TUM Data Innovation Lab

Munich Data Science Institute

Technical University of Munich

&

TUM Chair for Application and Middleware Systems

Final report of project:

Generation of synthetic segmented medical images for tumor detection

Authors	Karl Richter, Niklas Hölterhoff, Sena Terzi, Nikola Selic
Mentor(s)	M.Sc. Rene Schwermer, M.Sc. Kathrin Khadra, M.Sc. Simona Santamaria
Project Lead	Dr. Ricardo Acevedo Cabra
Supervisor	Prof. Dr. Massimo Fornasier

Aug 2022

Abstract

Accounting for over 1.7 million deaths in 2018, lung cancer is one of the most common causes of cancer death worldwide. While tumors are hard to detect with the human eye, computer-aided diagnoses carries the potential of supporting doctors in the early discovery of nodules. Recent studies have shown that neural-network based object detection models can perform at a comparable error rate to doctors. While the training of these models requires massive amounts of training images, those need to contain not only the information whether an image contains a tumor (label), but also the precise location of the tumor (segmentation). Since medical images are highly sensitive and hospitals cannot share their data with other parties due to privacy constraints, training data for these models is scarce.

In this report, we first identify different methods to generating synthetic training images in the literature, then cluster them into three approaches, and lastly demonstrate the effectiveness of the individual approaches end-to-end. Our results indicate, that the inpainting of extracted tumors into healthy images can increase the performance of an object detection models significantly, yet, not achieve comparable results to a human radiologist.

Contents

1 Introduction

1.1 Motivation

Lung cancer is one of the most common cancer types worldwide. Coupled with the high mortality rate, the efficiency of lung cancer diagnosis and treatment is of high importance. In 2021, over 235,760 new cases and over 131,880 estimated deaths have been projected for the U.S. alone [32]. The chance of survival is significantly higher if lung cancer is diagnosed at an early stage [32]. In modern therapy, medical images play a crucial role in the diagnosis: they allow doctors to inspect a patient without any physical intervention. However, the reliable detection of tumors in medical images remains challenging - even for experienced doctors. A trained radiologists with several years of experience has been shown to detect lung tumors at an accuracy ranging from 0.54 to 0.87 [31]. This is where computer aided diagnosis (CAD) systems come in. Recent studies have shown that a neural network (NN) based object detection models can detect tumors in the chest area at comparable accuracy rates, providing the doctor with a second opinion [31]. However, the training of these models requires massive amounts of training images, that not only contain the information whether an image is tumorous or not (label), but also the precise location of the tumor (segmentation).

1.2 Problem

There are three main challenges that hinder the application of recent methods in the medical environment: First, medical images are highly sensitive and scans from multiple hospitals can usually not be combined at central location to train a joint model, due to data privacy constraints. Second, the generated images are very sensitive to the settings of the utilised scanner, its manufacturer as well as the physique of the patients in a certain region, which hinders the transfer and exchange of pre-trained models between hospitals. And lastly, the datasets of existing medical images are highly imbalanced and sparse. Fortunately, most people that undergo screening, end up not having a tumor on their chest. We will show in section three, that existing open datasets do not provide sufficient tumorous images to train an object detection model.

The project partner we collaborate with, Ryver.ai, provides hospitals with a service to generate synthetic medical images that can safely be shared without violating privacy constraints. The current objective is to generate synthetic images with a label whether the image contains a tumor or not. Together, we plan to identify innovative methods that allow to generate synthetic images that further contain a segmentation and allow to train an object detection model.

1.3 Project goals

The goal of this project is twofold: On the one hand, we aim to identify promising approaches towards generating synthetic segmented medical images from a theoretical perspective. On the other hand, we aim to implement and benchmark different annotation methods in an end-to-end pipeline, to provide the project partner with a recommendation on promising approaches to purse for an industrial implementation of the task.

1.4 Project plan

In the following, we will provide a short overview on the timeline and stages we defined for our project. In stage one, between the end of April and the end of May, we performed an extensive literature review to identify and categorize different approaches to generating segmented synthetic medical images from a theoretical perspective. Here it needs to be stated that the task we focused on in the project is not a dedicated field of research, thus we explored various directions to approach the task. We concluded the stage with a database of fundamental and recent literature that we identified as relevant for the task. In stage two, between the end of May and the end of June, we clustered the collected literature to derive three methodological approaches. For each approach, we defined distinct hypothesis that need to be validated in order to prove the effectiveness of the respective method. In stage three, between the end of June and the end of July, we trained models for the selected methods and implemented a testing framework to benchmark the different approaches.

1.5 Limitations

- We focused on identifying and clustering theoretical approaches to solving the data sparsity issue in literature and benchmarked promising implementations. However, we did not aim to propose a new neural network architecture or perform extensive tuning of hyperparameters of the selected methods.
- We only provide experimental implementations that allow for the comparison on a defined task, not the industrial application.
- As one team member dropped out after two thirds of the project, not all methods have been implemented and benchmarked to the state we desired.

2 Literature Review

In this section, we briefly introduce the various types of components utilised in this project. First, we introduce methods to generate synthetic images based on a set of original images. Second, we introduce methods to detect objects on images, allowing to segment tumors on medical images. Third, we introduce a method to convert non-tumorous images into tumorous ones and vice versa.

2.1 Synthetic image generation

Synthetic image generation describes the process of artificially creating images, that closely resemble the training data. Generative adversarial networks have become state-of-the-art and thus lay the foundation for synthetic image generation in our report.

2.1.1 Generative adversarial networks

Generative adversarial networks (GANs), first proposed by Goodfellow in 2014 [9], are a powerful class of neural networks for synthetic image generation: GAN based synthesis apporaches have achieved state-of-the-art performance in various image generation tasks, including super-resolution [22], image-to-image translation [42] or text-to-image synthesis [36]. First papers highlight the possible applications of GANs within the medical domain like cross-modality synthesis, segmentation, image reconstruction, classification and detection [38].

GANs consist of two networks that are trained concurrently, one network dedicated to image generation, called the generator G, the other one dedicated to discrimination, called discriminator D. The generator G takes pure random noise z as an input z, sampled from a prior distribution $p(z)$. For simplicity, $p(z)$ commonly uses a Gaussian or a uniform distribution for sampling. As a result of G, the output x_g should resemble visually the real sample x_r, drawn from the real data distribution $p_r(x)$. Hereby G learns a nonlinear mapping function, that is commonly parametrized by θ_g as $x_g = G(z; \theta_g)$. In contrast, D takes a real or generated sample as an input. A single value y_1 is returned by D, indicating the probability of the input being real or fake. D hereby learns a mapping parametrized by θ_d can be denoted as $y_1 = D(x; \theta_d)$. As a result of successful training, the generated samples form a distribution $p_g(x)$, which should approximate $p_r(x)$. It is the discriminator D's objective to differentiate these two groups of images, whereas the generator G is trained to confuse it as much as possible [38].

Over the past years, different types of GAN's have emerged: They are all based on the above mentioned principles, but perform slight adaptions to the general model. These variations can be grouped into three categories: First, variations in the objective of D, where different losses of D are proposed, in order to stabilize training and avoid mode collapse. These include f-GAN [26], ls-GAN [24] and WGAN [1]. Second, by passing additional information to G, variations in the objective of G can be achieved. These include the generation of images with desired properties. GANs using these properties at generation are commonly referred to as conditional GANs (cGAN). A popular example for a conditional GAN is the pix2pix image-to-image framework [12]. Lastly, GAN's

can vary by their underlying architecture: Different approaches have been proposed to improve training performance: Prominent approaches in this category include replacing the fully connected layers with fully convolutional downsampling / upsampling layers in DCGAN [28] or changing the architecture to generate high-resolution images in a progressive manner, like PCGAN [34] or StyleGAN [20]. In recent years, StyleGAN has become the go-to architecture for generation of high resolution images, performing especially well with faces, dogs and cars [3], but also showing first promising results in the medical domain [8].

During training, StyleGAN makes use of a progressive increase of the output image resolution. This allows StyleGAN models the successful synthesis of high-resolution images. Further iterations of StyleGAN make it possible to include style transfer and stochasticity in the process of generation [20]. The latest released version, StyleGAN 3, features an internal representation that allows to create generative models better suited for video and animation [21].

2.2 Image segmentation

Image segmentation describes the process of identifying objects or areas on images. In the following, we will introduce image segmentation methods relevant for our report.

2.2.1 U-Net

U-net is a neural network architecture that was developed primarily for the purpose of segmenting images. U-net has been widely adopted by the medical imaging community as the primary method for segmenting images. From CT scans to MRIs to X-rays and microscopy, U-net is used in nearly all major image modalities [30].

U-net's architecture is twofold: It consists of a contracting path as well as an expansive path. Hereby the contracting path follows a convolutional network, consisting of the repeated application of two 3x3 convolutions, also called unpadded convolutions, with each followed by a recitified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. The number of feature channels is then doubled at each downsampling step. Steps within the expansive path perform an upsampling of the feature map followed by a 2x2 convolution, effectively halving the number of feature channels, followed by a concatenation with the correspondingly cropped feature map that resulted from the contracting path, and also two 3x3 convolutions followed by a ReLU. The final layer uses a 1x1 convolution to map each 64-component feature vector to the desired number of classes. The original U-Net architecture therefore has 23 convolutional layers [30].

2.2.2 RetinaNet

RetinaNet, first introduced by Lin et al. in 2017, is a one-stage object detection model. To address class imbalance during training, it makes use a focal loss function. A focal loss adds a modulating term to the cross entropy loss, to enhance learning on hard negative examples. As part of RetinaNet, a backbone network and two task-specific subnetworks

form a unified network. Using an off-the-self convolutional network, the backbone computes a convolutional feature map over a whole input image. On the backbone's output, the first subnet performs convolutional object classification; the second subnet performs convolutional bounding box regression. In both subnetworks, the authors propose a simple design that is specifically designed for dense, one-stage detection.[23].

2.2.3 DatasetGAN

All current segmentation models are united in their need for large sets of segmented data that can be used for training. This poses a high cost factor for building new segmenation models, especially in the medical domain, where segmentation can only be performed by experts in their field. DatasetGAN, first introduced in 2021 by Zhang et al., takes an novel approach towards generating massive datasets of high-quality segmented images with minimal human effort: It relies on the power of state-of-the-art GANs like StyleGAN that generate realistic images. DatasetGAN hereby decodes the latent code to create a semantic segmentation of the image. The decoder only requires few labeled examples to be fully functional [41]. With only few labelled images necessary as input, it is a very cost-effective way of creating a segmentation model, especially in the medical domain.

2.3 Inpainting

Aside from the synthesis and segmentation methods, we also considered a third type of method as relevant for our project. Inpainting is the task of masking and reconstructing certain parts of an image. The methods has its origin in the post-processing of photos, to remove undesired objects. Thus, a majority of the research has been driven by companies such as Adobe or the Microsoft Media Research department. In the medical domain, inpainting methods have first been applied in 2018 to extract tumors and remove undesired objects (such as pacemakers) from images.

Inpainting methods build on the architecture of GANs, consisting of a generator and a discriminator. However, as a large portion of the ground truth image in the unmasked area is known, the loss function has to be re-formulated to optimise for the adjusted objective.

In 2018, Sogancioglu et al. compared the performance of back-then state-of-the-art inpainting methods to extract tumors from chest X-rays [33]. They considered context encoders developed by Pathak et al. in 2016 [27], a semantic inpainting method developed by Yeh et al. in 2017 [37] as well as a contextual attention model developed by Yu et al. in 2018 [39]. They showed that the contextual inpainting model performs best in reconstructing medical images. In 2021 Guendel et al. have shown the contextual attention models capabilities to extracting tumors and placing the extracted nodules in healthy images to generate synthetic training data [10].

While the images generated with previous inpainting methods have sometimes been distorted or blurry in the inpainted area [40], GAN architectures with a higher capacity have recently been applied for inpainting. They aim at inpainting images with more granular and fine-grained textures. Among these methods are DAM-GAN, developed by Cha et al.

in 2022 [4] as well as AOT-GAN, developed by Zeng et al. in 2021 [40]. The AOT-GAN proposed a new generator architecture that uses aggregated contextual transformation to learn repetitive patterns. Further, the discriminator is trained to explicitly detect blurry patterns to facilitate the synthesis of fine-grained textures.

3 Data

In this section, we provide an overview on the different types of medical images as well as the availability of training data in the respective format. Further, we decide on a dataset to use for this report and outline the steps we took for data exploration and pre-processing.

3.1 Medical image types and datasets

In general, three types of medical images are distinguished: X-ray, Computer Tomography (CT) and Magnetic Resonance Imaging (MRI). The type of scan that is performed, depends on the patients symptoms and the disease a doctor suspects.

