DATA AUGMENTATION ON CHEST X-RAYS FOR IMPROVING PATHOLOGY CLASSIFICATION PERFORMANCE

by

Onur Adıgüzel

B.S., Computer Engineering, Middle East Technical University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2022

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Lale Akarun and my co-advisor Dr. Pınar Yanardağ Delul for their patience, motivation, immense experience and knowledge.

I would like to thank my jury members Prof. Olcay Taner Yıldız and Assist. Prof. İnci Meliha Baytaş for accepting to be in my thesis committee, and devoting their time to read the dissertation.

I would like to give my thanks to Enis Simsar to help our heatmap based inpainting method, Umut Kocasarı for helping Conditional GANSpace method, Yasin Durusoy for medical support and helping to find X-ray directions in GANSpace and Dr. Görkem Durak for his time to listen our works in this thesis and giving us valuable feedback.

I would also like to thank my family and my close friends who always give me emotional support whenever I need. They have lived all this journey with me and are always there for me.

In the middle of my M.S. years, COVID-19 crisis has arisen. We have lived with great fear in our homes for a while. I would like to thank all the people who work in the hospitals and put themselves at risk without any doubt. I would like to thank the scientists that have been working to find vaccines and other treatments by giving everything they have.

ABSTRACT

DATA AUGMENTATION ON CHEST X-RAYS FOR IMPROVING PATHOLOGY CLASSIFICATION PERFORMANCE

In recent years, deep learning techniques have made great progress. We can see applications of it in many fields such as economics, military, healthcare, and so on. Healthcare, in particular, is one of the most critical of these areas. While the world population is growing every day, healthcare professionals need more computerized technologies to make things faster. Proposed new methods are making important contributions to the healthcare system, but the lack of data is limiting development. Privacy issues prevent more patient data from being collected to use for training models. For example, chest X-rays are commonly used in pathology classification. However, studies are limited due to the lack of public datasets. To solve this problem, we focus on data augmentation on chest X-rays to improve pathology classification results. To this end, we demonstrate three methods. In the first, we propose a heatmap based image inpainting that uses X-ray images with observations and inpaints the large healthy areas to create new X-rays while preserving the labels. The second proposed method synthesizes images using an extended version of GANSpace by adding a conditional generator StyleGAN2-ADA. Finally, we demonstrate the manipulation of real and healthy X-ray images using latent space manipulation and GAN inversion. Our quantitative experiments show that heatmap based inpainting improves classification results from 86.1% to 87.7%. To provide a basis for our Conditional GANSpace method, the results of X-ray image generation experiments using StyleGAN2-ADA are also provided. The classification result of the dataset augmented using StyleGAN2-ADA is 87.36% and our Conditional GANSpace improves this result with the highest result of 88.5%.

ÖZET

PATOLOJİ SINIFLANDIRMA PERFORMANSINI GELİŞTİRMEK İÇİN GÖĞÜS X-RAY FİLMLERİNDE VERİ ÇOĞALTMA

Son yıllarda derin öğrenme teknikleri müthiş gelişme göstermektedir. Ekonomi, askeri, sağlık ve birçok alanda uygulamalarını görebiliriz. Özellikle sağlık alanı en kritiklerinden birisidir. Dünya nüfusu her gün hızla artarken, sağlık çalışanları işleri hızlandırabilmek için teknolojiye daha fazla ihtiyaç duymaktadır. Bu sebeple üretilen yeni yöntemler ciddi katkılar sağlamakta ama veri yetersizliği fazlasına engel olmaktadır. Modelleri daha fazla eğitmek için kullanılacak veriler kişisel verilerin gizliliği sebebiyle toplanamamaktadır. Örneğin, göğüs X-ray'leri patoloji sınıflandırması için sıklıkla kullanılmaktadır. Bunun üzerine çalışılan derin öğrenme yöntemleri ise sınırlı kalmaktadır çünkü çok az verikümesi bulunmaktadır. Bu problemi çözmek için göğüs X-ray'lerindeki patoloji sınıflandırma sonuçlarını arttırmak amacıyla veri çoğaltma konusuna odaklandık. Sunduğumuz ilk yöntem ısı haritası tabanlı imge tamamlama yöntemi. Bu, X-ray'lerdeki sağlıklı bölgenin büyük bir bölümünün tamamlanmasıyla yeni X-ray'ler oluşturmuş oluyor. Böylece, X-ray'in etiketi de korunmuş oluyor. İkinci yöntemde ise, koşullu üretici StyleGAN2-ADA eklenmiş GANSpace modeli ile imge sentezleme üzerine çalışma yaptık. Son olarak, sağlıklı ve gerçek X-ray'leri vektörlere dönüştürerek GANSpace ile manipüle etme çalışmasını gösterdik. Sayısal sonuçlara bakıldığı zaman imge tamamlama yöntemi orjinal dataset ile elde edilmiş sınıflandırma sonucunu %86.1'den %87.7'ye yükseltmiştir. Koşullu GANSpace yöntemi deneylerine bir taban oluşturmak için StyleGAN2-ADA modeli ile X-ray'ler ürettik ve çoğalttığımız verilerin sınıflandırılması bize %87.36 sonucunu verdi. Sunduğumuz Koşullu GANSpace methodu ise bunu da geliştirerek en yüksek sonuç olan %88.5'i elde etti.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xi
LIST OF SYMBOLS . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. RELATED WORKS . . . 6
2.1. Image Inpainting . . . 6
2.2. Image Inpainting in Medical Field . . . 8
2.3. Latent Space Manipulation . . . 11
2.4. GAN Inversion . . . 13
2.5. Data Augmentation in Medical Field . . . 14
3. BACKGROUND . . . 17
3.1. Discriminative Models . . . 17
3.2. Generative Models . . . 19
3.3. Generative Adversarial Networks . . . 23
3.3.1. Introduction . . . 23
3.3.2. Variants of GANs . . . 25
3.3.2.1. Deep Convolutional GAN . . . 25
3.3.2.2. Wasserstein GAN . . . 26
3.3.2.3. StyleGAN . . . 27
3.3.3. Applications of GANs . . . 29
3.3.3.1. Conditional Image Generation . . . 29
3.3.3.2. Image Editing . . . 32
3.3.3.3. Image to Image Synthesis . . . 34
3.3.3.4. Image Inpainting . . . 37

3.3.4. Evaluation Metrics 40
3.3.4.1. Inception Score 41
3.3.4.2. Fréchet Inception Distance 41
3.3.4.3. Nearest Neighbor Accuracy 42
4. METHODOLOGY 43
4.1. Deep Image Inpainting 43
4.1.1. Recurrent Feature Reasoning for Image Inpainting 43
4.1.2. Heatmap Based Inpainting 46
4.2. Conditional GANSpace 49
4.2.1. Stylegan2-ADA 50
4.2.2. GANSpace 52
4.2.3. Proposed Solution 53
4.3. Manipulation of Encoded Latent Vectors 56
4.3.1. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement 57
4.3.2. Applied Method 58
4.4. Dataset 60
5. EXPERIMENTS AND RESULTS 64
5.1. Visual Results of Image Inpainting Model 64
5.2. Evaluation Of Inpainted X-rays 66
5.3. Evaluation of Conditional GANSpace 67
5.4. Visual Results of Manipulation of Encoded Latent Vectors 73
6. CONCLUSION 76
REFERENCES 78
APPENDIX A: 91

LIST OF FIGURES

Figure 3.1. Difference Between Discriminative and Generative Models in Space. 22

Figure 3.2. Visualization of GAN Architecture. 23

Figure 3.3. Overview of the StyleGAN Architecture taken from paper [1]. . . . 26

Figure 3.4. Example results of generated faces taken from paper [1]. 28

Figure 3.5. Difference between GAN vs Conditional GAN. 30

Figure 4.1. Inpainting of Chest-1 Without Artifact. 44

Figure 4.2. Inpainting of Chest-2 Without Artifact. 44

Figure 4.3. Inpainting of Chest-3 Without Artifact. 45

Figure 4.4. Inpainting of Chest-4 With Artifact. 45

Figure 4.5. Inpainting of Chest-5 With Artifact. 45

Figure 4.6. Inpainting of Chest-6 With Artifact. 46

Figure 4.7. Each row shows results for different observations and X-ray images while each column shows different steps of our method. First Column: Original X-ray Image, Second Column: Heatmap is created for 5 different observations, Third Column: Regions with observations are masked, Forth Column: Masked region is chosen randomly to inpaint, Fifth Column: Inpainted X-ray Image. 48

Figure 4.8. N samples are chosen randomly from latent space. Embedded conditional vectors are concatenated to them. Obtained vectors converted to style vectors via mapping network of pretrained StyleGAN2-ADA. PCA is applied on each style space and components are computed for each class. 55

Figure 4.9. One random sample is chosen from latent space and converted to style vector as showed in Figure 4.8. Predefined components found for each class from their own component set are added to style vectors. Manipulated vectors are given to pretrained StyleGAN2-ADA. Final output X-ray images are obtained. Output X-ray images are obtained from our augmented dataset. 55

Figure 4.10. Healthy X-ray is given to Restyle Encoder. Output style vector can manipulate with GANSpace manipulation. Without manipulation, original image (Inverted Image) can be reconstructed. In the example, heart size is manipulated in both positive and negative direction. 59

Figure 4.11. All observations with presence, absence and uncertainty numbers of CheXpert dataset. 61

Figure 4.12. Different type of radiograph examples from CheXpert dataset. . . 62

Figure 5.1. Original, randomly masked and inpainted radigraphs of patient00003. 65

Figure 5.2. Original, randomly masked and inpainted radigraphs of patient00004. 65

Figure 5.3. Original, randomly masked and inpainted radigraphs of patient00005. 65

Figure 5.4. Manipulation of cardiomegaly on X-ray generated with cardiomegaly. 68

Figure 5.5. Manipulation of consolidation on X-ray generated with consolidation.. 68

Figure 5.6. Manipulation of edema on X-ray generated with edema class. . . . 68

Figure 5.7. Manipulation of pleural effusion on X-ray generated with pleural effusion. 68

Figure 5.8. Manipulation of lung size on X-ray generated with atelectasis. . . 69

Figure 5.9. Manipulation of lung size on X-ray generated with cardiomegaly. . 69

Figure 5.10. Manipulation of artifact on X-ray generated with cardiomegaly. . . 69

Figure 5.11. Manipulation of artifact on X-ray generated with consolidation. . . 69

Figure 5.12. Qualitative results of Manipulation of Encoded Latent Vectors method. Four different directions are shown as example. First row shows changes in heart size, second one is example of lung size direction, next is pleural effusion direction and widening of shoulders-clavics. 75

Figure A.1. Permission to use visuals from official publication of StyleGAN2 [1]. 91

LIST OF TABLES

Table 4.1. Example Labels from CheXpert Dataset 63

Table 5.1. Experiments Results 70

Table 5.2. Conditional GANSpace Directions 71

LIST OF SYMBOLS

c	Condition value/vector
$D()$	Discriminator function
$\exp()$	Exponential function
$E()$	Encoder function
$G()$	Generator function
$KL()$	Kullback–Leibler divergence function
$\max_D$	Maximum value of Discriminator function
$\min_G$	Minimum value of Generator function
$p()$	Probability of something
$Tr()$	Transpose function
X	Input image
V	Direction vector
$V()$	Value function
w	Style vector
W	Style vector space in StyleGAN model
$W+$	Extended style vector space in StyleGAN model
y	Output vector/score
Y	Output image
z	Latent vector
Z	Latent space
α	Coefficient that determines effect of direction vector
Δ	Difference between two values
Σ	Sum
θ	Direction vector

LIST OF ACRONYMS/ABBREVIATIONS

3D	3-Dimensional
ACGAN	Auxiliary Classifier Generative Adversarial Network
AUC	Area Under Curve
CAM	Class Activation Mapping
cGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
COVID	COrona VIrus Disease
CPU	Central Processing Unit
CVPR	Computer Vision and Pattern Recognition Conference
DCGAN	Deep Convolutional Generative Adversarial Network
e4e	Encoder For Editing
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
GHz	Giga Hertz
GPU	Graphical Processing Unit
IS	Inception Score
KCA	Knowledge Consistent Attention
kNN	k Nearest Neighbor
LOO	Leave-One-Out
MAE	Mean Absolute Error
MSE	Mean Square Error
NIH	National Institute of Health
NLP	Natural Language Processing
PCA	Principle Component Analysis
PCAM	Probabilistic Class Activation Mapping
PLCO	The Prostate, Lung, Colon, Ovary
pSp	pixel2style2pixel
RFR-Net	Recurrent Feature Reasoning Network

SOTA	State-of-the-art
VGG	Visual Geometry Group
WGAN	Wasserstein Generative Adversarial Network

1. INTRODUCTION

The world population is increasing day by day. This situation is leading to the emergence of serious problems. Healthcare is one of the services most affected by the unexpected population growth. The healthcare system must be carefully organized because it directly affects people's lives. Diagnosis, detection or treatment of diseases is extremely important in many cases. However, the dramatic increase in population prevents these services from being provided in the desired way and quickly. The number of patients per doctor is much higher than it should be. More patients mean that there is less time available for each examination. X-rays of the chest are often used for diagnosis or detection. According to [2], 1.5 billion chest X-ray examinations will be performed in 2018. Due to the extreme number of examinations, specialists have to work hard and take time out of their other workload to perform these examinations. This can lead to some problems, such as misinterpreting X-rays, or they may fail to detect a serious problem.

Computerized systems are playing an increasingly important role in healthcare. Especially in the last decade, deep learning has become one of the hottest topics in computer science. As a result, many techniques are also being used in the medical field. They aim to automate prediction, detection, classification or other examination methods. Analyzing blood samples, detecting heart problems and tumors, or diagnosing cancer are some concrete examples of deep learning applications in healthcare. Ortega et al. published a study on breast cancer detection [3]. Sun et al. present a deep model [4] for lung cancer detection. Multiple sclerosis is a disease that severely damages nerves and brain. Zhung et al. is working on a method for multiple sclerosis detection by classification [5]. [6] is published by Lakhani et al. to detect pulmonary tuberculosis by deep learning.

Deep learning models require huge amounts of data to successfully generalize their example domain. People generally think that deeper models can learn better,

but this is not always true. Amount of data is just as important as model depth. Deep models can overfit when trained with small datasets and memorize the input dataset. With more powerful systems, deeper neural network models can be trained. For this purpose, researchers need more and more data in each study. Therefore, data scientists or big data experts try to obtain new data for different domains. In addition to the amount of data, data diversity is another important issue. Experts must take care to collect a wide variety of data to obtain generalized models.

