handle the extended training time. With sufficient computational resources, prolonged training might allow the model to better capture the subtle grayscale textures and anatomical patterns present in liver CT data. Additionally, maintaining the original image resolution rather than down scaling would be essential for preserving fine structural details, such as tumor boundaries and soft-tissue gradients, which are critical for medical imaging applications.
Third, future work could explore diffusion models, which are a newer type of generative model that often produce more realistic and detailed images than GANs[36]. These models, such as Denoising Diffusion Probabilistic Models (DDPMs) [37], work by gradually removing noise from an image until a clear and realistic result appears. They are more stable to train than GANs and are less likely to generate repetitive or unrealistic images [36]. Because they can capture fine details, diffusion models could be very useful for medical imaging tasks, such as showing clear tumor boundaries or small tissue structures. However, they need more computing power and longer training times, but their success in other fields suggests they have strong potential for future medical research [38].
Pursuing either of these directions would help overcome the limitations in this thesis and bring synthetic medical imaging closer to practical use in research and model development.

References

[1] A. Krishan and D. Mittal, "Ensembled liver cancer detection and classification using ct images," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 235, pp. 232–244, Feb. 2021. Epub 2020 Nov 13.

[2] N. K. Dinsdale, E. Bluemke, V. Sundaresan, M. Jenkinson, S. Smith, and A. I. Namburete, "Challenges for machine learning in clinical translation of big data imaging studies," 2021.

[3] P. Paladugu, J. Ong, S. A. Kamran, E. Waisberg, N. Zaman, R. Kumar, R. D. Dias, A. G. Lee, and A. Tavakkoli, "Generative adversarial networks in medicine: Important considerations for this emerging innovation in artificial intelligence," 2023. Last revision on January 13, 2023.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[5] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 289–293, 2018.

[6] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C. W. Fu, X. Han, P. A. Heng, J. Hesser, *et al.*, "The liver tumor segmentation benchmark (lits)," *Medical Image Analysis*, vol. 84, 2023.

[7] Y. Yu, W. Zhang, and Y. Deng, "Frechet inception distance (fid) for evaluating gans," *China University of Mining Technology Beijing Graduate School*, vol. 3, no. 11, 2021.

[8] V. Racanelli and B. Rehermann, "The liver as an immunological organ," *Hepatology*, no. S1, 2006.

[9] Y. E. Parlar, S. N. Ayar, D. Cagdas, and Y. H. Balaban, "Liver immunity, autoimmunity, and inborn errors of immunity," *World Journal of Hepatology*, vol. 15, pp. 52–67, Jan. 2023.

[10] H. Kim, M. Jang, and Y. N. Park, "Histopathological variants of hepatocellular carcinomas: an update according to the 5th edition of the who classification of digestive system tumors," *Journal of Liver Cancer*, vol. 20, no. 1, 2020.

[11] L. M. Loy, H. M. Low, J. Y. Choi, H. Rhee, C. F. Wong, and C. H. Tan, "Variant hepatocellular carcinoma subtypes according to the 2019 who classification: An imaging-focused review," *AJR. American Journal of Roentgenology*, vol. 219, Aug. 2022. Epub 2022 Feb 16.

[12] Cancer Research UK, "How cancer can spread," 2023.

[13] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. IEEE Press, 1988. Reprinted as SIAM Classic in Applied Mathematics, 2001.

[14] E.-E. M. Azhari, M. M. M. Hatta, Z. Z. Htike, and S. L. Win, "Tumor detection in medical imaging: a survey," *International Journal of Advanced Information Technology*, vol. 4, no. 1, p. 21, 2014.

[15] Z. Zhang, S. Li, Z. Wang, and Y. Lu, "A novel and efficient tumor detection framework for pancreatic cancer via ct images," 2020.

[16] M. Moore, B. Patterson, S. Samuel, H. Sheridan, and C. Sorensen, *Neuroimaging DICOM and NIfTI Data Curation Primer*. Data Curation Network, 2020. Retrieved from the University Digital Conservancy, https://hdl.handle.net/11299/216582.

[17] D. Saxena and J. Cao, "Generative adversarial networks (gans): Challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–42, 2021.

[18] R. Saunders, "Gan." Unpublished lecture slides, 2025. Lecture slides, Generative AI course, Leiden University.

[19] H. Eghbal-zadeh and G. Widmer, "Likelihood estimation for generative adversarial networks," *CoRR*, vol. abs/1707.07530, 2017.

[20] T. DeVries, A. Romero, L. Pineda, G. W. Taylor, and M. Drozdzal, "On the evaluation of conditional gans," 2019.

[21] A. H. Bermano, R. Gal, Y. Alaluf, R. Mokady, Y. Nitzan, O. Tov, O. Patashnik, and D. Cohen-Or, "State-of-the-art in the architecture, methods and applications of stylegan," *Computer Graphics Forum*, vol. 41, no. 2, 2022.

[22] T. Wang and Y. Lin, "Cyclegan with better cycles," 2024.

[23] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *CoRR*, vol. abs/1702.05747, 2017.

[24] D. Nie, R. Trullo, C. Petitjean, S. Ruan, and D. Shen, "Medical image synthesis with context-aware generative adversarial networks," *CoRR*, vol. abs/1612.05362, 2016.

[25] P. Costa, A. Galdran, M. I. Meyer, M. D. Abràmoff, M. Niemeijer, A. M. Mendonça, and A. Campilho, "Towards adversarial retinal image synthesis," 2017.

[26] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[27] A. Higaki, Y. Kawada, G. Hiasa, T. Yamada, and H. Okayama, "Using a visual turing test to evaluate the realism of generative adversarial network (gan)-based synthesized myocardial perfusion images," *Cureus*, vol. 14, no. 10, p. e30646, 2022.

[28] I. Marin, S. Gotovac, M. Russo, and D. Božić-Štulić, "The effect of latent space dimension on the quality of synthesized human face images," *Journal of Communications Software and Systems*, vol. 17, no. 2, pp. 124–133, 2021.

[29] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pp. 1–2, Ieee, 2018.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, IEEE, 2016.

[31] ApX Machine Learning, "Interpreting fid scores." https://apxml.com/courses/generative-adversarial-networks-gans/chapter-5-evaluation-of-gans/interpreting-fid-scores.

[32] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017)*, pp. 6629–6640, 2017.

[33] J. Yoo, J. Park, A. Wang, D. Mohaisen, and J. Kim, "On the performance of generative adversarial network (gan) variants: a clinical data study," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 100–104, IEEE, 2020.

[34] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018.

[36] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[37] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[38] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *International Conference on Learning Representations (ICLR)*, 2021.