3.1.1 X-ray

An X-ray is generated by sending external radiation through the body, that passes through soft tissues (such as the skin and muscles) and is reflected by dense parts (such as bones and tumors), allowing to generate a 2D image [35]. The chest X-ray (CXR) allows to inspect the condition of the heart, the airways, the lungs as well as the bones. CXR allows to diagnose several lung diseases, such as: effusion, infiltration, masses, nodules, pneumonia and edema. Since X-rays are in 2D, they are fast, cheap and less harmful for the patient than other medical image types. X-rays are preferred for emergency diagnosis and commonly conducted when a tumor is suspected.

ChestX-ray8. The NIH (National Institues of Health) ChestX-ray8 dataset contains 112,120 chest X-rays, of which 60,361 are labelled as healthy and the remainder contains one of 14 chest pathologies [34]. 2,705 of the images are labelled with "tumor", but only 79 images are provided with a bounding box, locating the tumor in the image.

CheXpert. The dataset released by the Stanford ML group contains 224,316 images, of which 6,856 contain a "lung lesion", which could be either a tumor or a mass in the lung [11]. Of the available images, none are provided with bounding boxes, indicating the lesion positions.

JSRT. The dataset released by the Japanese Society of Radiological Technology (JSRT) contains 257 images, of which 156 contain a tumor [14]. All of the tumorous images are provided with a bounding box, indicating the tumour location.

Dataset	# of total images	# of tumorous images	# of seg. tum. images
ChestX-ray8	112,120	2,705	79
CheXpert	224,316	6,856	0
JSRT	257	156	156

Table 1: Comparison of the testing framework results

Figure 1: Tumorous chest X-ray from the NIH ChestX-ray8 dataset

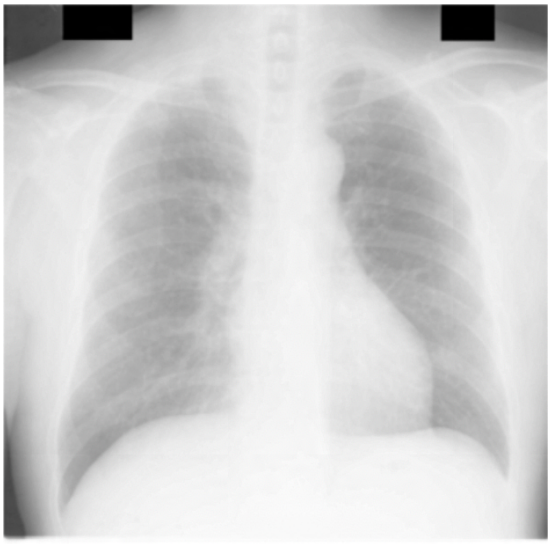

Figure 2: Tumorous chest X-ray from the JSRT dataset

3.1.2 CT

A CT image is constructed by combining multiple X-ray images from different angles to generate a 3D representation of bones, muscles, blood vessels and organs [5]. While the chest CT is more detailed, it is more expensive and takes longer to generate than the X-ray image. Thus, the CT scan is often only performed when there is a strong suspicion for malicious tissue or the X-ray scan indicated a further examination. For CT, we only found a single public dataset available that contains labelled tumorous images.

LIDC-IDRI. The LIDC-IDRI (The Lung Image Database Consortium) dataset [6], published by the National Cancer Institute (NCI) in 2011, contains 244,527 chest CT images, of which 1,018 contain a tumor.

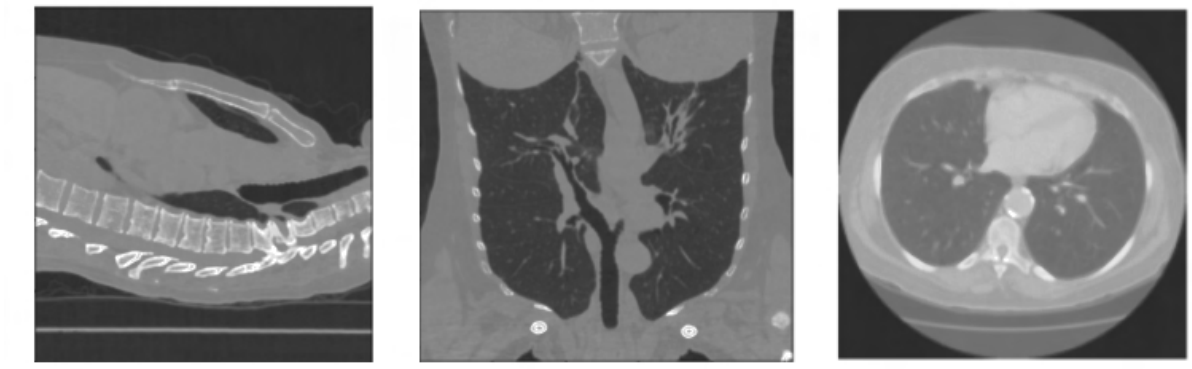

Figure 3: Chest CT image from the LIDC-IDRI dataset transformed into 2D

3.1.3 MRI

The MRI uses magnetic fields and computer generated radio waves to construct an image that allows to display not only hard masses, but also soft tissues (such as brain masses, blood vessels and organs) [25]. While the MRI poses the slowest and most detailed type of medical image, it is not suitable for the detection of hard objects, such as tumours in the chest area. Thus, there are no public datasets available containing chest MRI images with tumours and we excluded the type from our further research.

3.1.4 Conclusion

While MRI provides the most detailed medical images, X-ray and CT are more relevant for the early detection of tumors in practice. With the complexity of a third dimension in CT images, we agreed with the project partner to focus on X-ray images. Out of the three publicly available X-ray datasets, ChestX-ray8 seems the most relevant with regards to the total number of images and the number of segmented tumorous images.

3.2 Data exploration

The 112,120 images in the ChestX-ray8 dataset are provided by 30,805 patients. Thus, in some cases multiple images from different angles for the same patient are provided. Along with the images, the NIH published two CSV files containing metadata. The table `Data_Entry_2017_v2020.csv` contains metadata for each image, such as the finding of the image (eg. "no finding", "nodule" etc.), a patient id, the age of the patient, the gender as well as the image view position ("AP" for frontal view, "PA" for backside view). Further, it contains the original dimensions of the image, as all images have been resized to the same dimensions of 1024x1024 pixels. Further, the NIH provides a second file named `BBox_List_2017.csv`. The table contains the coordinates of the bounding boxes to indicate the location of the findings from the metadata table. As previously mentioned, only around 1% of the findings from the metadata table are provided with the coordinates of the bounding box.

3.3 Data pre-processing

As we will focus on healthy images as well as images with a segmented tumor in the following, we extracted these images from the 112,120 images. Further, we have created a table that contains the exact path to the respective files within our code repository as well as all the metadata outlined in the previous section. Lastly, we have removed few selected images that contained strong distortions or distracting elements, such as pacemakers or cables. Examplary images from the NIH dataset can be found in the appendix in section one.

4 Segmented synthetic image generation

In this section we outline three approaches to generating synthetic segmented images we identified based on the literature review in section two and recent publications. To the best of our knowledge, there is no existing overview on the different possibilities of combining methods to achieve this task.

4.1 Post-generation segmentation

The first approach treats the synthesis and segmentation of synthetic images as two separate tasks. In a first step, a model that generates both healthy and tumorous synthetic images based on original images is trained. In a second step, the synthetic images are segmented using a pre-trained segmentation model. The segmentation model is trained on segmented original images and transferred to synthetic images.

Figure 4: Post-generation segmentation

4.1.1 Technical approach

For the synthesis, any traditional conditional GAN model can be utilised. To perform the segmentation, traditional object-detection architectures such as U-Net and RetinaNet have been shown to perform well in the medical domain [31].

4.1.2 Challenges

While original medical images without a label (tumorous/non-tumorous) are sufficiently available to train a synthesis model, the availability of segmented original images to train the segmentation model poses a challenge. Segmentation models trained on insufficient training data are creating unreliable segmentations and make the synthetic segmented images not suitable as a training input for downstream machine learning models. This aligns with the original motivation for this project, which is around improving the performance of segmentation models for tumour detection by increasing the volume of data available for training.

4.2 Post-generation inpainting

The second approach again treats the synthesis and segmentation of images as separate tasks, however, with the difference that only healthy images are required for training the synthesis model. After generating a healthy synthetic image, a tumor is sampled into the healthy lung of an existing image and the respective segmentation mask is generated. To sample the tumor into the image, two approaches have been identified in the literature: Either the position within a given image is masked and inpainted with tumorous mass, what we denote as *tumorous inpainting* or, a tumorous region within a patient image is masked and inpainted with healthy mass, what we denote as *healthy inpainting*. The

latter approach allows to extract the shadows of tumors that can freely be sampled into arbitrary images. The approaches primarily differ by the type of images required for training: the tumorous inpainting approach requires tumorous images and the healthy inpainting approach requires sufficient healthy images.

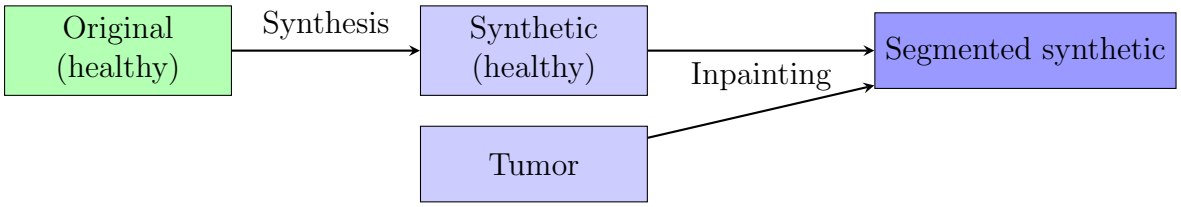

Figure 5: Post-generation inpainting

4.2.1 Technical approach

To perform the synthesis, any traditional unconditional GAN model can be utilised as only healthy synthetic images are supposed to be generated. To perform inpainting of tumors, recent GAN architectures have been proposed that evaluated the performance of inpainting on brain MRI images. For inpainting of healthy mass, any traditional inpainting method from the photo inpainting domain can be transferred to medical image inpainting, as shown by Sogancioglu et al. [33].

4.2.2 Challenges

As with all synthesis approaches, sufficient training data is required, thus for the inpainting approach using healthy images, enough data is available. When pursuing the tumorous inpainting approach, the availability of training data will likely pose a challenge, since tumorous pictures are not as widely available.

4.3 Segmented generation

The third approach treats the synthesis and segmentation of images as one interconnected task. For training, either already segmented original images, or the manual segmentation of synthetic images by a human within the training process are required.

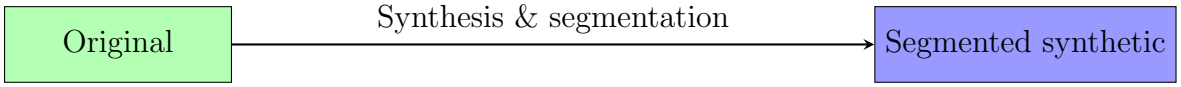

Figure 6: Segmented generation

4.3.1 Technical approach

The integrated synthesis and segmentation is a relatively complex and new approach, that has just recently sparked the interest of the research community. A promising approach, especially for the medical domain, are the combination of StyleGAN and DatasetGAN. As introduced in section two, StyleGAN is a synthesis model that learns a dense latent space during the training process. This latent space is then leveraged by DatasetGAN for

feature extraction. DatasetGAN then requires only few of the generated synthetic images to be segmented by a human annotator to train a segmentation model.

4.3.2 Challenges

By leveraging the few-shot concept and pre-extracted features, DatasetGAN seems promising for generating segmentations using only few segmented images as input. However, as StyleGAN and DatasetGAN are tightly interconnected but developed separately and at a high pace, it finding stable versions without conflicts remains a challenge. Further, the application of DatasetGAN has mainly been researched on medium to large sized objects in images. Whether DatasetGAN also performs sufficiently on small objects needs to be evaluated.

4.4 Conclusion

While the post-generation segmentation approach seems the most mature and researched, our project lacks the enough data to train an object detection model. While sufficient healthy training images are available, we will focus on the post-generation inpainting as well as the segmented generation approach for our further experiments. Here it needs to be stated that we were unable to test the segmented generation using the combination of StyleGAN and DatasetGAN during this project, thus cannot evaluate the feasability for medical images.

5 Experiments

We designed a set of experiments to compare the performance of the described heterogeneous approaches: This allows us to compare not only some approach-specific metrics, but their impact on the end-to-end performance of object detection models for tumours.

5.1 Experiment design

We designed an experiment that consists of two stages: In stage I, we generate segmented synthetic medical images containing tumors using three different approaches. With the image, a segmentation mask indicating the location of the tumor is provided for each image. In stage II, we train an object detection model on each of the sets of generated images. Further, we train the same model on the original images provided by the ChestX-ray8 dataset. This provides us with a baseline to compare the impact of the generated training images on the performance of the object detection model.