Although collecting data is easy for some domains like animals, trees, flowers, and houses, there are some others for which there are few public records, and collecting more data is impossible due to privacy concerns. For example, it is forbidden to share health data. This type of data can only be collected by hospitals, but the authorities do not allow this without the patient's consent. The lack of data prevents deep learning from improving for these types of domains. Therefore, researchers use data augmentation techniques to increase the amount of data available to them. Some simple methods such as rotating, scaling, and flipping have been used for many years. However, these do not always provide desired variations in data. Hence, studies focus on generating new images rather than manipulating existing images. Generative models are used for this task. These aim to generate unobserved and diverse images through learning.

The introduction of Generative Adversarial Networks (GANs) opens a new era for generative models and deep learning. Following the publication of Goodfellow et al. [7], many studies in this field have focused on these models, leading to a rapid improvement in the field GAN. In the short time, many different types of GAN models have been proposed and applied in various fields. Some of the different variants of GANs are Deep Convolutional GAN (DCGAN) [8], Wasserstein GAN [9], StyleGAN [1] and many others. The original GAN structure contains a generator network and a discriminator network. The generator network aims to learn the joint probability distribution of the given data for a given domain and tries to fool the discriminator network. The discriminator network, on the other hand, tries to detect whether the given input is real or generated by the generator. These two networks compete with each other and

make each other better models. While the discriminator distinguishes whether the given input is real or fake, it also gives feedback to the generator. If it finds the correct results most of the time, it means that the generator cannot do its job well. Hence, the generator optimizes itself according to the coming feedback and tries to generate more realistic images.

In addition, the other GAN models contribute to the original or earlier GANs and achieve better results or specialize for different tasks. Although Goodfellow's GAN model is revolutionary, it has drawbacks and needs improvement. Radford et al. introduced DCGAN to solve its stability problem. They replace the max-pooling layer with convolutional layer and add batch normalization to make it more stable. Wasserstein GAN introduces a new metric called Earth Mover (Wasserstein distance) for its discriminator. This provides a real valued output to evaluate how much fake or real a generated image is. In other words, it measures the fakeness value of an image instead of a binary classification. Another variant StyleGAN introduces a mapping network that converts the given random input into a meaningful latent vector to make it more informative. A new method, adaptive instance normalization, is also introduced in this paper. This makes the model more stable and standardizes the model during training. The results of StyleGAN can fool the real people in addition to the discriminator network.

In this study, we aim to augment chest X-ray data to improve the classification results. To this end, we use several GAN methods. The first is a GAN based deep image inpainting model with the combination of the Probabilistic Class Activation Mapping [10] model. We create a pipeline that finds the location of the labeled observation. In this way, we can create a random mask that has no intersection with this given location. Finally, the masked region is inpainted with RFR-Net [11]. In this way, a new X-ray image is created while preserving its label. The next method is augmentation with Conditional GANSpace. This is extended version of GANSpace [12], a state-of-the-art latent space manipulation method. While GANSpace uses StyleGAN2 [1] as a generator, we have integrated StyleGAN2-ADA [13] to generate X-ray images with

observation conditions. We introduce this method to create a conditional GAN that also has control over the output X-ray images. Thanks to this extension, the output X-ray images have the desired features at any strength. Finally, we demonstrate the manipulation of the encoded latent vectors. We combine the study of Restyle [14] and GANSpace. While Restyle encodes images to find their latent vector, GANSpace finds directions to manipulate the original image.

The contributions of this thesis to the literature can be counted as the followings:

- We create a pipeline that is a combination of a heatmap module and an image inpainting module. The heatmap module is chosen to preserve label after inpainting operation.
- Our second method extends the GANSpace study. StyleGAN2-ADA, conditional GAN model, is integrated to the GANSpace. This allows to synthesise and manipulate X-ray images with the desired labels.
- The method of manipulating inverted images is used in the medical field. We encode healthy X-ray images and pass latent vectors to GANSpace. We can manipulate them with any direction of observation.
- We improve pathology classification results with two different augmentation methods. Heatmap-based inpainting improves the score by 1.6% and achieves a final score of 87.7%. Conditional GANSpace also improves this score by 0.7% more and reaches 88.5%.
- Experiments show that our Conditional GANSpace method, which provides controllable conditional image generation, gives better results than the conditional image generation method of StyleGAN2-ADA. While result of the StyleGAN2-ADA is 87.35%, our Conditional GANSpace method improves it by 1.12% and got 88.47%. This means that controlling the features of X-ray images during generation improves classification results and contributes to data augmentation.
- We show how powerful GANs are even in the medical field and data augmentation task. Our proposed methods heatmap based inpainting and conditional GANSpace improve the score of classification by augmenting the CheXpert dataset.

The rest of the work is divided into 5 further chapters:

Chapter 2 contains an extensive literature review. It contains recent studies on the techniques used in this thesis.

Chapter 3 explains the theoretical background information on the topics used in this thesis. In this chapter, brief definitions, examples, and visual representations are added to make the studies more understandable.

Chapter 4 contains the complete methodology with all details. All modern models used are explained in detail. All extensions and contributions to the literature are included in this chapter.

Chapter 5 shows the experiments and their results for the proposed models. It also gives details about the dataset. All qualitative and quantitative results are demonstrated in this chapter.

Chapter 6 is the conclusion of the thesis. It summarizes the thesis and gives an overall view of the study. It also includes some possible future work.

2. RELATED WORKS

In this section, we overview state of the art researches related to our study. These are image inpainting, latent space manipulation, GAN inversion, and data augmentation in the medical domain.

2.1. Image Inpainting

Inpainting (also called completion or retouching) is the process of filling the missing, removed, or damaged part in an image. One of the main goals of this is to make sure that inpainted area is not noticeable and looks as realistic as possible. Researchers have been focusing on the image painting task for almost the last two decades. One of the first papers was published by Bertalmio et al. [15] in 2001. After that, many other methods in the same field have been improved [16–18] in order to inpaint images more realistically and less prone to error. These methods include statistical and patch-based operations.

In recent years, with the great advances in Deep Learning, researchers have applied methods based on neural networks. One of them is the context encoders [19] proposed by Pathak et al. They aim to fill in missing parts from their environment by using a convolutional neural network. In their proposed method, they use a classical encoder-decoder method. The encoder aims to extract fixed length features into the latent space and the decoder tries to generate the inpainted image by using the features. They use AlexNet for the encoder part. The decoder needs additional layers to pass on the relationship between pixels. The encoder is connected to the decoder by channel-wise, fully connected layers to pass the information. Up-convolutional layers receive the features one by one and try to reproduce the original image with inpainted version. Reconstruction loss (L2 norm) and adversarial loss are used together. The reconstruction loss checks the coherence of the whole image with its context. The adversarial loss works as a discriminator in the generative adversarial network, while the

context encoder works as a generative model. By using the adversarial loss, the system tries to distinguish the original and the generated images.

Yangi et al. [20] introduce a method that uses image content and texture constraints with joint optimization. They call this approach multi-scale neural patch synthesis. They use a deep classification network that matches and adjusts patches according to features in the middle layers. This preserves contextual structure and produces high-frequency details. Their network is an encoder-decoder convolutional neural network. The approach uses the following operations to maintain consistency. The first is that the output of the encoder-decoder framework generates global content constraints. The second is to use the similarity between the missing region and the remaining region, which provides texture constraint information via local neural patch similarity. The model uses a three-level pyramid in the multi-scale local neural patch synthesis approach. At each level, the size of the images is scaled by half and they have 3 images with different resolutions. Then they perform inpainting task by starting the most downsampled version of the image to the original image in a coarse to fine manner. The inpainting task is performed by 2 different subnetworks. One subnetwork, called the Content Network, uses an encoder-decoder framework and is trained to fill in missing areas. It uses holistic loss to compute the error in the output. The other subnetwork, the texture network, uses some layers of the pretrained VGG-19 to enhance the visual content produced by the content network. It uses local texture loss to find pixel-wise errors.

A novel approach is proposed by Yu et al. [21]. Their proposed method consists of a deep generative model that can inpaint multiple and arbitrary holes in an image. Their contribution is the context attention layer, which generates patches for missing regions by learning where the background pixels should come from. Their network consists of two subnetworks that operate coarse to fine. The first subnetwork takes the image and the mask for hole region as input and generates a coarse, inpainted output. This is a dilated convolutional network. It uses the reconstruction loss to generate a draft patch. The second subnetwork, the refinement network, uses two encoders that

work in parallel. One of them focuses on hallucinating content while the other focuses on the background feature. The context attention layer is used in the encoder that focuses on the background feature. The information provided by the context attention layer is used in the generation of the patch. In this refinement network, they use reconstruction loss with the GAN losses for global and local consistency. They also introduce a spatially discounted reconstruction loss that is different from the coarse network. This loss uses a weight mask because missing pixels close to boundaries are less problematic than the ones close to the center of the hole.

The last image inpainting method mentioned here is proposed by Yu et al. [22]. Their method consists of a network that has a dynamic feature selection mechanism that can be learned during training. All layers consist of this mechanism for each channel to extract better spatial features. This increases the quality of inpainting and color consistency. They realize that these gated convolutions involve semantic segmentation while selecting the features besides background, mask and sketch. Furthermore, they use the contextual attention model [21] for the inpainting task. After getting inpainted image from the model, they present a practical discriminator SN-PatchGAN which is fast and produces high quality inpainted images. It is also stable during training. This discriminator is based on a convolutional network. An important difference of this network is that it provides a 3D shape feature as output. GAN loss is applied to each output feature. This allows the system to focus on different locations and different semantics of the input.

2.2. Image Inpainting in Medical Field

Medical inpainting has been studied for many years. In the early studies, some classical computer vision and machine learning methods are applied to inpaint the needed areas in the X-ray images. After the great advances of deep learning, new and more successful methods have used in the literature.

Hogeweg et al. [23] have published a study that detects and removes foreign objects in chest radiographs to improve the analysis of these images. They first detect the objects by applying the K-nearest neighbor classification method to each pixel. For each group of pixels, they apply some post-processing methods and segment the foreign objects. After segmentation, they remove the objects. To fill in missing pixels, they select the best matches from the neighbor square patches by calculating the sum of square differences. This was one of the most important studies before deep learning methods are applied for this purpose. In recent years, many deep learning models have been used for X-ray inpainting.

In 2018, Sogancioglu et al. [2] proposed to apply some existing methods of image inpainting to chest radiographs. They applied three methods, namely Context Encoder [19] by Pathak et al, Semantic Image Inpainting [24] by Yeh et al, and Contextual Attention [21] based inpainting by Yu et al. The first method contains two opposing frameworks, namely encoder and decoder. Both of them are convolutional neural networks. The encoder takes the images with masks and converts them into smaller compressed data. The decoder then takes this compressed data and attempts to create the original image by filling the missing region. The second method, Semantic Image Painting, uses the DCGAN architecture for inpainting. They train the DCGAN with natural images and then use the generative model to fill in missing areas in images. Generating pixels of the missing region with this model gives better results than computing over distant pixels. The last model is Contextual Attention, which uses neighbor pixels of the missing region. It first fills the region with coarse pixels as an initial result. Then this coarse result is passed to 2 different paths, namely a dilated convolutional network and a contextual attention layer. The results of these networks are passed to another neural network and the final refined region is generated.

Armanious et al. propose a method [25] in order to inpaint MRI scans differently than chest radiographs. Although the inputs are different, the purpose is the same. It is based on a generative adversarial network (GAN), more specifically, a conditional GAN. Cascaded U-Net [26] is employed for the generator. Two discriminators are

used for different purposes. One of them is a global discriminator that focuses on the entire image, while the other, called a local discriminator, focuses only on inpainted region. The generator takes an input of size 256×256 with random masking. It inpaints the missing area using the context information of the remaining part. Then, the discriminator takes the output of the generator and the target image and tries to figure out which image is real using cross entropy. The local discriminator, on the other hand, takes only the generated region and the target region as input. This model uses additional loss functions when training the network, namely style reconstruction loss and perceptual loss. While the former loss function uses the features extracted from the whole image using VGG-19 for loss calculation, the latter concerns pixel-wise differences.

Armanious et al. recently published a new paper [27], which is an extended version of [25]. This new model can inpaint the random missing regions instead of fixed and square regions. This work takes a 256×256 image with a random mask, while previous work uses an input of the same size with a centered 64×64 mask. This model is also based on the conditonal GAN, but MultiRes-UNet [28] is chosen in the generator instead of Cascaded UNet. Similar to the previous work [25], two discriminators are used. The global discriminator focuses on finding real images by taking an inpainted image and a target image as input. The local discriminator, in turn, checks the only inpainted region with the same region in the target image. Non-adversarial losses are also used in this model. VGG-19 is used as a feature extractor to compute style reconstruction loss while the perceptual loss focuses on pixel-wise differences by computing the mean absolute error (MAE).

Another recent study was published by Le et al. [29]. They proposed a deep learning model for removing foreign objects and inpainting the missing region. For this purpose, they detect the objects in the chest radiographs and then segment the areas with foreign objects. After segmentation, the area is masked and removed from the original image. The masked object is inpainted seperately by using the Fast Marching algorithm. Finally, the inpainted areas are inserted back into the original image.

Tran et al. [30] proposed a state of the art method in 2020. Their goal is to inpaint missing or damaged regions in chest radiographs. For this purpose, they use a two-stage model with a coarse-to-fine method. The first network is based on the U-Net [31] and produces a coarse output from images with random holes. This output is passed to the second network, which is an encoder-decoder framework. It has a more complex structure than the first network as it tries to find more significant features to produce a higher quality image. The second network has a discriminator to inpaint images better and more realistically.

2.3. Latent Space Manipulation

Generative Adversarial Networks (GANs) use latent vectors that are randomly selected during image generation. Some methods use these vectors only as an input of first layer, while others can perform more computations in the deeper layers. More computations with latent vectors offer some advantages to these methods. It can contribute to different features in deeper layers. In StyleGAN2 [1], style vectors are used as input in each layer of the generator network. Because latent vectors are a critical component of generative models, even though they are just random vectors, the researchers start to study whether manipulating them makes a difference in the output images. They show that the manipulations give the user a great deal of control over the output images. Moreover, this process requires no additional supervision or computation. Only basic arithmetic operations are applied to the latent vector and a new image is generated from the manipulated vector.

Goetschalckx et al. [32] present one of the earliest studies of latent space manipulation called GANalyze. The authors aim to find high-level attributes in the images that BigGAN [33] generates. They focus mainly on memorability as a fine-grained attribute, but also use aesthetic and emotional valence for other pattern attributes. They show that memorability is not only related to object class, but also to color, shape, or size. Therefore, finding directions to edit these attributes in the images can make them more memorable or less memorable. For example, in the study, the image

of a cheeseburger becomes more memorable when it is made rounder, larger, and more colorful. They show that the aesthetic and emotional valence of images also change when they are brighter or clearer. Their method has a transformation function

$$T_\theta(z, \alpha) = z + \alpha\theta \tag{2.1}$$

that takes latent vector with a coefficient α indicating how much the vector moves in a predefined direction. They also try to find the best assessor function A that calculates memorability score of the output image and optimizes the MSE loss.