5.2 Stage I: Image generation

In this section, we briefly outline the implementation and training process for each of the three generational approaches selected in the previous section. Further, we provide the reader with a visual overview on the achieved results.

5.2.1 Contextual Attention Inpainting

Contextual attention (CA) models are the latest evolution of what can be described as traditional inpainting models. They have their origin in the inpainting on photos of objects and persons [39]. However, in 2018 Sogancioglu et al. compared the performance of various traditional inpainting methods on medical images and found the CA model to perform best [33]. Further, in 2020 Guendel et al. have shown the effectiveness of CA based models to perform training data augmentation on medical images [10].

Implementation. While Guendel et al. have shown the effectiveness of CA inpainting on tumorous medical images, there is no public implementation of their modified model available that would allow to reconstruct the results on a different dataset. Thus, we based our implementation on the original implementation by Yu et al. [13] and reconstructed the changes described by Guendel et al. This included some adaptations on the DataLoader to handle the black and white format that the ChestX-ray8 x-rays images are provided in. Further, we modified the inference function to handle batches of images instead of one image at the time. The adapted implementation used for this report can be found here [16].

Training. In contrast to the approach by Sogancioglu et al., which included the downsampling of the training images from 1024x1024 to 512x512 pixels to reduce training time, we trained the inpainting model on square 256x256 images with a masked area in the center of 128x128 pixels. The area around the mask is denoted as "context" for the model to base the inpainting on. Compared to the training data for inpainting on normal photos, the ratio of 2:1 among context and masked area is relatively small. However, as shown

by Soganciogl u et al., the context is sufficient for the less complex medical images. For each of the healthy 1024x1024 images in the ChestX-ray8 training dataset, we extracted 10 squares from random locations in the image with the dimension of 256x256. Thereby, we extracted 600.000 unique squares for training. We trained the contextual attention model on one NVIDIA Tesla V100-SXM2 GPU for 60.000 iterations. The training took around 24 hours.

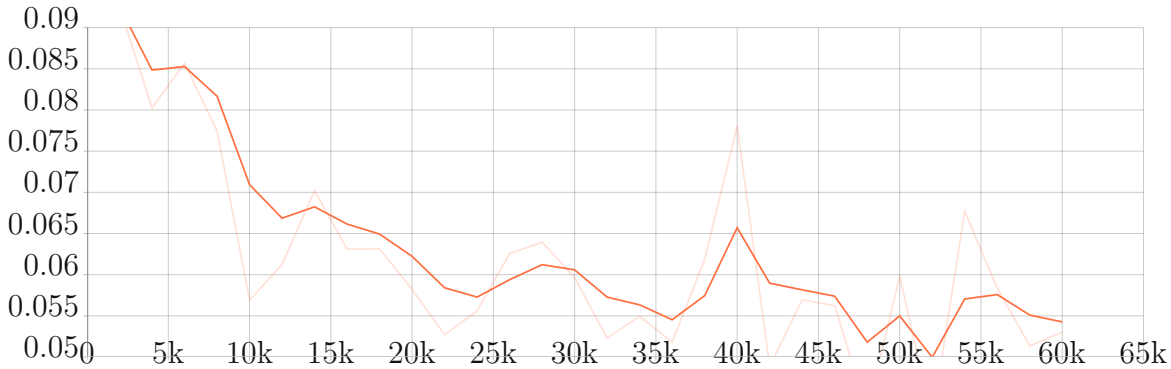

Figure 7: Contextual Attention: AE-Loss

Results. The performance of the contextual attention inpainting model can primarily be assessed by considering the AE-Loss metric. The figure 7 shows progress during training and the convergence around a value of 0.055. When comparing to the benchmarking of inpainting methods on medical images by Soganciogl u et al., this value does not reconstruct the performance they have been able to achieve, however, for the limited training time of our experiment the result can be considered sufficient. Aside from the evaluation of metrics, we inspected the masked and patched images. To visualise the differences between the original and the inpainted image, we subtracted the inpainted from the original image. In figure 8, the inpainting has been performed on tumorous images that have been inpainted with healthy mass. The subtraction of the inpainted (healthy) image from the original (tumorous) image results in an extracted tumour.

To convert healthy X-ray images into tumorous ones, we placed the extracted tumour at a semi-random location within the X-ray. As we aim to generate tumorous images containing lung cancer, the tumour location is restricted to be within two bounding boxes, indicating the position of the left and the right lung respectively. The bounding boxes for the lung are derived using visual inspection for a set of exemplary images. In figure 9, the original healthy image, the same image with the added tumour (red box) and location of the lung (yellow boxes) as well as the final tumorous image can be inspected. Further images can be found in the appendix in section two.

5.2.2 AOT-GAN Inpainting

The Aggregated Contextual-Transformation GAN (AOT-GAN) builds on the latest evolution of convolutional GAN models [40]. The model is specifically designed for the inpainting on high-resolution images. Similar to the CA inpainting approach, the AOT-GAN was mainly developed for inpainting on photos of objects and humans. To the best of our knowledge, recent inpainting methods have not been evaluated for the inpainting on

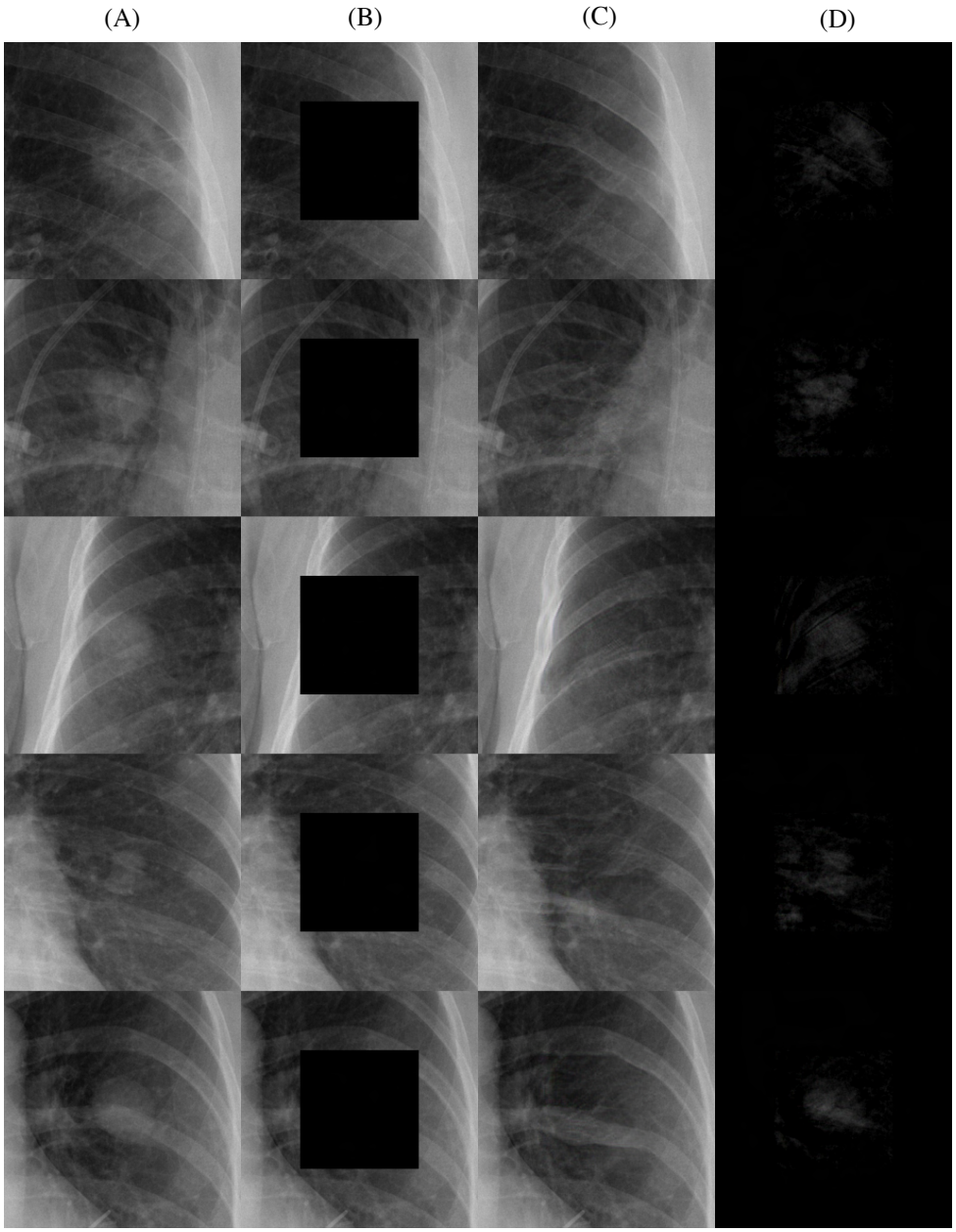

Figure 8: CA Inpainting: (A) Original, (B) Masked, (C) Inpainted, (D) Nodule

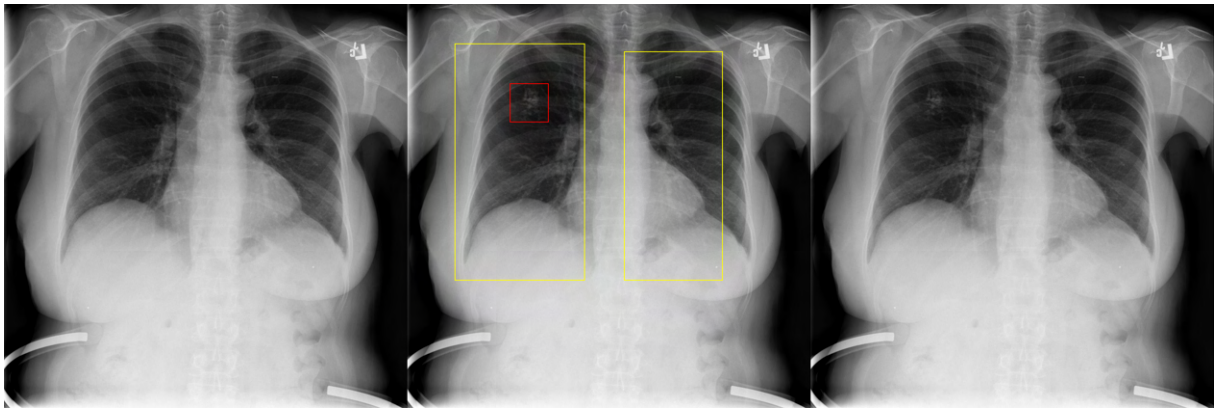

Figure 9: CA Inpainting: (A) Healthy, (B) Inpainted with bounding boxes, (C) Inpainted

medical images before.

Implementation. The publication by Zeng et al. provides an implementation of the model that is openly accessible [29]. While training the model on a small set of test images, we noticed several small and some major issues in the implementation. The issues have been around typos in the variable utilisation, a broken model saving function to store the training checkpoints as well as logical issues in the inference function. The latter caused the images for inpainting to be augmented twice, resulting in visually bad inpaintings while the training metrics seemed good. Therefore, we forked the original implementation and created a modified version that we used for this report. The code can be found here [15].

Training. We used the same 600.000 square 256x256 patches we extracted for the CA inpainting method to train the AOT-GAN to allow for comparability. We trained the AOT-GAN for a similar number of iterations (61.000) on an NVIDIA Tesla P100 GPU. The model training took approximately 22 hours. During the training process, the L1 loss as well as the perceptual loss converged to a value of 0.008 and 0.04 respectively.