The face specific latent space manipulation method called InterFaceGAN is introduced by Shen et al. [34]. This work aims to find the connection between latent vectors and output image semantics. For this purpose, they employ some well-known classifiers from the literature. Then, a large set of latent vectors is randomly selected and the corresponding images are generated using GANs, namely StyleGAN and PGGAN [35]. The generated images are classified according to different categories such as gender, age, smile, and so on. Since the latent vector of each image is known, the classification is also applied to latent vectors. In other words, the latent vectors are linearly separated according to their semantics. This process allows finding latent subspaces so that authors can determine the correlation between latent vectors and image semantics. Therefore, latent codes can be manipulated by the desired subspace and images can be edited by age, gender, eyeglasses, smile.

Another study called Closed Form Factorization was published by the same authors Shen et al. [36]. They claim that previous work in finding directions requires too much computation. They may need to query a large set of latent vectors and compute directions. This also requires synthesizing images and annotating them [36]. Some other methods train more models to synthesize an image with the desired features. Instead of all these workload, authors present a novel method that focuses only on pretrained layers of the generator. They show that each layer of the generator projects the input into a different visual space. This means that the variations in the images

come from the pretrained weights of the layers. The paper shows that decomposing the weights and finding eigenvectors provides the directions of the generator. Since the operation depends only on the weights of the generator, this method can be applied to any kind of GAN such as StyleGAN, BigGAN, ProGAN. Their experiments show that they can manipulate object orientation, shape, posture, zoom and many other disentangled features.

2.4. GAN Inversion

Image synthesis has reached a new level in recent years. Major advances in Generative Adversarial Networks (GANs) play the main role in this success. While they synthesize high-quality images, they have some intermediate latent codes (or style codes for StyleGAN [1]) that allow the user to manipulate images. StyleGAN, for example, maps latent vectors to style vectors using a pretrained network. The output of this network corresponds to W space. The generator synthesizes an image from this style vector. When the style vector is modified by adding another vector, the output image is manipulated in a particular direction. Therefore, the real images must be inverted into the latent vectors to manipulate them. If the inverted code is given directly to the generator, the original image is synthesized again. If it is manipulated by other directions, edited version of the original image is synthesized.

Previous studies have attempted to encode images into the W space of StyleGAN. However, the results failed because the reconstructed images were not close enough to the original images. Drawing lessons from these, researchers have introduced new studies [37–39] which work based on optimization based inversion. These methods invert images into a new space called $W+$. It contains 18 style vectors for each layer of StyleGAN. Their problem is that it takes several minutes to generate only one single image. Therefore, encoder-based inversion methods are introduced. Richardson et al. propose a method called pixel2style2pixel (pSp) [40]. They create an encoder based on Feature Pyramid Network [41]. This network extracts feature maps of a given random image at 3 different scales. Each feature map is used to obtain some

of the 18 style vectors. The extracted feature maps are given to a pretrained small map2style network [40] to find style vectors. The final style vectors are passed to the appropriate layers in the StyleGAN generator. Another novelty of this method is that the discriminator is not trained from scratch but an pretrained StyleGAN generator is used.

Although recent works have achieved successful results for inversion, encoding images into $W+$ space may not yield the best style codes for editing. The style vectors in the output style code need to be closer to W space. Tov et al. present a novel encoder [42] called Encoder for Editing (e4e). This study aims to invert images to style codes where the style vectors are closer to W space. According to the authors, style vectors become less editable when they move further away from W space. They improve the encoder of [40] and it returns only one style code w and 18 offsets indicating the style vectors of each StyleGAN layer. They try to regularize the offsets, since the variance of the offsets must be as small as possible to achieve high quality and editability.

2.5. Data Augmentation in Medical Field

Data in medical imaging is quite inadequate due to privacy and lack of labeled data. Researchers are trying to augment the existing data using some techniques, but they do not provide the desired variations in medical images. They are only changes in the angle, the size of the images or the coordinates of the pixels in the images. Thanks to the development of Generative Adversarial Networks (GANs), data augmentation becomes more realistic. This technique generates new and unseen images in the domain. In medical imaging, there are some research works on data augmentation with GANs.

In 2018, Moradi et al. published a study [43] on data augmentation in chest radiographs for classification. They used normal frontal chest radiographs and radiographs with cardiovascular disease from the dataset NIH PLCO. They modify the model of Radford et al. [8] by changing the structure and number of convolutional layers. Their

next step is to classify normal and abnormal X-ray images by using a model similar to VGG. They create 3 different experimental setups to test if the augmentation works. The first experiment contains only original images from the NIH dataset. The second experiment contains images with traditional augmentations such as transformation and cropping. The last experiment is performed with original images and the images generated by GAN.

Kora et al. [44] proposed to use a Deep Convolutional Generative Adversarial Network (DCGAN) [8] model to generate new chest X-ray images to augment limited dataset. They use the dataset [45] published by Kermany et al. in 2018. The number of normal and abnormal chest X-ray images is quite small, which leads to overfitting of deep learning models in some other studies. They train the DCGAN and generate images by giving 100×1 vectors to the system.

2020 is a difficult and unexpected year for the whole world because of the outbreak of Covid-19. This disease is unknown at the beginning of the outbreak, but the drastic increase of patients gives more information about it. Experts have discovered that computed tomography (CT) of the chest provides more accurate results than other diagnostic tests. In the field of medical imaging, research began on how to get computers to detect COVID-19 disease from chest X-rays. One such study was pulished by Waheed et al. [46]. They propose a data augmentation model of chest radiographs because there are only hundreds of public datasets to study on Covid-19. Therefore, they propose a model called CovidGAN based on Auxiliary Classifier Generative Adversarial Network (ACGAN) to augment the chest X-ray images. For this purpose, they collected public images from 3 different chest X-ray datasets [47–49] for Covid-19. After the generation of new chest X-ray images, a classification model is applied to check the quality of the images. This model is a modified version of VGG16 by adding some extra convolutional layers at the end of it. Then they compare the result of the classification experiment with and without generated chest X-ray images.

Another study published to classify chest X-ray images for Covid-19 disease is proposed by Saleh Ablahli [50] in 2020. As mentioned earlier, due to the limited data for X-ray images for Covid-19, they first try to generate synthetic images by using GAN model by [51]. After generating new images, they show them to the experts and eliminate the bad images based on their comments. Then they perform different experiments to find the best classification method for Covid-19. In the first experiment, they use a convolutional neural network they created for classification. In the second experiment, they use Inception-V3 [52]. In the last experiment, they use ResNet-152 [53] for classification.

A different type of augmentation technique is proposed by Guendel et al. [54] in 2020. Instead of generating entirely new chest X-ray images with generative models, they used a technique called local feature augmentation. They take the X-ray images having some diseases like lung cancer because they have small nodules somewhere on the lung, and extract those nodules to implant other healthy images. Their solution consists of two steps. The first step uses an inpainting method that is a modified version of [19]. A predefined bounding box is extracted from the X-ray image and the nodule at the center of the box is inpainted. Then, the difference between the original and inpainted nodule section is determined. This provides the extraction of the nodule. In the second step, this extracted nodule is implanted into other healthy X-ray images. During implantation, some traditional augmentation methods such as rotation or flip are applied to the nodule.

3. BACKGROUND

Although our main interest in this thesis is the Generative Adversarial Network (GAN), which is one of the most commonly used generative models, it is better to analyze both generative and discriminative models to provide a better ground.

3.1. Discriminative Models

Discriminative models essentially compute the conditional probability of a given vector, which may be raw data from a text, an image, or extracted features. These models output the probability distribution over the target classes y for the given input x which is represented as $P(y|x)$. Discriminative models attempt to classify a given input by using information from observed data. This information must be essentially gathered from large datasets containing data with their labels. These models have some parameters and these parameters are optimized by observing each input label pair. Then new and unseen data is passes as input and the model tries to find its correct class with the highest probability.

Although discriminative models are considered supervised learning because most methods use them, there are also some counter examples. For example, some clustering algorithms can be used for classification. These methods do not require prior information while classifying data. For given data, clusters can be created and it is said that these data belong to the same class. However, it cannot give the exact label of the data because it does not know the actual label of the data in a cluster.

Discriminative models aim to learn to create boundaries for classes in a given dataset. These models should have the best mapping functions that can discriminate classes by learning from input-output pairs in the training set. These training sets can be really huge and the models need to be designed accordingly to achieve minimum error rate. For instance, there are 1000 different classes with more than 14 million

images in ImageNet [55] dataset in the current version. Even though complex models may be required for optimal classification, they are generally simpler models than generative models. The reasons for this will be explained in the next chapter.

Humans also learn by methods of discernment from birth to death. When babies are born, they have no idea about the world. As they grow up, they experience the world and learn about everything in it. For example, they see animals and categorize them based on their specific characteristics. If an animal sounds like "meow", it should be a cat. Likewise, animals with wings should be birds. At babies' early ages, they may misclassify animals because they do not see enough data yet. They may match a zebra and a horse in their mind because they have common characteristics, and a baby who has never seen a zebra may say it is a horse. When they get to know the animals in more detail by looking at the shape of the nose, the length and color of the feathers, the length and shape of the tail, etc., they can determine the exact breed.

The best discriminative models are designed based on the human brain and are called artificial neural network. It is similar to the neural connections in our brain. There are many layers, neurons and countless connections between them. Each layer learns different things. While some of them are small but crucial details, others are big and obvious features. Neural networks take an image and divide it into small parts in the first layers. Each small part can activate a different part of the network and it can recognize the shape or size of the nose, ear and tail. Then it can combine different features and classify the whole image.

Some examples of discriminative models are logistic regression, conditional random fields, and random forests. Besides, there are some deep learning based classification models such as Res-Net [56], Inception [57], Alex-Net [58], and their variants. These models are some of the most successful baseline methods for classification. They are also adopted by many other deep learning applications. The ImageNet dataset mentioned earlier has been used in many competitions. In each competition, great base models or state of the art models are presented. Since ImageNet contains numerous

images, the models that are trained with it have hundreds of millions of parameters. EfficientNet [59] is one of the latest and best classifications tested on ImageNet. It has a top-1 accuracy of 88.4% and was the best when it was released. In early 2021, another method Meta Pseudo Labels [60] is proposed based on EfficientNet. It is the best model with a top-1 accuracy of 90.2% at the time of its publication.

Measuring discriminative models can be simple and completely objective. There are some quantitative metrics that can show how successful these models are. This makes it possible to compare different discriminative models and decide which model gives the better results, unlike generative models whose results are quite subjective and difficult to measure, as we will see in the next section. This makes it advantageous for use in industry. It can be used for various applications in industry. For example, the military or national defense of countries are quite critical areas for their power. Applications that use discriminative models can be used in this field to detect enemy units or unexpected vehicles at the borders or aircrafts. This ensures unlimited protection without constant human surveillance if they are successful enough to be applied. Another application could be healthcare systems. With the world's population increasing rapidly every year, some systems are no longer adequate to protect human health. Most of these systems require interpretation by experts, and each expert has to examine patients whose numbers are much larger than they should be. This has absolutely undesirable consequences. The systems with discriminative models help the experts to examine many more patients in less time. This can save some people's lives.

3.2. Generative Models

Unlike discriminative models, generative models attempt to learn the joint probability distribution, $p(x, c)$, of the data. They create a space by using the features of the given data and generate new ones by sampling from that space. The joint probability formulation is formally described as

$$p(x, c) = p(x) \cdot p(c|x) = p(c) \cdot p(x|c). \tag{3.1}$$

We can also see this formulation inside of the Bayes' Theorem which is

$$p(c|x) = \frac{p(x|c) \cdot p(c)}{p(x)} = \frac{p(c,x)}{p(x)}. \tag{3.2}$$

Generative models must have the following characteristics to be a successful model. The model must generate new data which must be in the sample space. In other words, the generated outputs must be the same category as the data in the training set. The second characteristic is that the outputs must be different from the observed data. The parameters of the model must be optimized to generate unseen data. To illustrate this, we can imagine a dataset consisting of tens of thousands of dogs. These may contain many different breeds, colors, angles, or positions, as they are taken in real life. Generative models aim to learn the complete data distribution of this set. While doing this, the models do not require any kind of annotation or label.

Since generative models can be considered unsupervised learning because they can learn with unlabeled data, they can also be supervised learning. These models can generate data for some specific categories. Unsupervised learning gives generative models the advantage that they usually do not require annotations.

While discriminative models have a simpler task, which is only to classify based on some features of the inputs, generative models have a much more difficult task. Discriminative models can accomplish their task by checking whether the image has certain features. For example, the shape of the nose, teeth, ears, tail, or color may be sufficient to predict whether the given image is a dog. Their exact position or location relative to each other does not matter much. However, in a generative model that is intended to produce a realistic dog image, each part must be in the correct position relative to each other. Otherwise, the image is completely different from the training set and looks uncanny when viewed. Learning a joint probability distribution requires a more complex architectural model and parameters. Another difficulty is that optimizing these parameters requires more data.

Another difference between discriminative and generative models is that the former finds the boundaries between classes in the set. For the digit space, discriminative models find the best function that separates each digit and maps the given inputs for each output. Generative models, on the other hand, find where the distribution of the data lies in the space.

In the previous section, we have already mentioned how the human brain functions as a discriminative model. Similarly, the brain is also a great generative model. As humans speak or think about something, they can generate anything with a single word. This happens so quickly that even humans cannot understand this process themselves. In fact, this can happen whether man wants to do it or not. It is a completely instantaneous process in the brain. Moreover, there are no limits to these generations because people have experienced countless things in their lives and collect unlimited data. Even if conditions change simultaneously, they can easily create new images. These images can be very diverse. This proves how complex the human brain is and how successful the generative model is.

One of the disadvantages of generative models is the difficulty of measuring their outputs. The evaluation of the generated images is a subjective matter. There is no metric to measure how successful the output of a generated image is. If a quantitative evaluation must be made, the question for these models is what the real valued results might be. Therefore, the qualitative results, which may be expert judgments, do not provide an objective comparison between different models. One result of this problem is that the number of people working in the field is smaller than for discriminative models. This prevents the rapid improvement of generative models compared to other models. Nevertheless, new methods have been introduced in recent years that show the improvement of generative models.