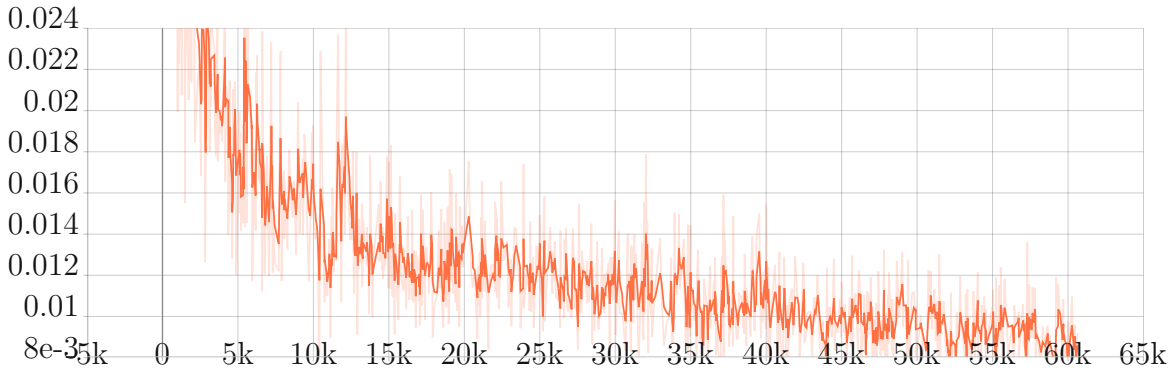

Figure 10: AOT-GAN: L1-Loss

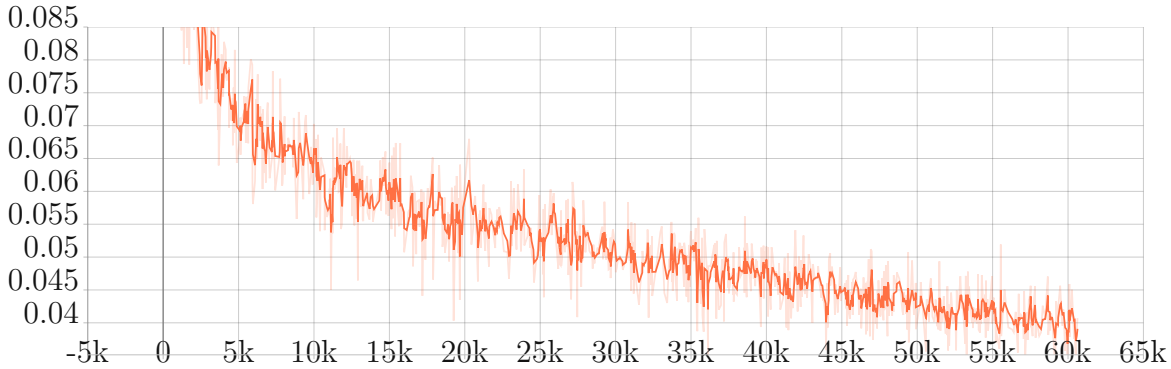

Figure 11: AOT-GAN: Perceptual Loss

Results We further inpainted the same masked images containing tumours as for the CA method. The patches and extracted tumors can be found in figure 12. When comparing the inpainted patches to the ones generated using the CA method, one can notice that the structures within the image are slightly sharper and eg. the rips look more similar to the original image than for the CA method. Further, the extracted tumours are more sharp, indicating a more precise inpainting.

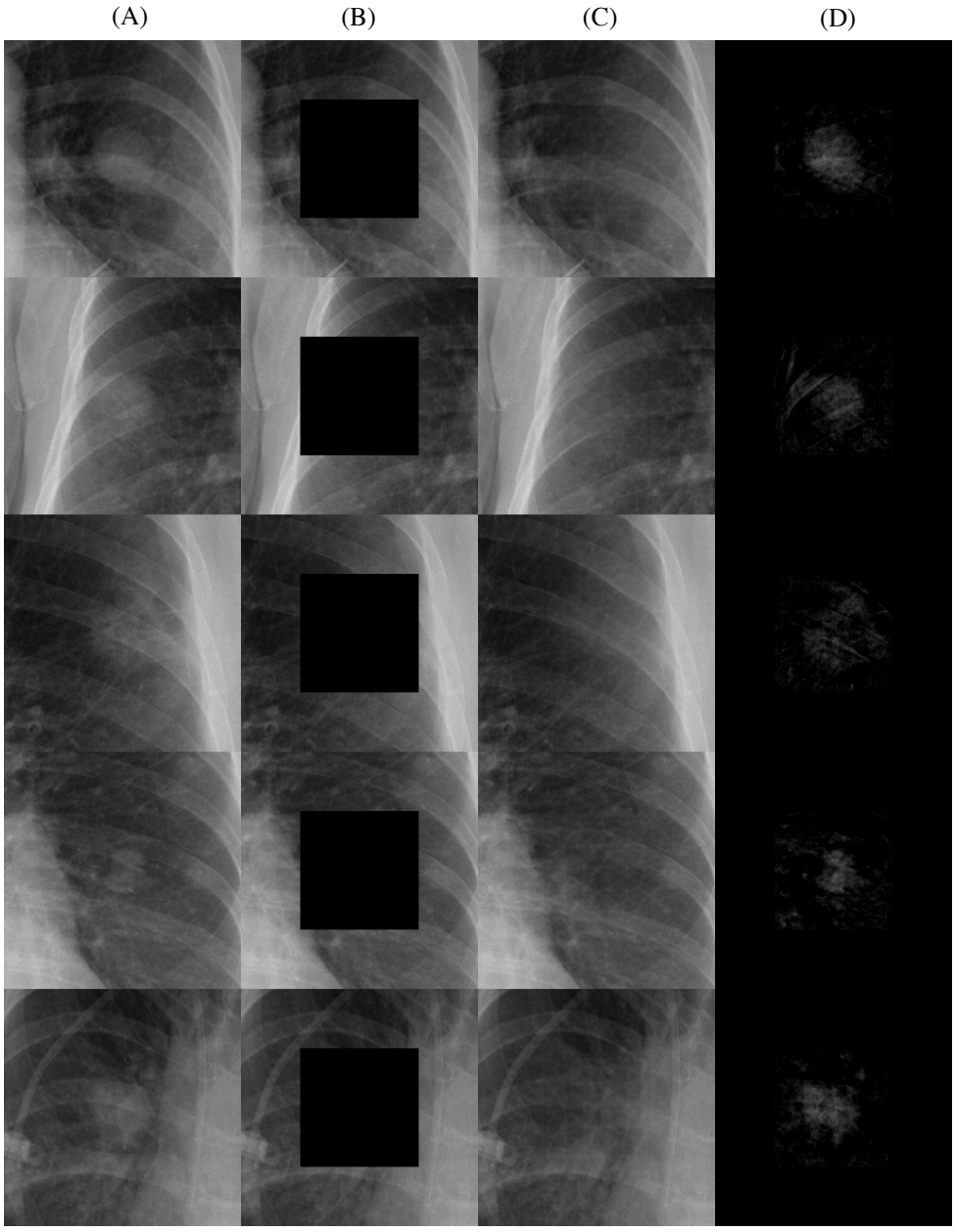

Figure 12: AOT-GAN: (A) Original, (B) Masked, (C) Inpainted, (D) Nodule

5.2.3 StyleGAN & DatasetGAN

StyleGAN is a conditional style-based GAN that allows to generate synthetic high-fidelity images. DatasetGAN is a few-shot model that builds on the trained StyleGAN layers that allows to segment the generated synthetic images.

Implementation. While StyleGAN has first been proposed in 2018 with an open-source implementation available, the architecture of the model as well as various further implementations have evolved rapidly. In parallel, DatasetGAN has been proposed, that builds on the internal architecture of StyleGAN. As a majority of the implementations seems mainly supplementary for the submitted papers, the code is to a large degree fragile and insufficiently documented. Despite experimenting with several combinations of StyleGAN and DatasetGAN implementations, we could not perform the end-to-end synthesis and segmentation as we planned for this approach. Thus, we will not further regard the segmented generation approach for this experiment and the testing framework. However, we will briefly summaries the implementations and experiments we conducted with both components individually for completeness and as a starting point for further research.

StyleGAN and DatasetGAN implementations	
Model	Issue
Original Tensorflow implementation	DatasetGAN needs a Pytorch Checkpoint
Official Pytorch implementation	CUDA Compatibility issues with newer GPUs
Official Pytorch StyleGAN3 implementation with StyleGAN2 flag	Working with adaptions on the original implementation to handle B/W images

Table 2: Comparison of tested StyleGAN implementations

Training We trained the StyleGAN using 8,000 images from the ChestX-ray8 dataset. The training data was randomly selected and is comprised of both images with nodules and healthy images. We trained the model using a single NVIDIA A100 Tensor Core GPU for 1.3k iterations.

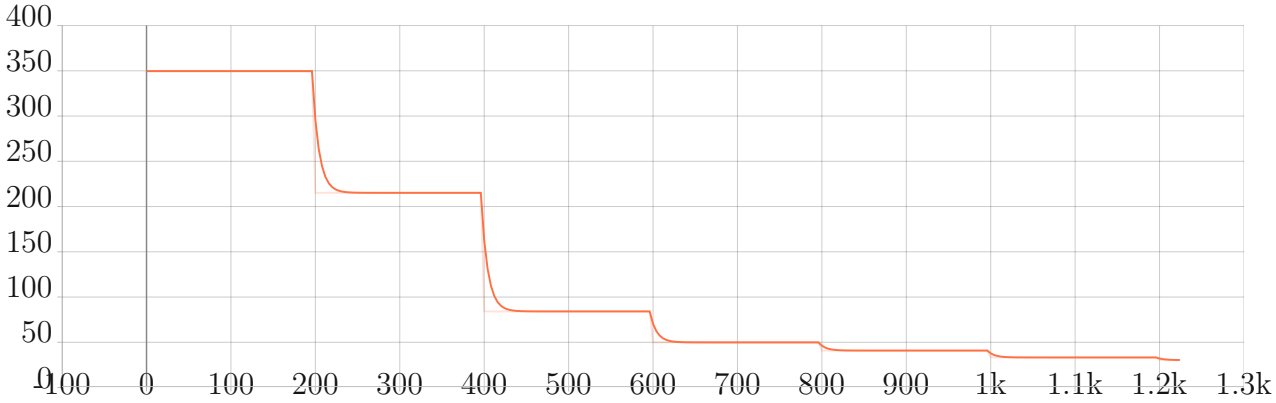

Figure 13: StyleGAN: FID50 Score (logged every 100 iterations)

Results Already after 1,000 iterations, StyleGAN was able to produce synthetic images in good quality. A sample of the generated images can be viewed below. The FID score

quickly decreased to about 40.

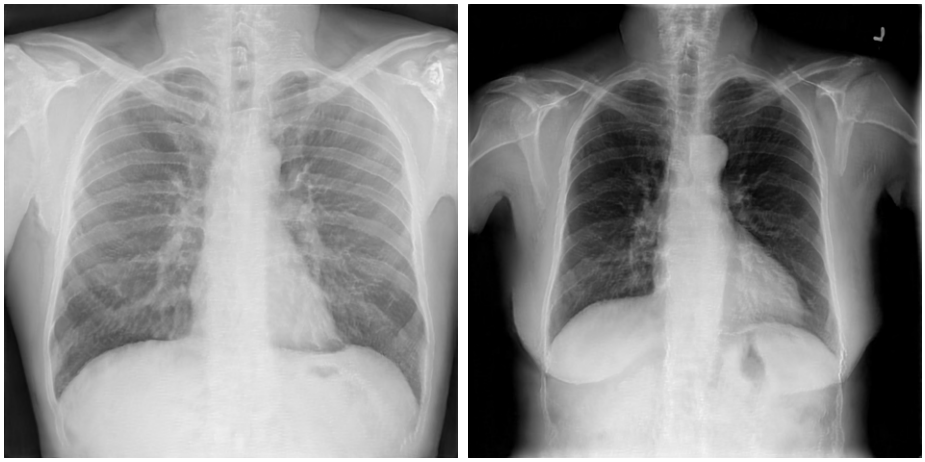

Figure 14: StyleGAN generated synthetic images

5.3 Stage II: Testing framework

To allow for the comparison of the generated training images, we developed the previously mentioned testing framework. In this section, we will outline the design of the framework and introduce the implemented components.

5.3.1 Framework design

The testing framework trains an object detection model on a provided set of training data to evaluate the model performance on a ground-truth test dataset. In the below figure, training data generated using three methods is fed into the framework that yields comparable model performance results. The framework allows to assess the end-to-end impact of the generated training data on the model performance. Thus, it allows us to give a recommendation on the method to pursue for generating segmented synthetic training data.

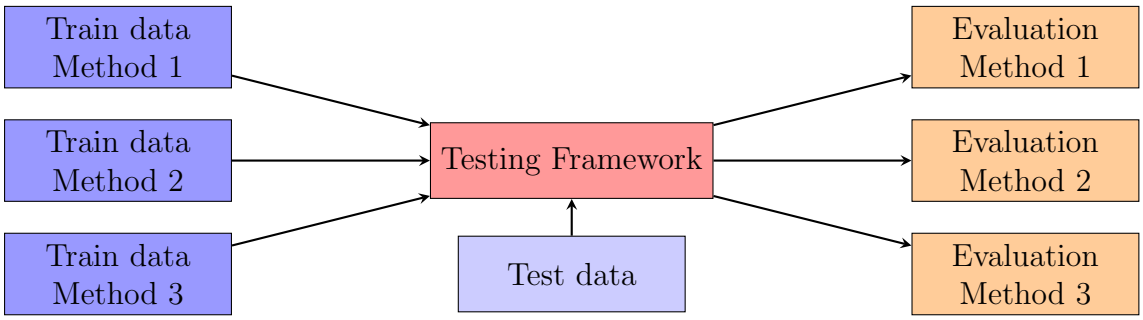

Figure 15: Testing framework high-level overview

5.3.2 Framework architecture

The testing framework is built around an abstract training pipeline that allows to plug-in any bounding-box based object detection model. For this report, we focused on RetinaNet

for the object detection, as Schultheiss et al. have shown in 2021 that the model can perform at comparable error rates to a trained human radiologist [31]. In a first step, the model is trained on the provided training data generated via the respective method, as well as a list of bounding boxes, indicating the position of the tumours on the training images. In a second step, we use the trained model to perform an inference step on a set of test images to predict the location of the tumours. For this project, we used the 70 ground-truth images, as well as the bounding boxes of the respective tumours, provided by the NIH Chest-8 dataset. In a final step, we compare the predicted tumour locations with the actual tumour location and calculate our evaluation metrics.