Generative models can be helpful when creating characters in games or animations. They can speed up the processes and save money and time. However, one of the most important applications of generative models is data augmentation. To achieve

better results in deep learning, large amounts of data may be needed. In some fields, such as medical imaging, this is sometimes not possible. Even if the models are great candidate for some purposes, they cannot be trained and give poor results due to lack of data. To overcome this problem, data augmentation has been proposed as a solution. Although some basic techniques are applied, generative models are the best option because they can provide unobserved and diverse data to expand current dataset. This also contributes to the development of discriminative models.

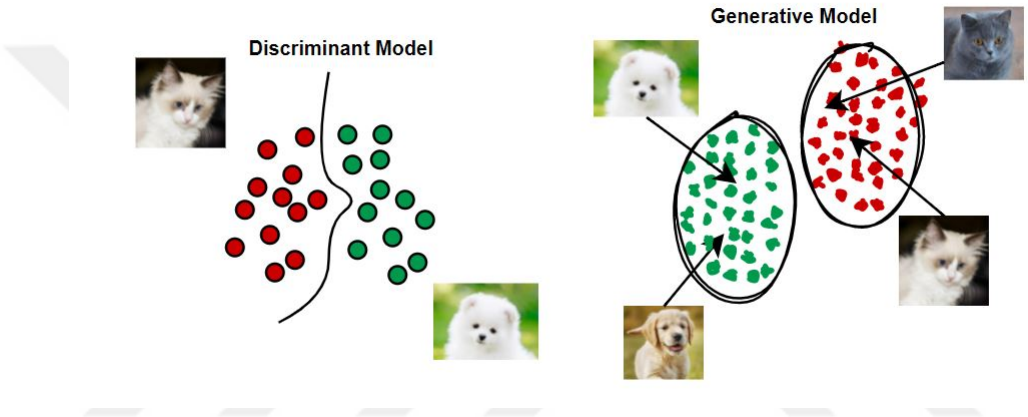


Figure 3.1. Difference Between Discriminative and Generative Models in Space.

Naive Bayes, autoregressive model, Gaussian mixture model, Hidden Markov model, latent Dirichlet allocation are some traditional models that have been used for data generation. In recent years, with the great development of deep learning, these have been replaced by Variational Autoencoders and Generative Adversarial Networks. Since methods applied in this thesis are developed based on Generative Adversarial Networks (GANs), we will give information and some details about them in the next section. Then, famous variants of GANs will be presented. Finally, the application of GANs will be described in detail with numerous examples. While these works include the earliest ones to show how they improve, they also consist of state-of-the-art methods.

3.3. Generative Adversarial Networks

This chapter will focus on what Generative Adversarial Network is, how it works and where it is used.

3.3.1. Introduction

Generative models have taken a completely different path after Goodfellow et al. published one of the most famous papers in Deep Learning: Generative Adversarial Network (GAN). This study is the beginning of a new era for generative models.

The generative adversarial network consists of two parts, the generative and the discriminative network. The task of the generative network is to generate plausible new samples from a given random vector. The discriminative network, on the other hand, tries to figure out whether a given input is generated or is a real image. Figure 3.2 shows the general architecture of GANs.

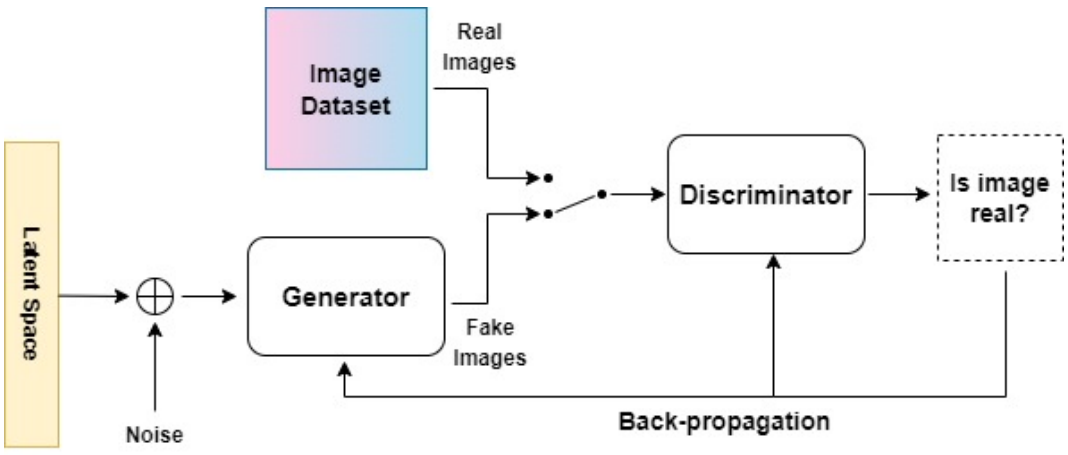


Figure 3.2. Visualization of GAN Architecture.

A generative network takes an input from a random distribution such as the Gaussian distribution and learns to generate realistic and unobserved outputs in the problem domain. The main task of this network is to fool the discriminative network. Since the discriminative network learns whether the given input is a generated image or a real image, it tries not to be fooled by the generative network and finds the fake images

with the highest probability. With this system, both networks make each other better. The generative network uses the values from the backpropagation of the discriminative network to optimize its parameters. Similarly, the discriminative network learns from the inputs, either generated or real, to produce better results. However, the goal in this system is to make the discriminative network to succeed half of the time. In other words: If it gives the correct result with a probability of 0.5, the generator works best.

Goodfellow et al. summarize this in the paper [7] as follows: "D and G play the following two-player minimax game with value function $V(G, D)$." General formula is

$$\min_G \max_D V(G, D) = E_{x\ p_{data}(x)}[logD(x)] + E_{z\ px(z)}[log(1 - D(G(z)))]. \tag{3.3}$$

Generative adversarial networks become so popular after the publication of the first model because they solve some difficult problems such as image super-resolution, image-to-image translation, and are the most commonly used method for data augmentation. Data augmentation may be the most critical issue in deep learning. Because the models require large amounts of data to learn the problem domain, sometimes the available datasets for the problem may not be large enough to produce good results. Less data can cause problems such as overfitting, which means the model can memorize the training data. This happens because the model is too complex for the limited data. To solve this problem, some basic data augmentation techniques are applied to the data, however they do not generate new and unseen data. These basic techniques are zooming, cropping, flipping, transformations, scaling, and some other physical changes to the current data. They do not provide unobserved data and may not always increase the success of the models. The modern models from GAN are quite successful in generating realistic data in the problem domain. In particular, in some areas such as medical imaging, where the amount of data is quite limited, these models provide thousands of new data for detection, classification, or other types of problems.

3.3.2. Variants of GANs

Generative adversarial networks have been used since the GAN paper by Goodfellow et al. [7] used to synthesise new images for different problems. Different problems may require different architectures or modifications to existing GANs to solve their own problems. Until the revolutionary GAN paper is published, many new studies are proposed to reflect the current state of the art. Some important studies are Deep Convolutional GAN by Radford et al. [8], Wasserstein GAN by Arjovsky et al. [9], and StyleGAN by NVIDIA Labs [1].

3.3.2.1. Deep Convolutional GAN. The concept of generative adversarial networks is revolutionary, but the models introduced are unstable. Although some studies are conducted to solve this problem, no successful solution can be found. In 2016, Rasford et al. [8] proposed a GAN model improved with convolutional layers. They removed the max-pooling layers from the GAN model of [7] and added convolutional layers instead. The name of the network comes from this improvement.

To create a more stable system, they also added batch normalization [61] before most of the layers in their GAN. Thanks to the batch normalization, the flow of the gradient in the model becomes more efficient and successful.

Even though generative adversarial networks are designed to synthesize data, they may be used for unsupervised classification. The layers of both generative and discriminative networks can be modified as feature extractors. While DCGAN is introduced with a new type of GAN, the authors prove that these networks can perform well in image classification. They train the DCGAN on the ImageNet-1k dataset and use the trained network as a feature extractor. They convert the network into a complete classifier and test it on the CIFAR-10 dataset.

Another contribution of the study is that they can visualize and manipulate the intermediate results to obtain different outputs. These results are promising for future

studies. They show that there might be many different applications for generative adversarial networks.

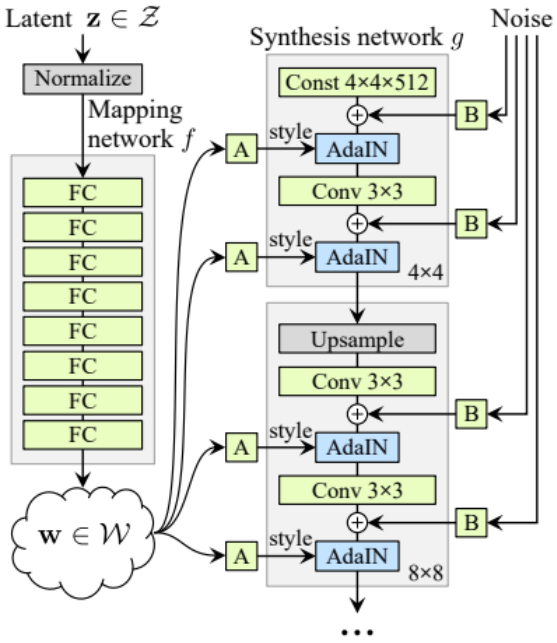


Figure 3.3. Overview of the StyleGAN Architecture taken from paper [1].

3.3.2.2. Wasserstein GAN. The discriminator of GAN is introduced to determine whether a given input is real or fake. The other proposed GANs are also used it in this way. This leads to the problem that the loss value may not decrease even if the generated images are visually better and more realistic. Since the system does not focus on whether the quality of the image is getting better, it only makes a binary classification. In 2016, Arjovsky et al. propose a new model called Wasserstein GAN [9]. They calculate a score which shows how much the image is fake or real as a floating value.

They introduce a new distance metric called Earth Mover (also referred as Wasserstein distance). It calculates the cost of transferring from one probability distribution to another. The authors' goal is to find the minimum value of this cost. This distance method is used as a loss function in this model because it allows the calculation of how much real or fake the image in terms of a floating number. Using this score instead of a binary classification between 2 classes provides more detailed and better feedback

to the generator. As a result, generator optimizes itself better and generates more realistic results.

Previous generative adversarial networks use Adam optimizers based on momentum, which makes the model unstable in the training phase. In this paper, the authors use the RMSProp optimizer instead and their model works more stable.

3.3.2.3. StyleGAN. Each new type of generative adversarial network focuses on improving the discriminative network. Researchers do not change anything in the generative network and consider it as a closed book. They assume that improving the discriminative network will ensure that more realistic images are generated. Researchers from NVIDIA Labs propose a new method in 2019 called Style-Based Generator Architecture for Generative Adversarial Networks [1]. In this paper, Karras et al. make no changes to the discriminative network of previous GANs, but make major improvements to the generator.

This model differs from the previous ones in its input. While other networks take a vector from a random distribution, the input of this network is always constant and its size is $4 \times 4 \times 512$. When the input reaches deeper layers, it is upsampled and its size is doubled in width and height in each layer. In other words, the size becomes $8 \times 8 \times 512$ at the output of the first convolutional layer, then $16 \times 16 \times 512$ in the second and so on until the last layer, which produces an output of size $1024 \times 1024 \times 512$. Each layer affects the style in a different part of the image. According to the paper [1], each resolution affects the style in different level. For example, the first layers of the network control the pose and hairstyle. The layers at the end of the network control the skin or eye color and fine details. After each process in the convolutional layers, a different noise is added to the results as bias. Their experiments show that adding noise to different layers results in significant changes in style. For example, adding noise to all layers or reducing the number of layers changes the curly in the hair or synthesizes a less realistic (as in animation) background or skin. After noise is added, each image is normalized using an operation called adaptive instance normalization (AdaIN). This

provides stable and standardized intermediate steps in the network. The style of each detail can also be strongly controlled with these new additions to GAN.



Figure 3.4. Example results of generated faces taken from paper [1].

Another contribution of this study is that it proposes a new element, called the mapping network, which replaces the input from the latent space with a new one generated using a fully connected neural network. This network generates a new intermediate space from the latent space, which is used to obtain a sampled vector. These vectors from the mapping network are added to the images in the AdaIN step. Two different vectors, namely w_1 and w_2, created using two different mapping networks are included in the network. These vectors are called style vectors because they transfer the style of two different images to synthesize and create a new image. These two styles reflect the different sides of the images. For example, while one vector transfers the hairstyle of the first image, the other vector transfers the hair color of the second image. In the Figure 3.4 from the paper [1], the result of this transfer can be clearly seen.

3.3.3. Applications of GANs

The great advances in generative adversarial networks make them very popular in the field of deep learning. While some researchers try to find better way to obtain more realistic and high-resolution images by improving GANs, others try to apply existing GANs to new domains that also may require some modifications or improvements. This section contains the most popular application areas of GANs.

3.3.3.1. Conditional Image Generation. The original paper of GAN proposes a model where the discriminator network determines whether the input is real or fake. It has no idea what the label, class, or category of the image is. Later works start to focus conditional generation of images according to their labels. The difference between the standard GAN and conditional GAN can be found in the Figure 3.5. In classical GAN [7], the system just takes a random noise as input and generates an image in the given domain. In conditional GANs, a new input is introduced, namely a class that specifies to which category the synthesized image belongs. While the class is given as input to the generator, it may also be given to the discriminator. In another method, the discriminator can calculate the probability to which class the image belongs.

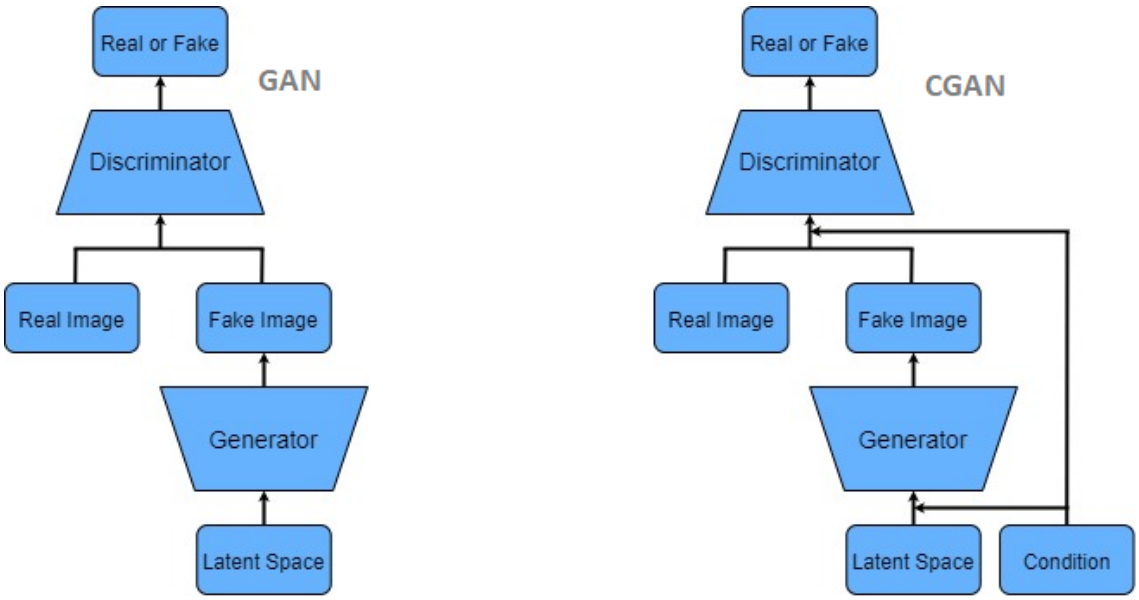


Figure 3.5. Difference between GAN vs Conditional GAN.