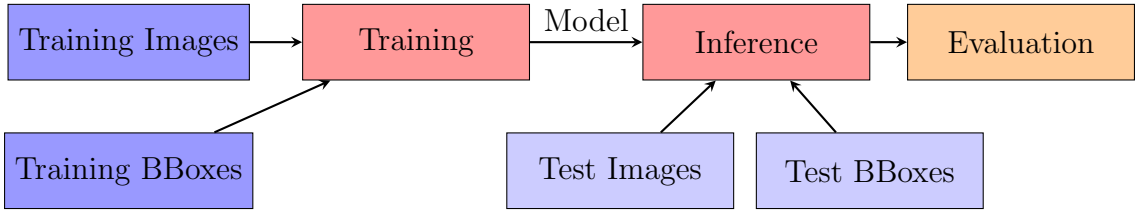

Figure 16: Testing framework architecture

5.3.3 Implementation

The testing framework is implemented in Python and allows for interaction via a command-line interface (CLI). The program is invoked by calling the main function via the CLI `python main.py`. This would start the framework with the default training dataset, the default benchmarking model as well as the default parameters. However, by passing parameters to the CLI, different testing configurations can be experimented with. The following parameters are available via the CLI:

`--sources`. Allows to provide a list of methods to benchmark and compare. For each method, a CSV needs to be provided that contains a list of exact paths to the training images as well as the coordinates of the bounding box indicating the tumour location on the image.

`--benchmarks`. Allows to provide a list of benchmarking models to train on the provided training datasets. As per default, a RetinaNet model is trained. For the RetinaNet model, we based our framework on the PyTorch implementation based on the original paper, using a ResNet35 backbone. The original implementation can be found here [2]. However, due to some outdated dependencies, we forked the original implementation and performed some minor bugfixes. The implementation we used can be found here [18].

`--accelerator`. Allows to specify the PyTorch backend the benchmarking model is trained on. As per default, the RetinaNet implementation only supports a GPU as a PyTorch backend. To allow faster development cycles and local testing, we extended the implementation to also support CPUs and the newly released Apple M1 PyTorch backend. The default backend is `cpu`. The previously mentioned backends can be activated via `gpu` and `mps` respectively.

`--num_epochs`. Allows to specify the number of epochs the benchmarking model is trained for. The default is set to 50.

`--pretained_model_path`. Allows to specify a path to a PyTorch checkpoint to continue the benchmarking model training from. This parameter can be used only when a single benchmarking model and training dataset is specified.

While the CLI provides a good interface to start the testing framework, it lacks visual capabilities to allow for a comprehensive overview on the different methods. Therefore, we integrated the open-source model management platform MLflow*. When starting a new testing framework run via the CLI, we automatically start a new MLflow run for each training dataset the benchmarking model is trained on. MLflow allows to log training parameters, evaluation metrics as well as the model itself to a respective run. When starting the testing framework, we directly log parameters such as the name of the training dataset, the number of images in the dataset as well as the ratio of the train-test split. After the model training during the inference on the test images, we further log all evaluation metrics to the respective MLflow run. MLflow provides a simple graphical user interface (GUI) that can be started via `mlflow ui` in the testing framework directory. The figure below provides and example of the MLflow UI with various runs of the testing framework logged.

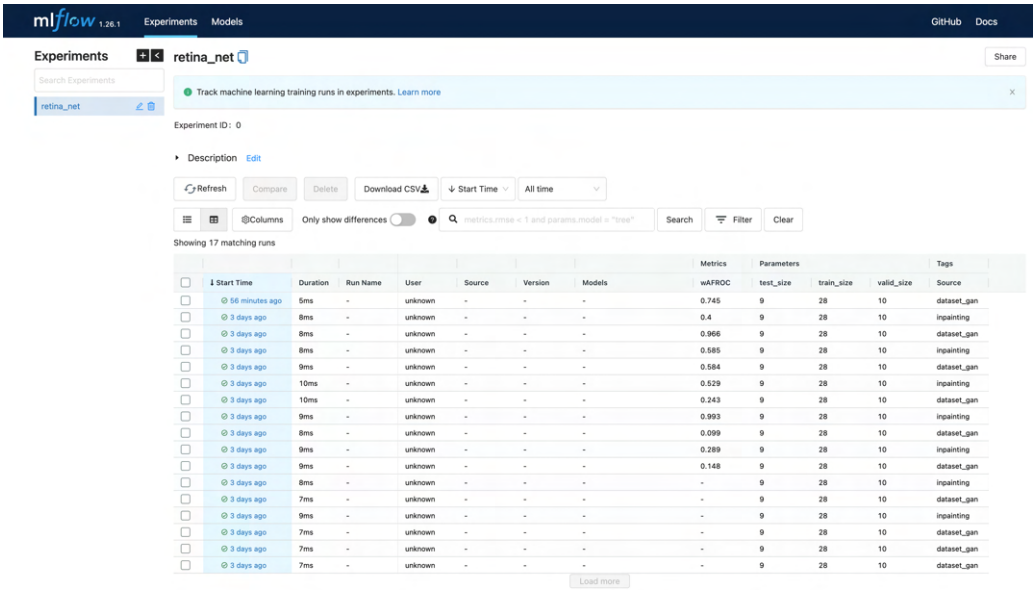

Figure 17: MLflow UI: Comparison of the performance of various training dataset and benchmarking model combinations

For the interaction with MLflow, we built a custom gateway that abstracts many of the complex functions into simple API calls. The implementation of the MLflow gateway can be found here [17]. The implementation of the testing framework with a more detailed usage guide can be found here [19].

5.3.4 Results

To assess the impact of the training images generated in stage I on the performance of the benchmarking model, we evaluated each method using the testing framework. We

*https://www.mlflow.org

implemented a custom logic to classify the predicted nodule bounding boxes according to their overlap with the ground-truth bounding box. Once the overlap is over a certain threshold, the prediction is classified as true positive, otherwise as false positive. This method allows us to derive a score for each method, that is comparable to the FROC score [7]. To have a baseline for comparison, we also evaluated the performance of the benchmarking model trained on the original set of segmented NIH images only. This evaluation can be found as Baseline in table 3. Further, we evaluated the performance of the images generated using the CA Inpainting method as well as the AOT-GAN Inpainting method. Both results can be found in table 3.

Testing framework results			
Method	# of training images	Test dataset	Benchmark dataset
Baseline	79	0.03	-
CA Inpainting	6.402	0.62	0.27
AOT-GAN Inpainting	6.402	0.65	0.28

Table 3: Comparison of the testing framework results

5.4 Discussion

First of all, the results allow to confirm the initial hypothesis of this report: Training an object detection model on a very small dataset does not yield good results (see the Benchmarking method). Second, both the object detection model trained on the dataset generated using the CA inpainting and the AOT-GAN inpainting method respectively perform significantly better. To allow for comparability between the two methods, we have inpainted the same healthy images using the nodules extracted via the respective inpainting method. The results show that the images inpainted with the AOT-GAN extracted tumors slightly outperform the images inpainted with the CA tumors. This might be due to the more granular inpainting of the AOT-GAN and the resulting sharper tumor contour that can be placed on the images. However, both methods seem to perform significantly worse on the benchmarking dataset, compared to the test holdout dataset of the respective method. This might be due to the bad visibility of the tumors on some of the images. During the inpainting process, we have manually removed some of the extracted tumors that have barely been visible. These might be the tumors that cause the drop in performance [†]. Further, there is a possibility that the object detection model overfitted to the exact characteristics of the tumors we placed on the healthy images. We tried to mitigate this effect by applying random transformations (such as rotation and scaling) before inpainting the tumors. An example of a good and a bad prediction by the object detection model trained on the AOT-GAN dataset can be found in figure 18 and 19 respectively. More images can be found in the appendix in section four.

To place the observed results into a larger context, we want to recap the performance of a human radiologist and an object detection model that was trained on even more images in a closed clinical trial [31]. Schultheiss et al. have found the human evaluator

[†]Note: the benchmarking dataset contains only the described 79 nodules, whilst the test holdout set contained around 900 images.

to perform at a FROC score between 0.54 and 0.87, whilst the best performing object detection model performed at a score of 0.81. While the object detection models trained on the synthetic images do not allow to reconstruct these results, they show the potential of augmenting healthy images with tumors extracted from very few segmented images.

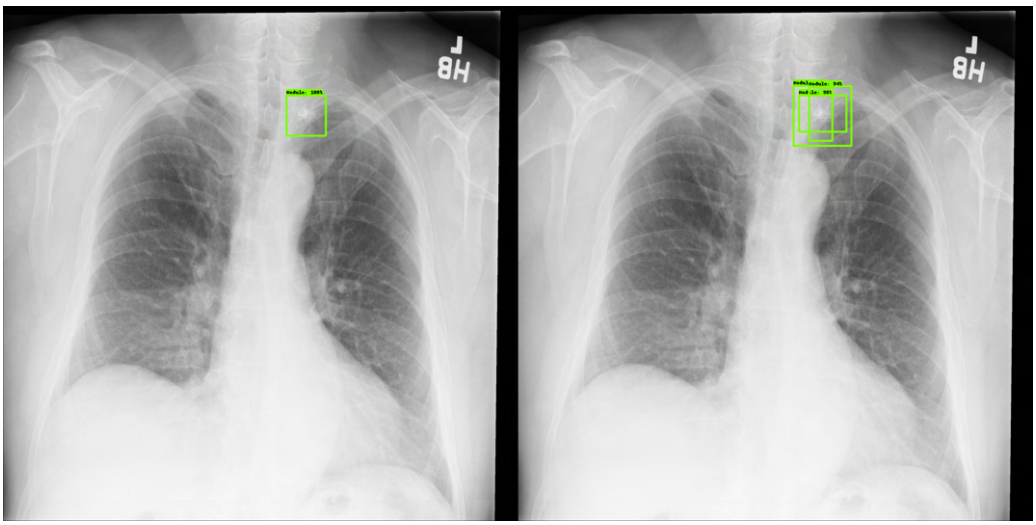

Figure 18: Good prediction: Comparison of the ground truth location (left) with the predicted tumor location (right) of the AOT-GAN inpainting method

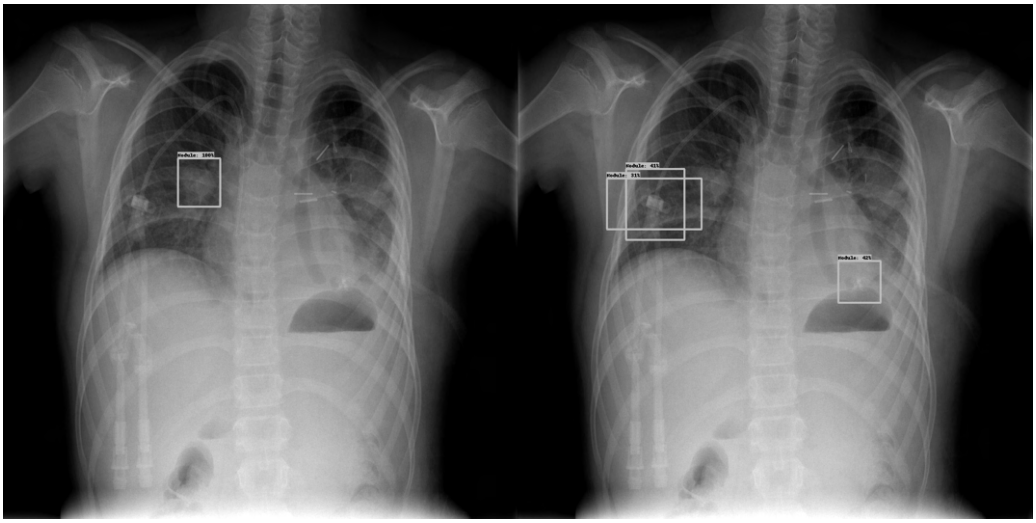

Figure 19: Bad prediction: Comparison of the ground truth location (left) with the predicted tumor location (right) of the AOT-GAN inpainting method

6 Conclusion & Outlook

To conclude, we have successfully shown how the bottleneck of segmented tumorous images for the training of object detection models can be mitigated. Based on a publicly available dataset with only very few segmentation's, we have been able to construct a synthetic dataset that significantly improves an object detection models performance. Further, we have clustered individual methods to identify three distinct approaches to generating a synthetic segmented dataset: Post-generation segmentation, post-generation inpainting as well as the segmented-generation. We have demonstrated the effectiveness of the second approach, and innovated by experimenting with more recent inpainting methods. Further, we conducted experiments with the segmented-generation approach, and identified a lack of reliable implementations.