The pioneering work for this task was done by Mirza et al. [62]. In this study, the authors added label variables for both the generative and discriminative networks. While the random vector z is given to generator, a label which contains the information about the category of the generated image is also given as input. On the other hand, the discriminator not only determines whether the given input is real or fake, it also finds the class based on the trained data. This new type of discriminator provides more information to the generator so that it can learn better than the original GAN model. The authors conduct experiments on the MSNIST dataset and show that it can perform better than some existing models while outperforming some unconditional GAN models. Nevertheless, they note that although it is not the best model for the task, it is a proof of concept for conditional GANs.

As previous methods have suggested, generative adversarial models can generate images from a class with conditioning. However, the effects of the large number of classes were unknown until Odena et al. published a new method called Auxiliary Classifier GAN (ACGAN) [63]. This method uses all 1000 classes of the ImageNet dataset [64]. They show that conditioning network with a large number of classes is also possible. The discriminator network of the method gives two outputs. One of

them shows whether the given input is real or fake and the probability distribution of the input for the classes. Moreover, this method produces images with a size of 128×128. Another contribution of this work is that generating a higher resolution allows higher accuracy in distinguishing the classes.

Earlier methods in conditional generative adversarial networks suffered from controllable diversity in the specified domain. It is also important to synthesize a realistic image by keeping it in the domain. Bodla et al. [65] introduce a conditional GAN called FusedGAN that solves all these problems. Their model has two generators, including a conditional GAN and an unconditional GAN. The unconditional GAN is responsible for sketching images without any conditions. It does not know or care about conditions. For example, this generator may generate a dog image, but it does not paint the image because it depends on the conditions. Then the conditional generator takes the sketch as input and completes it according to the given conditions. This generator now paints the image of the dog for the given example in black and white or with some shapes on its body.

Generative adversarial networks can be applied for specific purposes that might take too much time for humans. LoGAN [66] was introduced by Mino et al. for generating logos. Since designing a logo can be a lengthy process and requires too much effort, the authors try to use GANs to make this process short and simple. They improve AC-GAN with the loss function of WGAN-GP, to provide stability during training. They use the twelve different colors while conditioning GAN. The architecture of GAN consists of three parts, a generator, a discriminator, and a classifier, which is used to classify the logos as in AC-GAN.

In addition to conditioning the generative adversarial network with labels, the use of textual content is another method. Instead of just using the name of the class or category, some networks can generate an image based on the description given as text. Stap et al. [67] present a method that first generates an image from a textual description and then manipulates the result to look like a person's thoughts using the same textual

data. For the generation step, they modified StyleGAN [1] and added a conditioning mechanism and called it textStyleGAN. They create kind of preprocess network that takes the textual description and the image and creates joint embedding space. They feed the StyleGAN network from this space and synthesize the images. They also use an attention mechanism for words to determine critical features in descriptions. After generating images, they use a second network to manipulate the results. The manipulation is based on features such as age, smile, and gender.

3.3.3.2. Image Editing. The development of Generative Adversarial Networks creates new application areas. After conditional GANs generate plausible results, more studies are published that improve them and use them for specific purposes such as image editing. As the name suggests, studies for this purpose aim to synthesize new outputs based on an image by modifying features of the image.

Peranau et al. [68] propose a model called IcGAN based on the conditional GAN model of [62]. In cGAN, a conditioning label is given to GAN with the latent space variable z. This study uses a different approach while giving these inputs to GAN. They use two encoder frameworks to create the inputs from GAN. One of them creates the latent variable z by encoding a given input image. The other encoder extracts features of the same image such as hair color, hairstyle, and gender and creates a one hot encoder vector named y. Then the features extracted by the encoder are edited as desired. These two latent space and conditioning vectors are passed to cGAN and the system generates a new image accordingly. For a given image with a man with black hair and eyeglasses, it can be edited as a blonde woman without eyeglasses by changing the encoded feature representation of vector y.

Generating high-resolution images with GAN is becoming a problem. Previous methods cannot create realistic and visually plausible images at high resolution. Want et al. propose a new GAN [69] for generating high-resolution images while editing images with conditional GAN. Their input to the model is a semantic label map of images. Therefore, they use the map and image tuples in training. They use two

generators in their model. The first is called the global generator and is employed to synthesize an image with the desired changes. This generator takes the label map with a size of 1024×512 and generates the output according to the manipulation. Then the second network, the Local Enhancer, takes this output and increases the resolution by 4 times. The result is an image with a resolution of 2048×1024. They also choose 3 discriminators to distinguish images with 3 different scales. They downscale image by a factor of 2 and 4. While the discriminator trained with the lowest resolution gives feedback to the generator about global details, the one trained with the original output focuses on fine details.

Face aging is more specific subcategory of image editing and can also be overcome with GANs. Antipov et al. claim that previous face aging methods which are based on prototyping and modeling have some problems needed to overcome. Prototyping based methods apply same kind of changes to each image and all of them looks like each other eventually. Modeling based methods require many images from different ages for a model (or person) to learn specific changes. However to get that many images for every person is quite challenging task itself. Authors propose a new method [70] to overcome these problems. They aim to preserve identity with their model. Their approach includes two steps. In the first one, they try to find the best vector which can identify the features of the person in the input image and will be used to preserve identity. They employ an encoder to find the initial attribute vector. Then, this vector is given to FaceNet model [71] as input for face recognition. This network makes the optimization for the best identity preserving attributes. In the second step, they introduce a new GAN model called Age-cGAN which can takes the optimized attribute vector and the age conditions to generate most realistic images while preserving identity.

Karras et al. present a new model [35] that focuses on the growth of both the generator and the discriminator network in GAN. Their contribution to this study is that they start from a low resolution image and enhance it in the deeper layers. Therefore, they generate images with large resolutions such as 1024×1024, unlike the previous models. This system also solves the problem of instability and slowness of the

previous GANs. A new metric that improves the Wasserstein distance is also presented in this work. Finally, they contribute CELEB-HQ dataset, which is a higher quality version of the CELEBA dataset.

In image editing methods, models generally create a vector of image attributes and synthesis operation is done after these attributes are changed. These vectors contain binary attributes and when one of them is changed from 0 to 1 or vice versa, the image changes. Lin et al. consider this approach to be quite limited and inefficient for image editing. Therefore, they introduced a new method RelGAN [72] that uses relative attributes rather than binary ones. For attribute editing, they use real-valued attributes so that the difference between modified and original values gives the relative attributes. They also claim that interpolation of edited attributes is not successful in the previous methods. Hence, they add a new discriminator to the system that is responsible for controlling the interpolation success of the generated image. The generator in the model takes the relative attributes of the original image and creates a new one and passes it to 3 discriminators. One discriminator checks the realness of the image, one evaluates whether the output matches the input and relative attributes. The last discriminator, as mentioned before, evaluates the interpolation of the output.

Shen et al. published a paper on editing faces with GANs [73]. They propose a model called Interpreting Face GAN (InterFaceGAN). We have provided the details in the Latent Space Manipulation section of Related Works chapter. They synthesize an image from some random latent vectors. Then they classify images using predefined SOTA classifiers. While separating the images linearly, they also create latent subspaces. These subspaces allow the correlation of image semantics and latent vectors. Finally, they can use these correlations for image editing.

3.3.3.3. Image to Image Synthesis. Generative adversarial networks are applied various domains, as mentioned in the previous two sections. With the development of GANs, new application areas are proposed. One of these areas is image-to-image synthesis, which is a kind of style transfer. This work aims models to learn to combine

visual representation of one image with the content of the other image. In other words, mapping features of a source image to a target image while preserving its content or predefined features.

Image-to-image translation is an application area for which other methods were applied before GANs. However, in training, these methods require pairs of images to learn the relationships between two domains and map them. Since this is a difficult problem in any domain and image, Zhu et al. introduce a GAN method called Cycle-GAN [74]. They introduce a model that uses two model functions for images X and Y. While one mapping function generates image Y from image X, the other function does the opposite. If the model succeeds in converting Y to X, which is its original version, it can be used as an image-to-image translator with minor modifications. With this idea, they create two generative models that map images between two domains. However, when converting images from Y to X, they use some specific features of the image of Y. As a result, the output image becomes a synthesis of X and Y. For the discriminator part, they again use two discriminators, one for each generator. One of them discriminates images from X to Y and the other vice versa.

Another method for image-to-image synthesis is introduced by Huang et al. It is called Multimodal Unsupervised Image-to-Image Translation (MUNIT) [75]. To overcome the problem of required image pairs, they use a system that includes two encoders and GANs. They assume that images from different domains can be encoded in a common latent space for their content. However, their styles come from different domains. For translation purposes, they first encode an image with the encoder of its domain and create a content variable. Then the style vector is taken from the other domain. The GAN model, trained in the second domain, takes the content vector and the style vector as constraints and creates a new image. This image contains the content property of the first image, while the style of the image comes from the other domain.

Generative adversarial networks are commonly used in the medical field. MRI images suffer from insufficient contrast due to cost and time issues. Contrast can play an important role in MRI images in detecting certain medical problems. With the recent development of deep learning and GANs, many methods are proposed to solve this problem. Dar et al. present a model [76] to improve contrast in MRI as an image-to-image synthesis application. They apply two different networks and compare them to find the better approach. The first model uses a classical generator and discriminator network with VGG16 as the feature extractor. The source image is passed to the generator to learn the contrast features and the new image is synthesized accordingly. Then, the discriminator distinguishes between the real image and the generated image, while the VGG16 extracts the features of the generated image and the original image and calculates the perceptual loss for the feedback. The second model uses two generators and two discriminators. The two generators are trained with different sets of images, namely images with high and low contrasts. One of the generators tries to synthesize images with higher contrast, while the other does not. Then one discriminator distinguishes original high contrast images from generated images and the other discriminates original low contrast images from generated images. Then the generators use cycle loss to improve their results.

One of the earliest studies using GANs to generate biological images is introduced by Osokin et al. [77]. The authors attempt to generate cells that originally have two different types. Some of them have red signals while the others have green signals. The cells with red signals have only one type of red signal, indicating that the cell is actively growing in that area. Green signals, on the other hand, may indicate that the cell has at least one of 41 different proteins. These provide information about the geometry of the cells. Generating new images provides information about green signals because current technology does not allow us to observe how many different proteins the green signals contain. Observing the process of generating the GAN layers and the results could provide the opportunity to find out what is happening in the green signals and learn details about different types of proteins in these cells. In addition, cells that generate red signals could provide a better understanding of growth and

division phases. For these purposes, they propose a GAN based on DCGAN [8]. Their model consists of two different GANs. While one of them learns to generate cells with red signal, the other learns the connection between green and red signals. The latter focuses on the green channel, but deeper layers link the intermediate results of the earlier GAN so that it can generate images containing both channels. They modify the original DCGAN by adding a Wasserstein objective function to it.

Previous methods for image-to-image synthesis cannot generate images by taking sample images as input. They also cannot generate high-resolution images, as shown in the study by Xiao et al. [78]. They propose a new method based on GAN to solve these problems. While previous GANs use a sampled latent space code as input, this method takes two images as input. One image has the attribute to be transferred to the second image, which does not have that attribute. They also provide a method to transfer multiple attributes at the same time. An encoder framework is used for this purpose. It creates disentangled attribute vectors and these vectors are modified as desired to transfer them to the target image.

In generative adversarial networks, the discriminator is used only to determine whether the input image is real or fake. Some studies also try to find a score of realness or fakeness of images. Another approach is to use a discriminator to find the exact difference between real and fake images. For this purpose, Emami et al. propose a method called SPA-GAN [79] that uses an attention mechanism in the discriminator. This mechanism gives feedback to the generator on how to distinguish the real and fake images. More specifically, it indicates the feature locations in the image in determining process. They also use a new loss function called feature map in order to preserve the special features of the original image.

3.3.3.4. Image Inpainting. Inpainting is the process of filling missing, removed or damaged parts of an image. With this application one can restore old images that are somehow damaged. Similarly, the broken parts found during archeological excavations can be completed in the digital environment with the help of inpainting. This can be

done by human experts, but successful models can do it in less time. From a deep learning perspective, models can provide results that match the original because they can train many samples that the expert has not yet seen. In addition, removing objects is another difficult task that takes too much time to do manually. The task of image inpainting is also used for this problem. Researchers in this field assume that with this technique it is possible to remove objects from the image and fill the hole with suitable background or neighbor pixels.

The improvement of the technique over the years and the successful results are prompting researchers to apply image inpainting in certain fields such as medical imaging. In particular, removing unwanted objects from medical images can be great benefit to experts who examine and interpret them. In recent years, many studies have been published to solve this problem using image inpainting. Image inpainting cannot only provide complete and realistic medical images for interpretation, but also augments the available data for future studies. Since the publicly available data for medical imaging is quite limited, data augmentation becomes a very important task.

Following the rapid improvements in generative adversarial networks, this technique is now being applied to image inpainting area. Previous neural network models applied for image inapinting can achieve good results, but they also have some drawbacks. For example, while they complete the missing region in a face, inpainted region looks like the one in the training set. The model learns from the observed data and completes the area that is similar to them. To solve this problem, Dolhansky et al. [80] proposed to use GANs for this task. They introduced a GAN called Exemplar GANs (ExGANs) specifically designed for inpainting eyes. This model uses the conditional GAN approach and extends the system to include the specific features of the face. These features ensure that the identity of the original face is preserved by contributing to the model in different layers of the model.

In the early years of generative adversarial networks and other deep learning methods, models were trained and images were inpainted with known masks or oc-

clusions. Therefore, the results for some random images with occlusions are not very successful. Chen et al. [81] introduce a method for inpainting random occlusions in images at GAN. For this purpose, they use a detection algorithm to mask the occluded area. After masking, the generative model inpaints the area. Their generative model first encodes the given input and finds the closest vector in the domain. Then it generates the inpainted image using that vector, and a discriminator evaluates the image.

Autonomous cars have improved greatly in recent years. They have a multiview system with multiple cameras and different angles of view for better and safer driving. When recording video images, some images may be partially or completely lost. For partial loss of images, Yuan et al. proposed an inpainting method [82] based on conditional GANs. This method uses the same scene from different perspectives. For a missing region in an image, they use left and right views of the same image to recover the region. They use an encoding network in the generator and create latent vectors for the image with the missing region, right and left view images. They obtain the spatial transformation using these three images and get a final latent vector that contains the information about the missing region. The next steps are similar to classical GANs. The generator takes this vector and creates an image that is inpainted. The discriminator distinguishes between the real and the fake images.

Chen et al. propose a GAN model [83] to complete the missing regions in images. They use a generative network and two discriminative networks, a global discriminator and a local discriminator. The local discriminator works like the classical discriminators in GANs and distinguishes whether the generated (or inpainted) image is real or fake. The global discriminator is used to measure the quality of the generated image. It checks the integrity of the image and the coherence of the texture. The generator network in this model is different from previous GANs. It takes the broken image as input instead of random noise. The authors show that this method provides higher quality and better inpainting compared to previous classical GAN methods.

After the outbreak of Covid-19 in 2020, we encounter a new disease. Although it is completely mysterious at first, rapid spread of the disease is needed to find rapid diagnostic methods for it. Various types of diagnostic kits have been developed, but none of them provide as accurate results as computed tomography (CT) of the chest. As the whole world is affected, it has become very difficult to perform quick tests and get results. Therefore, models have begun to be developed to detect Covid-19 disease in the chest by checking CT. However, data are very limited due to privacy concerns. Waheed et al. [46] present a GAN based data augmentation method to increase the amount of data for disease detection. The model called CovidGAN is based on the Auxiliary Classifier GAN (ACGAN). To check the quality of the generated data, they use a classifier by modifying VGG16.

The studies focus on first to augment the existing data and then classifying it to detect the disease. Another study was published by Saleh Albahli et al. [50]. The main purpose of the study is to classify Covid-19 disease using CT images. However, due to the limited amount of data, the authors are forced to generate synthetic data to train models and achieve better classification results. They use the model of Bao et al. [51] to generate new CT images. Their classification experiments consist of three different convolutional networks. First, they create their own CNN model. Then they use Inception-V3 [52] and ResNet-152 [53] for the last experiment.

3.3.4. Evaluation Metrics

Evaluating generative adversarial networks is more difficult than evaluating other types of networks, and there is no metric on which there is consensus. The first thing we think of to evaluate the result is people's observations. They can rate or comment on the results, but this method has some problems. Objectivity cannot be guaranteed and the ratings may be biased for some reasons. The people who evaluate the results must be experts in the field of the problem. Since many images are generated for each experiment, manual evaluation may take hours or days, which is not practical. Nevertheless, some evaluation metrics have been introduced. Salimans et al. introduced the

Inception Score for evaluating generative adversarial networks [84]. Fréchet Inception Distance was introduced by Heusal et al. [85]. Another score is the nearest neighbor accuracy proposed in the work of Paz et al. [86].

3.3.4.1. Inception Score. In 2016, Salimans et al. proposed a metric that can be used in place of human evaluation for generated images by GANs. They named it Inception Score because this metric is based on a successful classifier Inception-V3 [52]. They give a large number of generated images to the classifier and calculate the score to measure how successful the GAN model is. They consider two important rules in the calculation.

- The diversity in generated images. For a car space, each images must be different type of car.
- Quality of the generated images, e.g, images must look like their real life versions.

They assume that both rules are valid in the generated images to get a high score. The calculation logic for the score is as follows. The images are given to the classifier and the result is a probability distribution of the labels in the set. This can be expressed mathematically as $p(y|x)$. According to the paper [84], if the images contain meaningful and unique objects, they have low entropy. Of course, if the opposite is true, the image has high entropy. The authors believe that the sum of distributions $\int p(y|x = G(z))dz$ must have high entropy if the images produced contain diversity as in Rule 1. They call this marginal distribution. Then they calculate the Kullback-Leibler divergence using the label and the marginal distribution and take the exponential value of the result. The formula is expressed as

$$\exp\left(E_x KL(p(y|x)||p(y))\right). \tag{3.4}$$

3.3.4.2. Fréchet Inception Distance. The Inception Score (IS) considers only the images generated when measuring the success of the network. This method may not be

right, because the real samples are also important to understand in what domain the network is working. Therefore, an improved version of the Inception Score, the Fréchet Inception Distance (FID), is proposed by [85]. FID computes the distance between the real and the fake (generated) samples, where a smaller distance represents better generated images. Similar to IS, this metric also uses Inception-V3 for computation, but instead of obtaining the probability distribution of the labels, it uses the feature extraction layer of the model. This layer is then used to estimate the mean and covariance values for both real and generated images. The general formula for FID is

$$FID = ||\mu_t - \mu_f||_2^2 + Tr(\Sigma_t + \Sigma_f - 2(\Sigma_t\Sigma_f)^{\frac{1}{2}}). \tag{3.5}$$

3.3.4.3. Nearest Neighbor Accuracy. The Two-Sample Testing classifier is used by Paz et al. [86] proposed. This technique aims to determine whether two different samples are from the same distribution or not [87]. It is basically a classification of samples from two different classes, therefore any type of model can be applied. In this case, the classification is applied to distinguish whether the images are real or fake (generated). This task provides us information about how close these two distributions are. In [88], a 1-Nearest Neighbor (1-NN) Leave-One-Out (LOO) method is proposed for the aforementioned technique. The reason for choosing this technique is that it is an unsupervised technique that does not require training. The application of this technique can be summarized as follows. For a selected sample, the nearest neighbor for each pixel is found using Euclidean distance. Then, the distribution of the selected sample is determined based on the label of this neighbor. After classifying many other samples in this way, the overall accuracy is about 50%, which means that these distributions are very close. The authors assume that the real samples are positive and the fake samples are negative. If the accuracy closes to 0%, it shows that GAN is overfitting and generating the samples in the real data. If the generated data is completely different from the real data, the accuracy closes to 100%.

4. METHODOLOGY

In this section we will describe our methods in detail. We choose to use different types of deep learning techniques for our purpose and compare them quantitatively. Our first method is based on Deep Image Inpainting. We use a state-of-the-art technique [11] and create a heatmap-based inpainting by creating a pipeline using Probabilistic Class Activation Mapping [10]. Another method is to generate images using Conditional GANSpace. We will refer this method cGANSpace. GANSpace [12] is one of the best latent space manipulation techniques that uses the StyleGAN2 [1] generator. We extend this method by using StyleGAN2-ADA [13] and conditionalize GANSpace. This method provides more controllable conditional generation of X-ray images. Finally, we also demonstrate manipulation of real X-ray images using GAN inversion and latent space manipulation. This method uses Restyle [14] study which encodes images and returns the latent vector of the input image. Then GANSpace is used again to generate the desired X-ray images with various manipulations. The details of all methods can be found in the following sections.

4.1. Deep Image Inpainting

4.1.1. Recurrent Feature Reasoning for Image Inpainting

Recurrent Feature Reasoning (RFR-Net) model was introduced by Li et al. [11]. It aims to complete large missing regions in images. They propose a method in which the image is recurrently inpainted. In each recurrence, they fill the pixels at the border and reduce the size of the missing region. They apply these processes in three steps. These are area identification, feature reasoning and adaptive feature merging. The area identification step uses Partial Convolutions [89] method to find features for missing pixels near the boundary and then updates the mask according to the results. This updated feature map and mask are passed to the second step, feature reasoning. The goal is to fill missing pixels with the best possible value to make the patch unrecogniz-

able and more realistic. This step consists of an encoder-decoder framework with an attention mechanism, namely the Knowledge Consistent Attention. This mechanism is different from the existing methods because this system is recurrent and each patch is dependent. All other intermediate images (some of them inpainted partially) contribute to the current attention in this system. After area identification and feature reasoning create partially inpainted images. In the last step, adaptive feature merging, the final image is generated by combining all intermediate images. In computing the final image, the pixels of each intermediate image are considered. For each image, the average of all valid pixels is calculated. This process is done for all the pixels and the final inpainted image is displayed as the result.

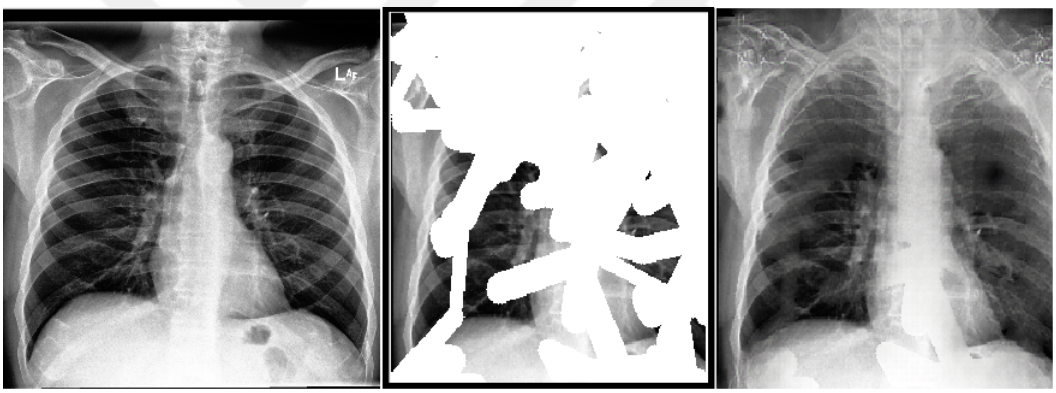


Figure 4.1. Inpainting of Chest-1 Without Artifact.

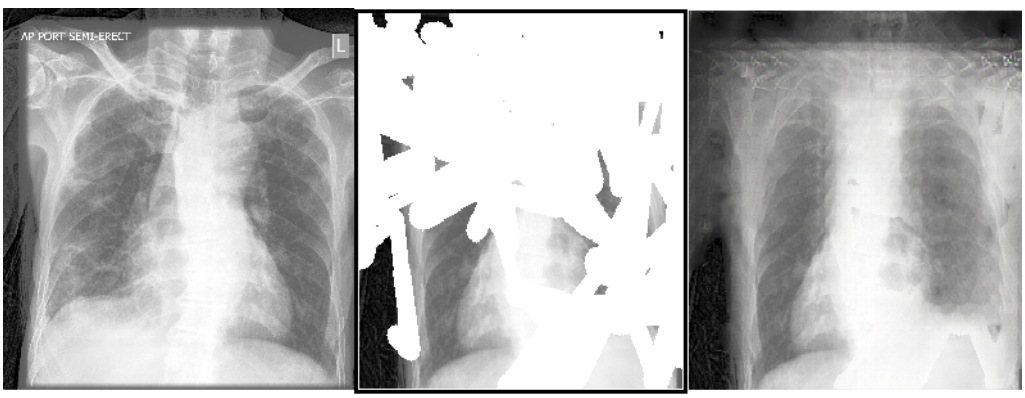


Figure 4.2. Inpainting of Chest-2 Without Artifact.



Figure 4.3. Inpainting of Chest-3 Without Artifact.

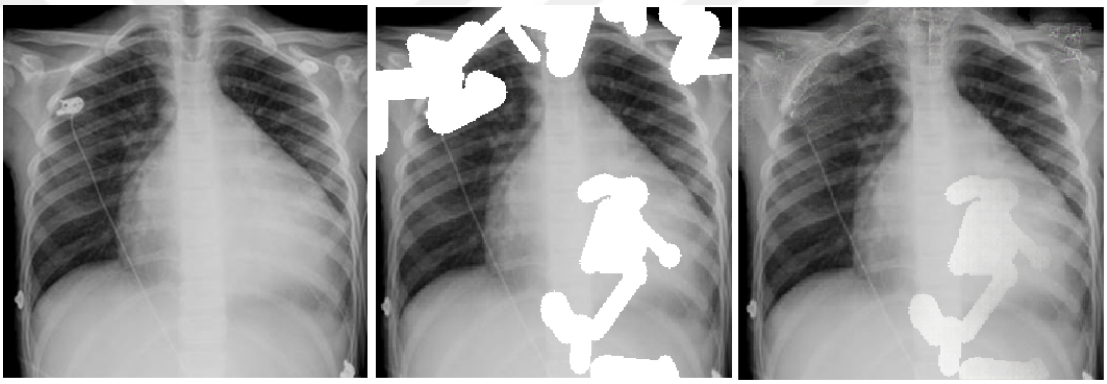


Figure 4.4. Inpainting of Chest-4 With Artifact.

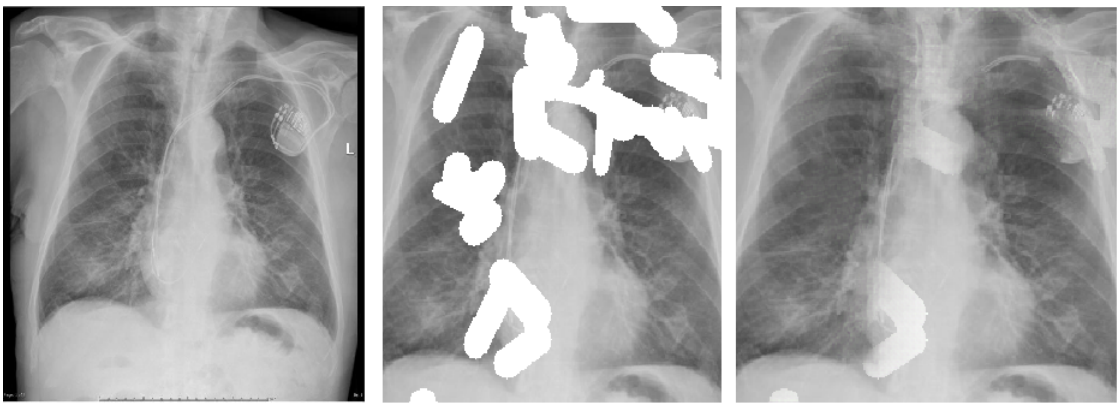


Figure 4.5. Inpainting of Chest-5 With Artifact.