As next steps, we propose to build an end-to-end pipeline based on the inpainting approach, for the application in an industrial setting. As part of this pipeline, we would recommend to extend our implementation by the segmentation of organs. This would allow to direct the system to inpaint tumors not only in previously specified regions, but in the exact location of an organ in an image. However, we would recommend to remain the "human-in-the-loop" approach and involve trained medical personnel in the process of evaluating the extracted nodules and assessing the quality of the inpainted images.

Aside from the inpainting approach, we are optimistic that the compatibility gap of segmented-generation methods can be overcome. As shown shown in our experiments, these methods have the potential to leverage synergies between the task of image synthesis and segmentation. Aside from that, these methods currently draw a lot of interest in the research community which promises relevant and groundbreaking innovations.

Lastly, we want to thank our mentors and supervisors for the continuous support over the whole period of the project.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: ed. by Doina Precup and Yee Whye Teh. Vol. 70. PMLR, July 2017, pp. 214–223. URL: https://proceedings.mlr.press/v70/arjovsky17a.html.

[2] benihime91. *RetinaNet*. Aug. 2021. URL: https://github.com/benihime91/pytorch_retinanet#40220b1.

[3] A.H. Bermano et al. "State-of-the-Art in the Architecture, Methods and Applications of StyleGAN". In: *Computer Graphics Forum* 41 (2 May 2022), pp. 591–611. ISSN: 0167-7055. DOI: 10.1111/cgf.14503.

[4] Dongmin Cha and Daijin Kim. "DAM-GAN : Image Inpainting using Dynamic Attention Map based on Fake Texture Detection". In: (Apr. 2022).

[5] *Computed Tomography (CT) Scan — Johns Hopkins Medicine*. URL: https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/computed-tomography-ct-scan.

[6] *Data From LIDC-IDRI*. DOI: 10.7937/K9/TCIA.2015.LO9QL9SX.

[7] James P. Egan, Gordon Z. Greenberg, and Arthur I. Schulman. "Operating Characteristics, Signal Detectability, and the Method of Free Response". In: *The Journal of the Acoustical Society of America* 33 (8 Aug. 1961), pp. 993–1007. ISSN: 0001-4966. DOI: 10.1121/1.1908935.

[8] Lukas Fetty et al. "Latent space manipulation for high-resolution medical image synthesis via the StyleGAN". In: *Zeitschrift für Medizinische Physik* 30 (4 Nov. 2020), pp. 305–314. ISSN: 09393889. DOI: 10.1016/j.zemedi.2020.05.001.

[9] Ian J. Goodfellow et al. "Generative Adversarial Networks". In: (June 2014). DOI: 10.48550/arxiv.1406.2661. URL: http://arxiv.org/abs/1406.2661.

[10] Sebastian Guendel et al. "Extracting and Leveraging Nodule Features with Lung Inpainting for Local Feature Augmentation". In: (Aug. 2020).

[11] Jeremy Irvin et al. "CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison". In: (Jan. 2019). URL: http://arxiv.org/abs/1901.07031.

[12] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: (Nov. 2016).

[13] JiahuiYu. *Contextual Attention Inpainting*. Oct. 2020. URL: https://github.com/JiahuiYu/generative_inpainting#3a53243.

[14] *JSRT Database — Japanese Society of Radiological Technology*. URL: http://db.jsrt.or.jp/eng.php.

[15] karl-richter. *AOT-GAN*. July 2022. URL: https://github.com/karl-richter/AOT-GAN-for-Inpainting#2b30c0f.

[16] karl-richter. *Contextual Attention Inpainting*. July 2022. URL: https://github.com/karl-richter/generative_inpainting.

[17] karl-richter. *MLflowGateway*. July 2022. URL: https://github.com/karl-richter/data-innovation-lab/blob/master/testing_framework/mlflow_gateway.py.

[18] karl-richter. *RetinaNet*. July 2022. URL: https://github.com/karl-richter/pytorch_retinanet#b6ea0ef.

[19] karl-richter. *Testing framework*. July 2022. URL: https://github.com/karl-richter/data-innovation-lab/tree/master/testing_framework.

[20] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: (Dec. 2018).

[21] Tero Karras et al. "Alias-Free Generative Adversarial Networks". In: *arXiv e-prints* (June 2021), arXiv:2106.12423.

[22] Christian Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: (2017).

[23] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *CoRR* abs/1708.02002 (2017). URL: http://arxiv.org/abs/1708.02002.

[24] Xudong Mao et al. "Multi-class Generative Adversarial Networks with the L2 Loss Function". In: *CoRR* abs/1611.04076 (2016). URL: http://arxiv.org/abs/1611.04076.

[25] *MRI - Mayo Clinic*. URL: https://www.mayoclinic.org/tests-procedures/mri/about/pac-20384768.

[26] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. "f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization". In: ed. by D Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/cedebb6e872f539bef8c3f919874e9d7-Paper.pdf.

[27] Deepak Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: (Apr. 2016).

[28] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: (Nov. 2015).

[29] researchmm. *AOT-GAN*. July 2022. URL: https://github.com/researchmm/AOT-GAN-for-Inpainting#4180346.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "LNCS 9351 - U-Net: Convolutional Networks for Biomedical Image Segmentation". In: (2015). DOI: 10.1007/978-3-319-24574-4_28. URL: http://lmb.informatik.uni-freiburg.de/http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net.

[31] Manuel Schultheiss et al. "Lung nodule detection in chest X-rays using synthetic ground-truth data comparing CNN-based diagnosis to human performance". In: *Scientific Reports* 11 (1 Dec. 2021), p. 15857. ISSN: 2045-2322. DOI: 10.1038/s41598-021-94750-z.

[32] Rebecca L. Siegel et al. "Cancer Statistics, 2021". In: *CA: A Cancer Journal for Clinicians* 71 (1 Jan. 2021), pp. 7–33. ISSN: 0007-9235. DOI: 10.3322/caac.21654.

[33] Ecem Sogancioglu et al. "Chest X-ray Inpainting with Deep Generative Models". In: (Aug. 2018).

[34] Xiaosong Wang et al. "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases". In: (May 2017). DOI: 10.1109/CVPR.2017.369. URL: http://arxiv.org/abs/1705.02315%20http://dx.doi.org/10.1109/CVPR.2017.369.

[35] *X-Rays — Johns Hopkins Medicine*. URL: https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/xrays.

[36] Tao Xu et al. “AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks”. In: (2018). URL: https://github.com/taoxugit/AttnGAN.

[37] Raymond A. Yeh et al. “Semantic Image Inpainting with Deep Generative Models”. In: (July 2016).

[38] Xin Yi, Ekta Walia, and Paul Babyn. “Generative Adversarial Network in Medical Imaging: A Review”. In: (2019).

[39] Jiahui Yu et al. “Generative Image Inpainting with Contextual Attention”. In: (Jan. 2018).

[40] Yanhong Zeng et al. “Aggregated Contextual Transformations for High-Resolution Image Inpainting”. In: (Apr. 2021).

[41] Yuxuan Zhang et al. “DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort”. In: (Apr. 2021). DOI: 10.48550/arxiv.2104.06490. URL: http://arxiv.org/abs/2104.06490.

[42] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.

Appendix

Section 1: NIH ChestX-ray8 dataset

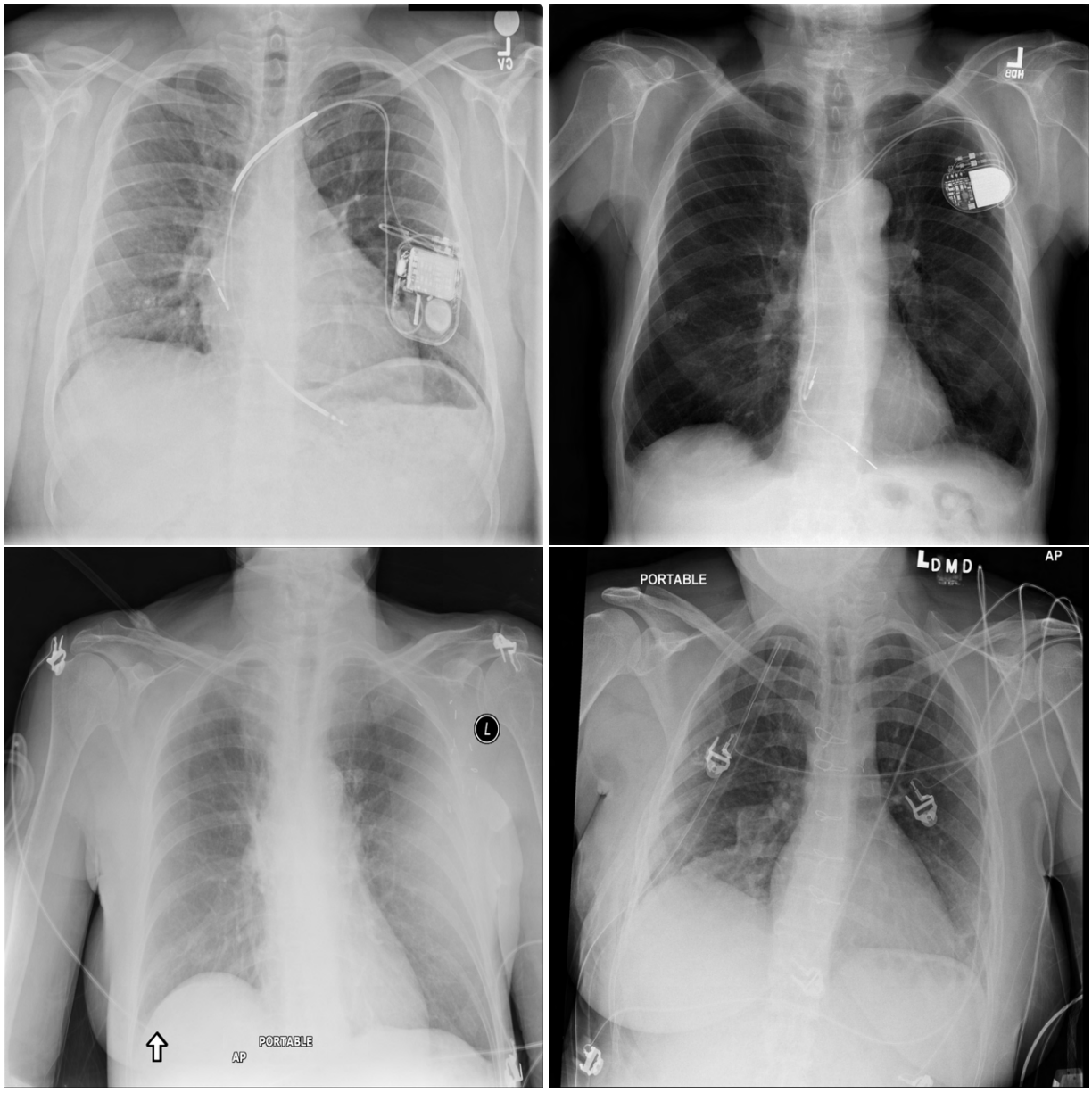

Figure 20: NIH ChestX-ray8: Exemplary images containing cables and pacemakers

Section 2: Inpainting

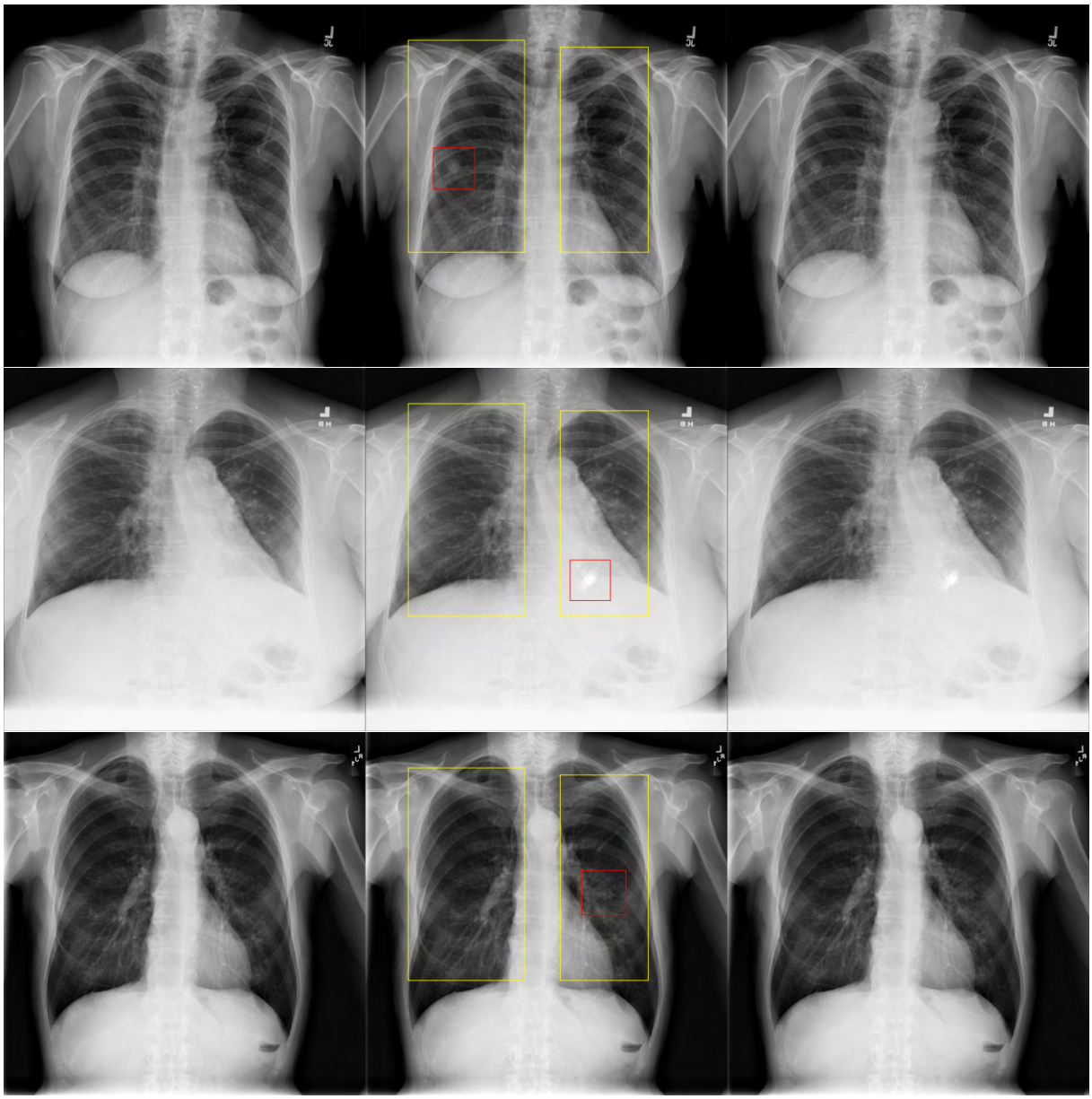

Figure 21: CA Inpainting: (A) Healthy, (B) Inpainted with bounding boxes, (C) Inpainted