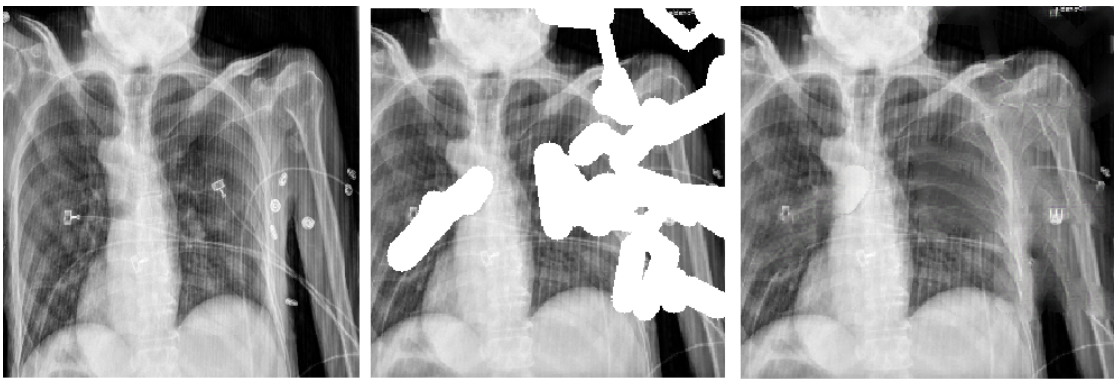


Figure 4.6. Inpainting of Chest-6 With Artifact.

4.1.2. Heatmap Based Inpainting

The last layer in classifiers is used to categorize images by calculating the probability based on the extracted features. Zhou et al. [90] have recognized that these feature maps in CNNs provide a clue to the placement of objects in the images. They show that using global average pooling instead of a final classifier layer (such as Softmax) can activate the position of objects that belong to a particular category of images. This happens because discriminative models aim to find the class of objects in a given image. Finding the class of an object requires finding the object pixels in the image, even if this process is not intentional. Finding an object means extracting the features of the object from a group of pixels. If these features are manipulated in some way, they can be used for mapping in the images, according to Zhou et al. [90].

They propose a new method for creating class activation maps (CAMs) of images. They use global average pooling after the last convolutional layer and before the softmax layer, as mentioned earlier. They use popular classification networks such as AlexNet, GoogLeNet because they do their job well and can find the features of objects in images with great success. Their final convolutional layers show the activation of units in the specific pixels of the image. Since each unit in this layer has information about specific shapes of objects, it can indicate the importance of pixels for specific objects. Moreover, the optimized weights show how important the units are for these specific objects. Hence, the authors claim that the sum of each unit multiplied by the corresponding weight for certain objects can give the activation of these pixels for the

object. Their results also confirm that this system works correctly.

In their work, Ye et al. [10] use the CAM method mentioned above as a basis for finding the localization of certain observations in chest X-ray images. They first use a convolutional neural network to extract the features of a given chest X-ray image. This yields the feature map of the image. Then, feature embeddings of fixed length are determined. These embeddings are computed for each pixel of the image. Multiplying these embeddings by the weights of classifier layer gives the likelihood of the observations. Thus, these embeddings indicate the importance of the pixels for each observation.

They also apply sigmoid to each likelyhood result to constrain it. These values give the probability of the observations for each embedding. The name of the method Probabilistic CAM comes from here. Then they use Multiple Instance Learning [91] to compute attention weights of embeddings for specific observations. These weights are used in computing the global average pooling. These weighted embeddings yield the final class activation values for each pixel. A threshold is applied to find areas that indicate the location of the observation. Example results can be found in Figure 4.7.

Finding a heatmap to locate observations allows us to avoid inpainting these areas. This is important because inpainting these areas can change the observational situation of X-ray images. While one X-ray image may obtain one or more observations, it may not contain any observation after operation. This may result in the need to change label of that X-ray image, which we cannot detect. Therefore, we must avoid to inpaint these areas.

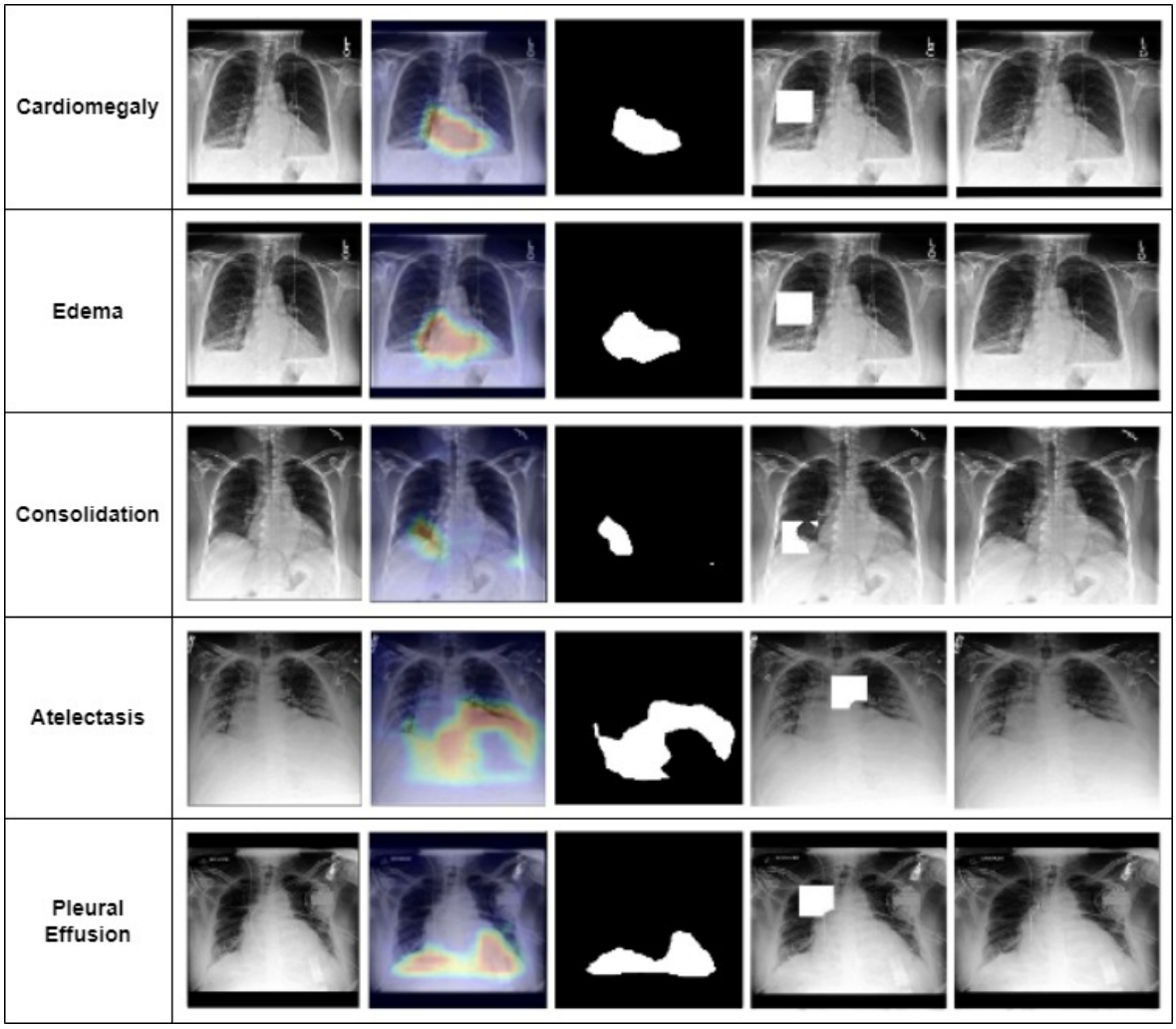


Figure 4.7. Each row shows results for different observations and X-ray images while each column shows different steps of our method. First Column: Original X-ray Image, Second Column: Heatmap is created for 5 different observations, Third Column: Regions with observations are masked, Forth Column: Masked region is chosen randomly to inpaint, Fifth Column: Inpainted X-ray Image.

We use the PCAM heatmap model for our method and apply it before the inpainting step. Although it provides the results as a colored heatmap of X-ray images, we use only probabilistic map results to create a kind of masks containing only 0s and 1s. After computing the probabilities of each pixel, we store the results where 1s indicate that these areas have observation with high probability, while 0s indicate the opposite. We calculate probabilistic map values for 5 different observations, namely cardiomegaly, edema, consolidation, atelectasis, and pleural effusion. Then, we add

the results of the individual pixels for each observation. This gives us an overall mask for the area containing at least one of the observations. We then use this when creating random masks in the inpainting process to prevent masking an area containing one observation for the reasons mentioned above. The example masks for probabilistic CAM results can be found in Figure 4.7

The steps above are used to determine the random mask without changing the label of the current X-ray image. Once the random mask is selected, we check the overlap with the heatmap mask. If the overlap is more than 20%, we select another random mask until we find the one that has less than 20% overlap. The area that overlaps is removed from the image mask and the remaining area becomes the final mask. After that, we add the mask to the X-ray image and give it to RFRNET [11]. Our module has been trained with the CheXpert dataset and is suitable for inpainting chest X-ray images. Inpainting radiographs with a mask large enough can provide the ability to obtain unobserved radiographs with predefined labels. This allows us to augment original CheXpert dataset with unlimited size. To this end, we write a small application that uses this pipeline end to end. For a given image, the application finds its heatmap, creates a random mask according to the mentioned rules, and inpaint the masked image. We use this application in our experiments. For more details on the inpainted images, see the Experiments chapter.

4.2. Conditional GANSpace

In this section, we will discuss another method for data augmentation on chest X-ray images. Our goal is to control the features of the X-ray image while applying conditional generation. To this end, we extend the GANSpace method by integrating the conditional generator StyleGAN2-ADA.

4.2.1. Stylegan2-ADA

Generative Adversarial Networks (GAN) require very large amount of data to generate realistic and high quality images. The reason for this is that the discriminator of GANs overfits when a small dataset is used for training. This results in poorer gradient values passing from the discriminator to the generator. Therefore, the generator cannot improve itself to generate better images. To learn better, hundreds of thousands of images are used in training GANs. These types of datasets must also have a large variety of images. Although some datasets with millions of images have been introduced in recent years, the number of these datasets is limited. Moreover, there are limitations in some areas that do not allow collecting large datasets. As mentioned at the beginning of this thesis, the medical domain is at the top of these domains.

Some data augmentation strategies are used to overcome the problem of small datasets leading to overfitting in many studies. These can be rotation or adding noise to images in classifiers [13]. While these augmentation techniques improve classification models, they can be harmful to generative adversarial networks. This is because GANs can learn from noisy data and generate some images accordingly, even if there is no such image in the dataset [13]. This is referred to as "leaking" in the work of Karas et al. because it leads to undesirable results due to unobserved data in the training set.

Karas et al. [13] present the paper Training Generative Adversarial Networks with Limited Data (also called Stylegan2-ADA) to apply appropriate augmentation techniques with probability that prevents discriminator overfitting. They show that this method ensures that there is no leakage to the output images from augmentations. This allows each domain to use GANs in generating new data, even when only a small amount of training data is available. In their study, they use different sized subsets of large datasets such as FFHQ or LSUN CAT, to find out how the size of the subsets affects the overfitting. They also categorize 18 different augmentation methods such as X-flips, rotations, color transforms, and additive noise and apply them to these subsets. Since there are many augmentation options, the authors conduct

many experiments to measure their effect and contribution to training. They proceed from different perspectives. One of them is whether applying different augmentations together improves the overfitting time. The experiments are conducted on different datasets. They show that augmentations do not always improve the results. Some of them work better on small datasets and others are helpful on larger ones. Another approach is that applying some augmentations with a certain probability improves the generator. For example, random rotation may confuse the generator during training and it may produce images with nonsensical orientation. Applying rotation with a probability makes the generator to see unrotated images. This makes the generator more robust and less likely to generate images with incorrect orientation. In their method, the discriminator is trained using only augmented images and this is one of their contributions with this study. Another conclusion from the experiment is that lower values of probability p give better results while the amount of data increases. Therefore, they conclude that the optimization of the augmentation depends entirely on the size of the training dataset and the optimal value of p, which is about 0.5.

This paper also introduces the conditional StyleGAN2. In the original StyleGAN2 [1], the output image is generated from a random latent vector without any interference. The only known information about the output image is the domain type that is trained. In StyleGAN2-ADA study, StyleGAN2 is extended by including the condition variable while training. The number of conditions adds an embedding layer to the generator network. The condition is given to it and the output is concatenated to the random latent vector z. This concatenated vector is passed to the mapping network and the style vector is obtained exactly as in the original StyleGAN2. The rest of the network works in the same way. Thus, StyleGAN2-ADA can be conditionally trained with a small amount of data. This is especially important for those areas that suffer from a lack of data.

4.2.2. GANSpace

Although Generative Adversarial Networks (GANs) are the most powerful image synthesis models, it is really difficult to edit or control the output images. By selecting random vectors from the latent spaces of GANs, many different images can be generated. However, it is problematic to make changes to an image generated from a fixed latent vector. It requires very expensive processes such as training multiple GAN models or using supervised models. Training a deep learning model for a specific purpose can require a lot of computational power and time. Therefore, controlling GAN with desired results becomes a difficult problem. Some initial works [92, 93] have been published for manipulating the output of GAN models, but they have quite limited editable options. In addition, some studies [94, 95] work with supervised models to produce controllable images. Even though some of these works do the desired job, they are far from easy to process the output images with many different options.

In late 2020, Härkönen et al. introduce a new method called GANSpace [12] that allows to control the generated images. They show that the latent or feature space of GANs can be manipulated in different directions, which can be found by applying Principle Component Analysis (PCA) to these spaces. This operation can be easily performed without requiring too much computational power and can be very effective on many different features of images such as shape, pose, lighting [12]. Another advantage of this method is that it does not depend on a particular GAN. It is applicable to different types of GAN. For example, StyleGAN2 [96] uses a mapping network that converts the latent vector z into the style vector w. The vector space W allows GANSpace to find directions, since these vectors are responsible for creating the style of the output image. On the other hand, BigGAN [33] does not have a style vector like StyleGAN. The output of the early layers allows to manipulate and control the output images.

The basic logic of the GANSpace method for StyleGAN2 is as follows: N sample vectors ($z_{1:N}$) are randomly selected from the latent space. These are converted to style

vectors $w_{1:N}$ via a mapping network. Then, PCA is applied to the $w_{1:N}$ vectors. The number of components can be arbitrary, but the studies in the paper [12] show that 120 components are sufficient to control general features of images. The later components have no significant effect on the output images. Finally, any component you choose can manipulate, with varying strength, any style vector w_i. In the equation

$$w' = w + Vx \tag{4.1}$$

V refers to the direction vector and x refers to the value that indicates how strongly it can be applied. This vector w_i is passed to the StyleGAN model normally and it proceeds to generate the output image.