Section 3: StyleGAN

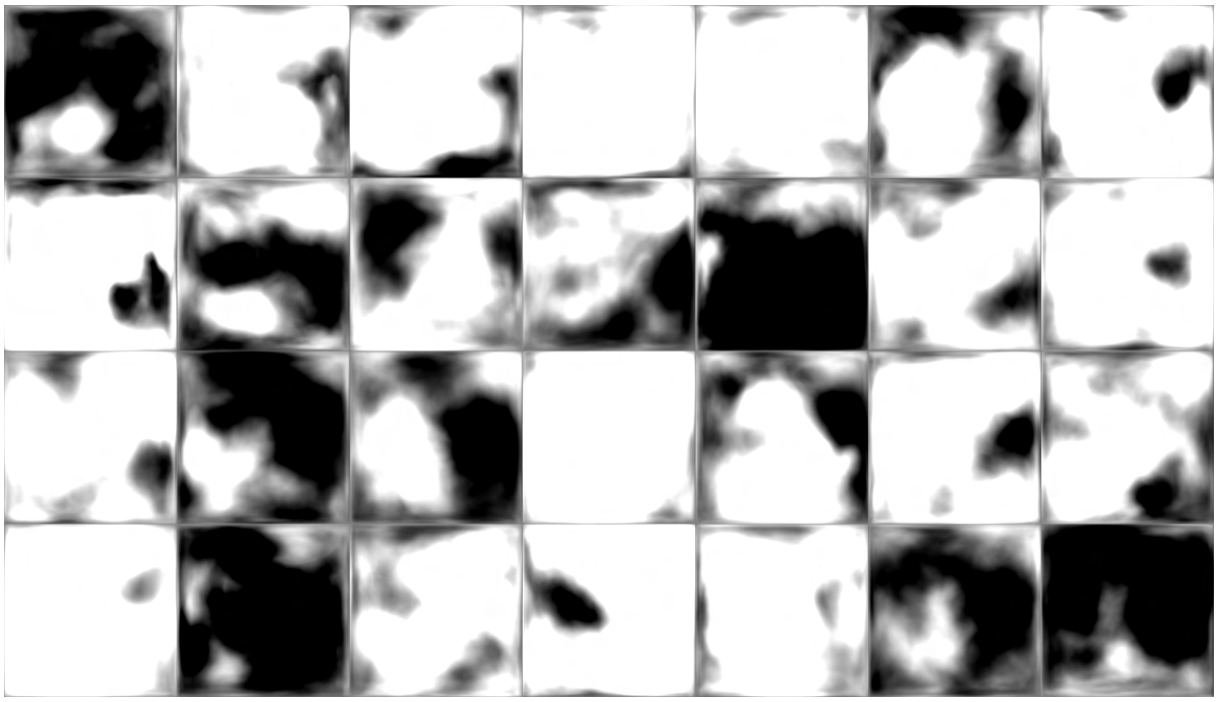

Figure 22: StyleGAN: Initial Model before training

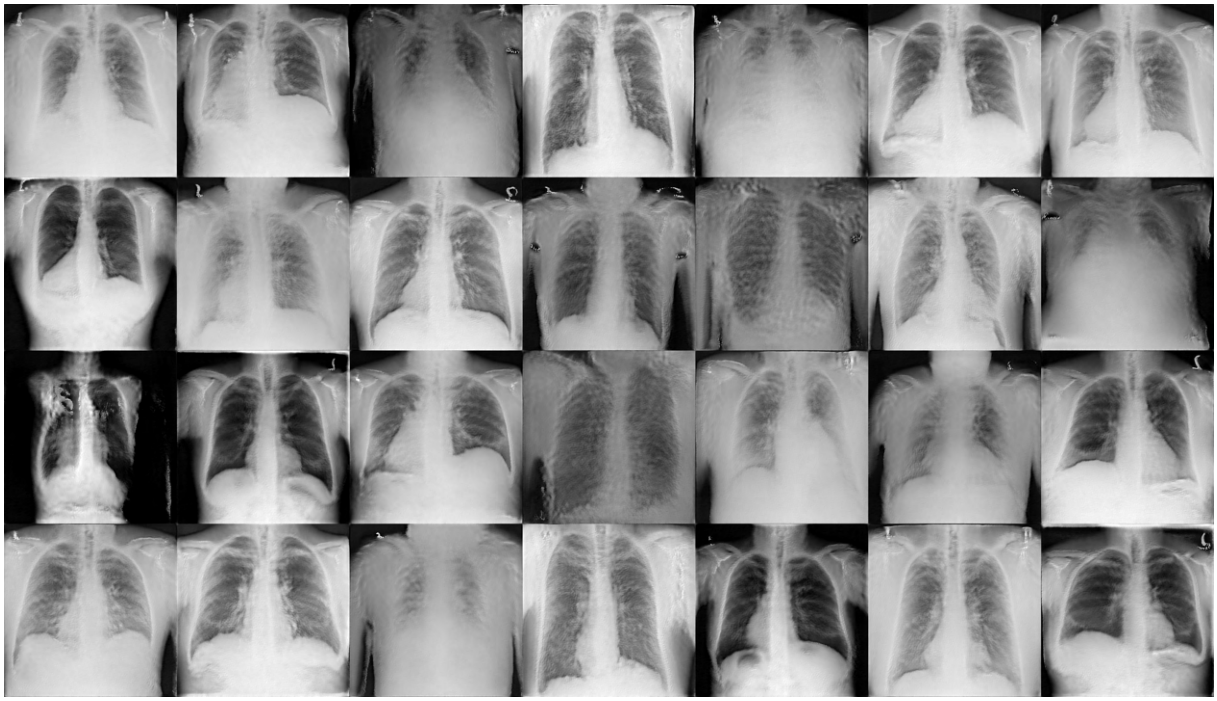

Figure 23: StyleGAN: Generated Images after 200 iterations

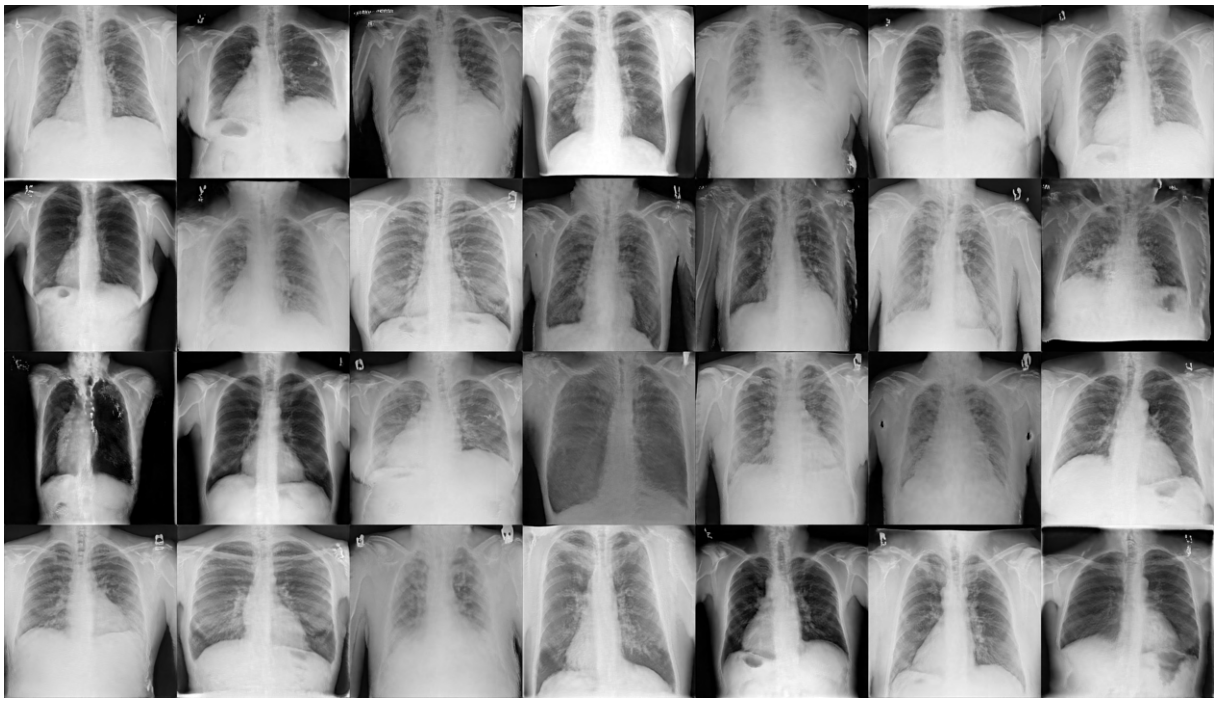

Figure 24: StyleGAN: Generated Images after 400 iterations

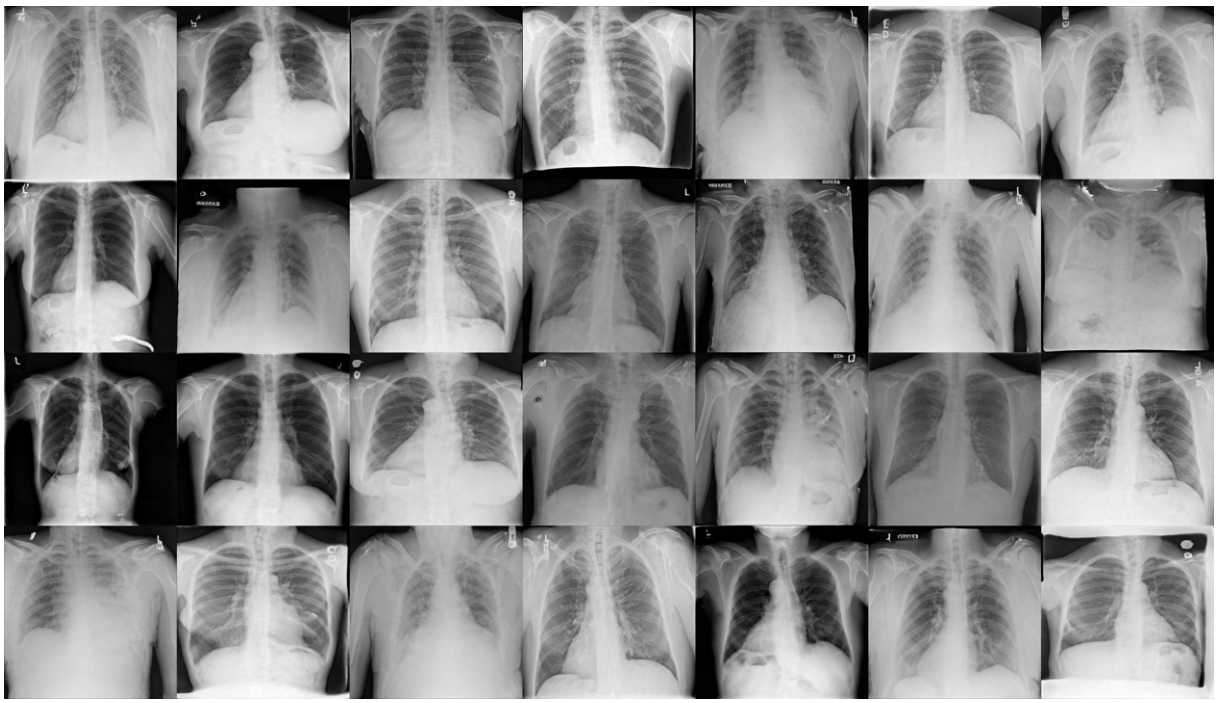

Figure 25: StyleGAN: Generated Images after 800 iterations

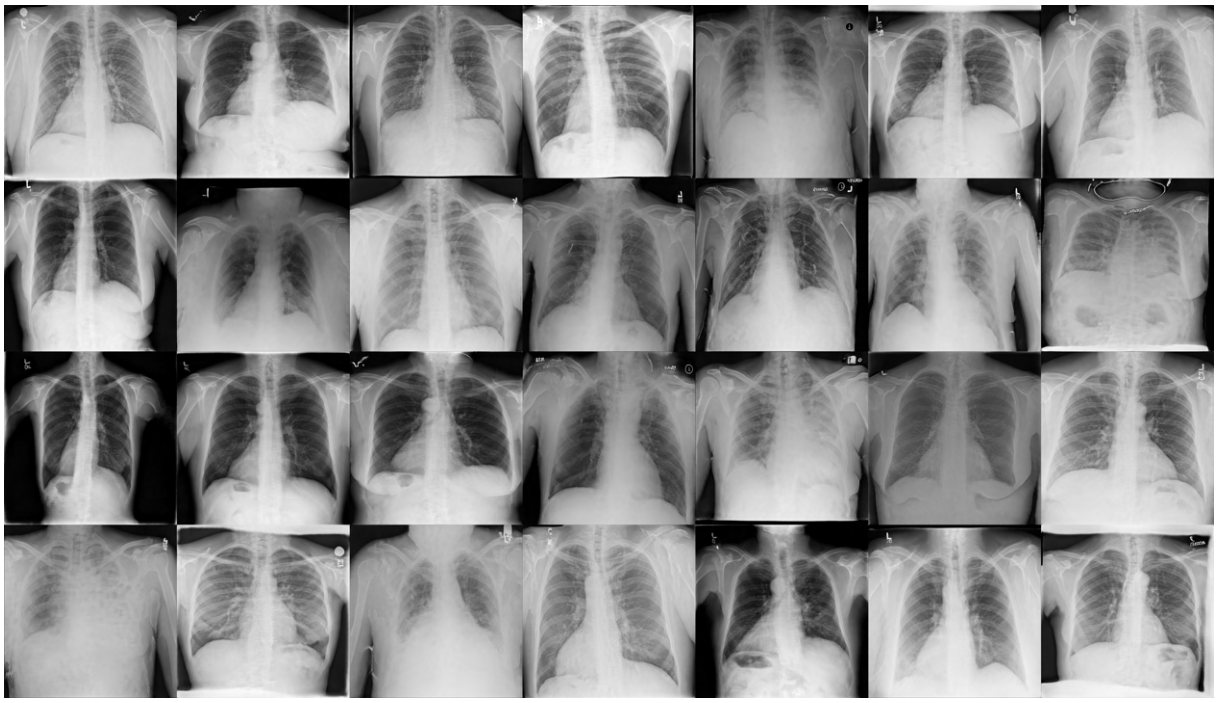

Figure 26: StyleGAN: Generated Images after 1200 iterations

Section 4: RetinaNet Predictions

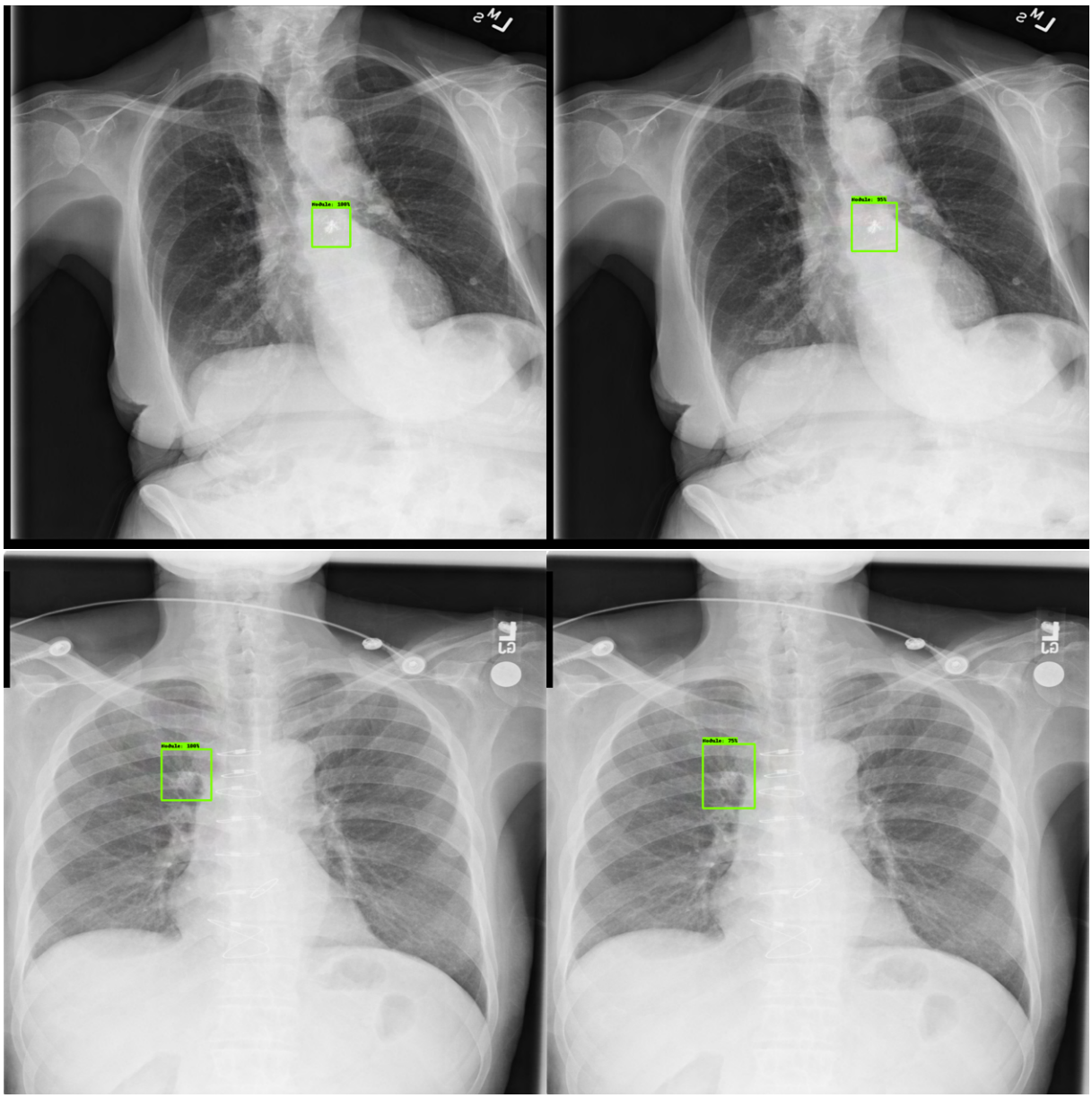

Figure 27: RetinaNet Prediction: Ground truth tumor location (left) and the predicted tumor location (right) of the AOT-GAN inpainting method

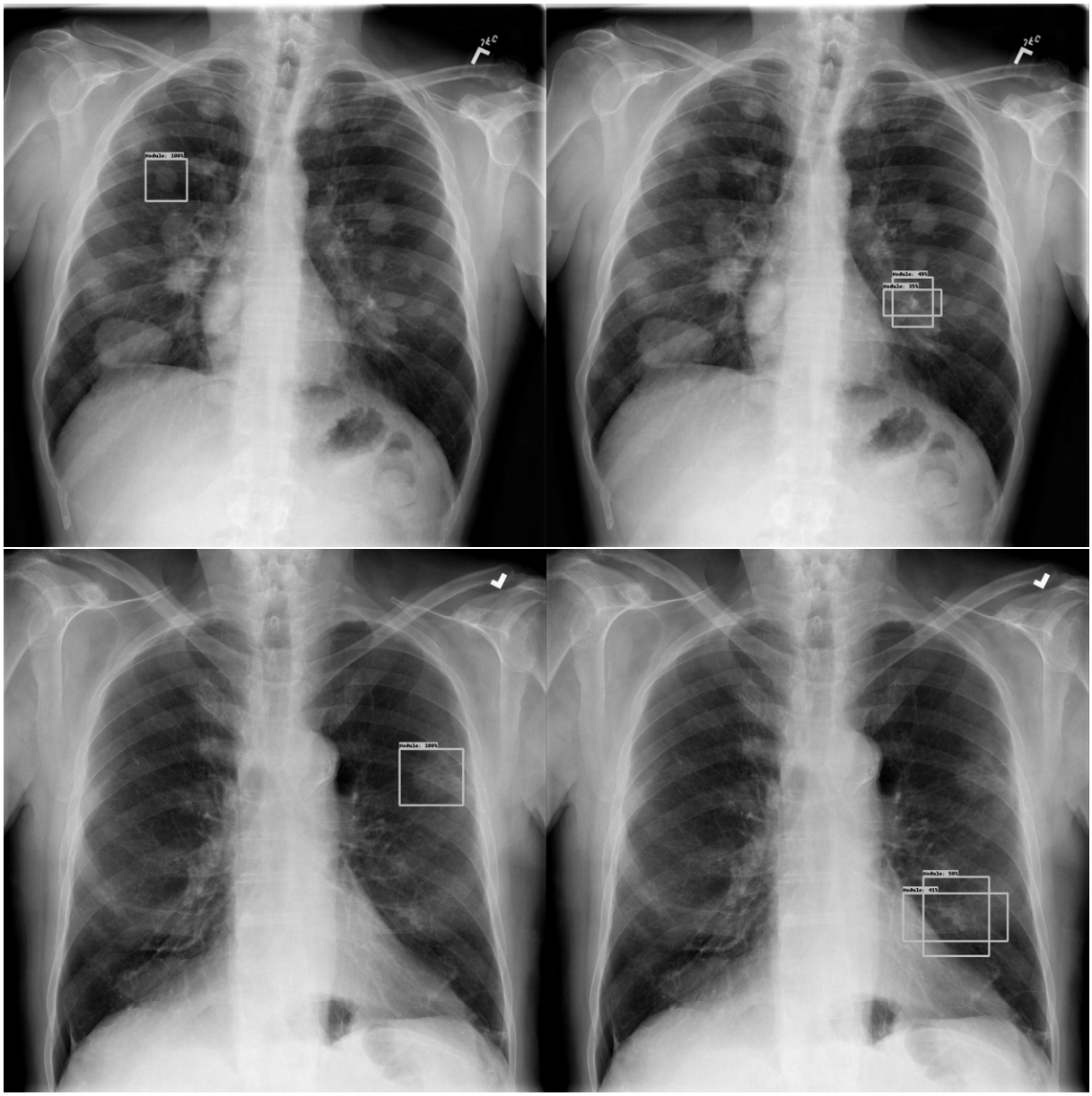

Figure 28: RetinaNet Prediction: Ground truth tumor location (left) and the predicted tumor location (right) of the AOT-GAN inpainting method