The effect of components on output images changes the layer range they are applied to. While some components show successful control and changes after being moved along some layers, others can be effective by moving along all layers [12]. Sometimes moving along more layers causes entangled changes in the output images. For example, moving along component v at all layers can change both the gender and hair shape for the model trained on the FFHQ dataset. However, if the number of layers is reduced and it is applied to only some of the early layers, only the gender can be changed. These changes can be bidirectional. The value x, indicating the strength of the manipulation, can be either negative or positive. For example, a latent vector z that produces the image of a young man can be manipulated as a little boy or an old man with the same component.

4.2.3. Proposed Solution

Our main goal in this study is to augment existing chest X-ray datasets using deep learning methods. The augmented data must consist of chest X-ray images with one of the 5 observations we discussed previously. Aforementioned studies StyleGAN2-ADA [13] and GANSpace [12] are perfect candidates for our purposes. We think

StyleGAN2-ADA can be conditionally trained with 5 observations and the generated X-ray images can be manipulated as desired. This allows controllable generation of conditional X-ray images. While the generation process gives us the data with the desired observation, the manipulation gives us the flexibility to change the strength of the observation or to edit observation-independent features of X-rays to create data diversity.

Our study extends the GANSpace method in a conditional way. The original method works with StyleGAN2, an unconditional generator that provides the simplest controllable directions. Our Conditional GANSpace consists of StyleGAN2-ADA instead of StyleGAN2 to generate conditional X-ray images. This allows us to synthesize images in a more controllable way. We can manipulate and determine not only the class, but also the features of the output X-ray images. In GANSpace, PCA is applied to style vectors $w_{1:N}$ mapped from randomly sampled latent vectors $z_{1:N}$. Our contribution is to add a condition vector to the random latent vectors before converting them to style vectors, just as StyleGAN2-ADA does. When PCA is applied to conditional style vectors, the components come from the distribution of a particular class. This means that the components must be found separately for each class. When using unconditional GANSpace, a single PCA operation is sufficient to find the directions. In our case, it is a conditional GANSpace and the number of PCA operations that must be performed is equal to the number of classes for which the generator has been trained.

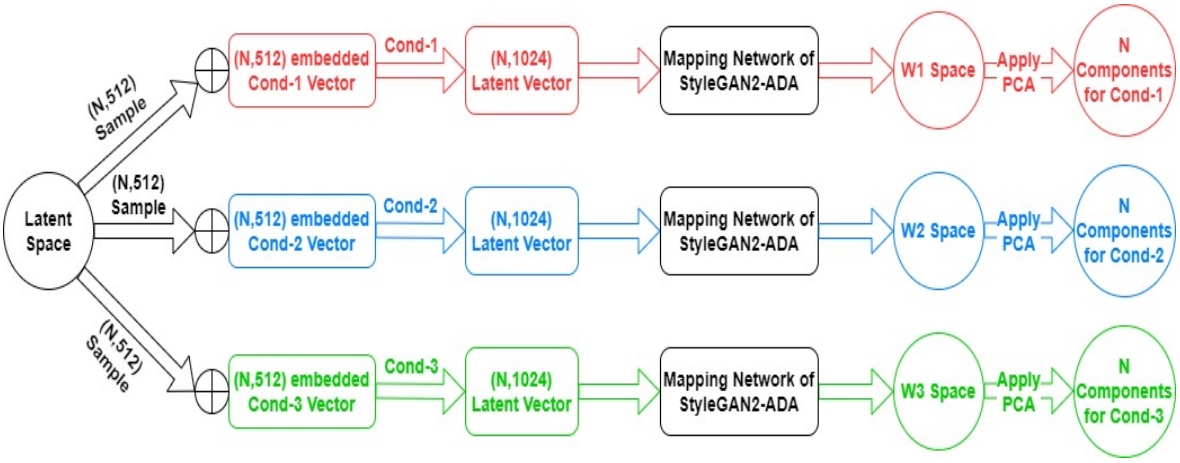


Figure 4.8. N samples are chosen randomly from latent space. Embedded conditional vectors are concatenated to them. Obtained vectors converted to style vectors via mapping network of pretrained StyleGAN2-ADA. PCA is applied on each style space and components are computed for each class.

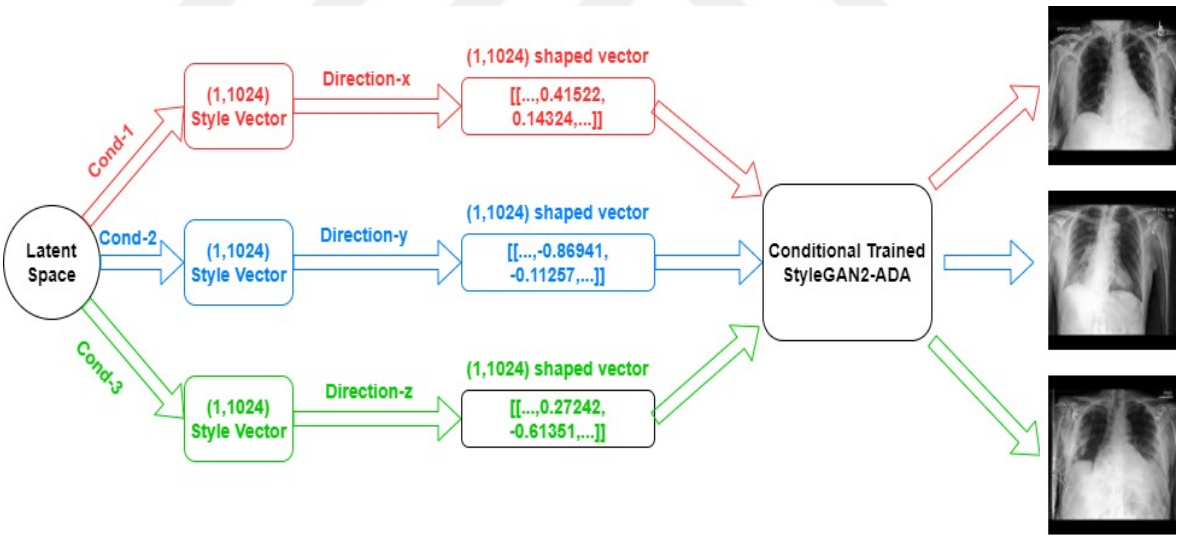


Figure 4.9. One random sample is chosen from latent space and converted to style vector as showed in Figure 4.8. Predefined components found for each class from their own component set are added to style vectors. Manipulated vectors are given to pretrained StyleGAN2-ADA. Final output X-ray images are obtained. Output X-ray images are obtained from our augmented dataset.

More precisely, we trained a generator model with 5 classes representing the observations. Therefore, the PCA operation is performed 5 times for each class separately. While the same directions can be found in each component set, completely different ones can also be obtained. Our 5 different component sets belong to the classes of cardiomegaly, edema, atelectasis, pleural effusion, and consolidation. For instance, the directions found for cardiomegaly may consist of some directions about the change in heart size while others do not. This is because the training patterns identified with cardiomegaly are directly associated with this feature. Since the distributions of the style vectors of the classes are separated in space, their PCA calculations result in different directions. In this way, we can create X-ray images that show observations more clearly.

Now, we will explain how Conditional GANSpace works. In Figure 4.8, N random samples from the latent space are repeated 3 times for 3 different classes. Embedded conditional vectors are created for each class using the embedding layer of the pre-trained StyleGAN2-ADA model. These vectors are concatenated with the latent vectors and passed to the mapping network. The result of this operation is a class-based style vector space. After PCA is applied to these spaces, 3 different sets of components are available to find directions for 3 classes. In Figure 4.9, the latent vector is randomly selected and converted into a style vector. Predefined direction components are added to them for manipulation. The manipulated vectors are passed to the pre-trained StyleGAN2-ADA model and the output images are generated as seen at the end of the figure.

4.3. Manipulation of Encoded Latent Vectors

As mentioned in the previous section, we use an extended version of GANSpace to manipulate the latent space to find useful directions that can be used for augmentation. In this method, the latent vectors are randomly selected for each operation. Although we know at the beginning of the process to which class the output image will belong, there is no way to find out the more detailed features of the generated X-ray image

in this process. We believe that we have a different kind of control over the output X-ray images when we decide which X-ray images to manipulate. We can manipulate X-rays with no observation. To do this, we need to find their latent vectors, which allows us to reconstruct the original X-ray image and which can be manipulated before reconstruction. The search for latent vectors is another problem referred to in the literature as image inversion. In recent years, some successful methods have been presented. We use Restyle Encoder [14] to invert an image into its latent code.

4.3.1. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement

Latent space manipulation has become a popular topic in recent years. These methods add meaningful direction vectors to the latent vectors of generative models such as StyleGAN. This allows a variety of changes to be made to the output images. Although new and more effective methods have been introduced in this area, they are limited to manipulating only random vectors. The solution to overcome this limitation is to invert real images into their latent vectors. Some previous works focused on two different types of methods, namely, encoder based and optimization based inversion. Encoder based methods attempt to generalize the inversion process and work with a trained model. They use a pretrained generator that reconstructs images from the inverted latent code and provides feedback to the encoder during the training phase. At the end of training, the trained encoder model can invert the given images into latent vectors in a forward pass. On the other hand, optimization based methods work per image. They start with a random vector for each image. A pretrained generator produces an output image in each iteration. The loss value is calculated based on the original image and the generated image. The loss value gives feedback to the model and it updates the vector accordingly until the loss value is small enough. Optimization based methods give better results but they can work per image and slower. Encoder based methods are preferable as they are faster and can generalize the encoding process.

Alaluf et al. introduce an encoder based method [14] that can provide better visual results compared to previous methods according to their comments in the paper. Their method departs from previous ones by providing an iterative solution. Others use a single forward pass in their learning phase. Restyle method concatenates the input image and the generated image from the previous iteration for a step t. The concatenated 6-channel image is passed to the encoder model. Residual latent vector Δ_t is obtained from the encoding operation. It indicates the offset between the previous and the current latent vector and is therefore added to the latent vector of the previous iteration w_{t-1}. This updated latent code w_t becomes the input to the generator, which is StyleGAN2. The generated image y_t goes back and becomes the input for the next iteration $t + 1$. This process repeats a small number of times. The iteration number does not exceed 10. Initially, a random latent vector w_0 is given to the generator and the output of y_0 becomes the first input, which is concatenated with the original image. The 6-channel input is generated by the input x and the generated image y_0.

4.3.2. Applied Method

GANSpace [12] allows to edit images by manipulating latent vectors. As mentioned earlier, this method takes a generator and uses its style space or feature space to find directions. Principle Component Analysis (PCA) is applied to the latent space of the generator and many different directions can be found. The directions are represented by some vector values that can edit images in different ways. The directions are used in many different studies to manipulate the latent random code as in the method that we demonstrate. Although this process is very useful in many fields, the randomness of the latent code may make the method useless in some cases. The status of the output image generated from the original latent code must be known before manipulation. Therefore, we use the inverted latent code from chest X-ray images that we selected earlier. Restyle Encoder [14] is used to invert X-ray images to their latent code. This inversion ensures that we get the latent code of X-ray images of healthy people. Any kind of observation can be added to this base X-ray image. In this way, we can generate hundreds of thousands of X-ray images as long as we have healthy

X-ray images.

Restyle Encoder is trained with the CheXpert [97] dataset for the inversion of chest X-ray images. An important detail here is that the generated X-ray images must resemble the original image as closely as possible. The loss of information must be minimal. In the original Restyle work [14], the authors use general purpose ResNet-50 feature extractor while computing the loss of similarity for non-face domains. However, we replace it with DenseNet-121 feature extractor that we trained on the CheXpert dataset. Our goal is to reduce the loss value while training Restyle Encoder and obtain better reconstructed images.

The inverted latent code in the Restyle Encoder corresponds to the style vector of StyleGAN2 [96]. This prevents the use of StyleGAN2-ADA as our proposed solution in the previous section, since StyleGAN2-ADA adds the condition vector to the latent vector before mapping it to the style vector. This means that the style vector contains the class information of the image to be generated. Therefore, we use the StyleGAN2 generator and unconditional generation.

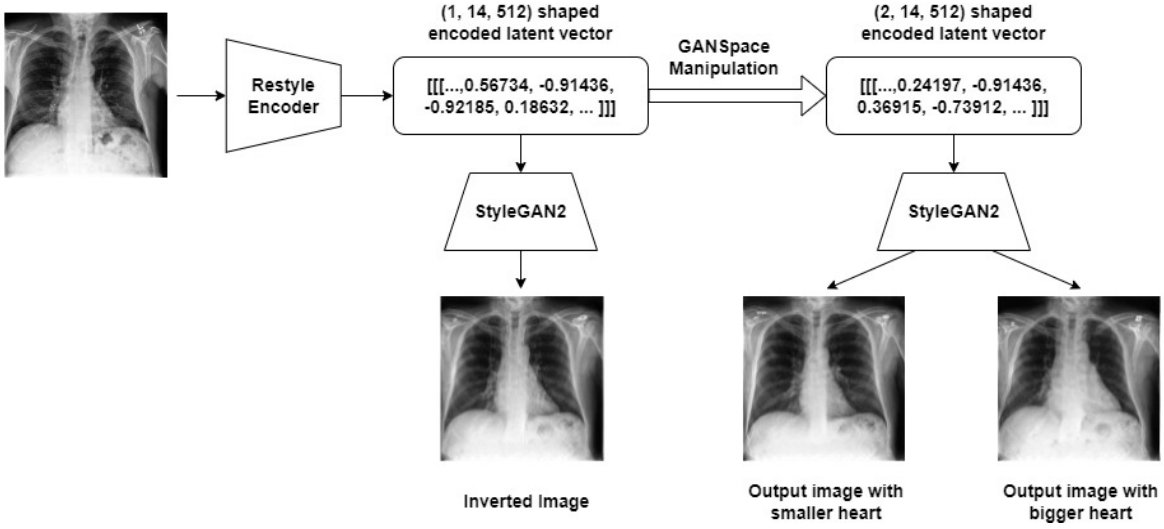


Figure 4.10. Healthy X-ray is given to Restyle Encoder. Output style vector can manipulate with GANSpace manipulation. Without manipulation, original image (Inverted Image) can be reconstructed. In the example, heart size is manipulated in both positive and negative direction.

Principal Component Analysis (PCA) is applied and 120 components are calculated using our pre-trained StyleGAN2 generator. Each component was carefully analyzed and many useful directions were found. Some of these directions include symptoms of cardiomegaly, edema, atelectasis, pleural effusion, and consolidation. Shoulder, lung or text on X-rays related changes are other examples of directions. Although this method can edit healthy X-rays in many ways, accurate labeling of output images must be monitored. This prevents us from producing thousands of X-ray images for each class without monitoring.

Figure 4.10 shows the pipeline of this method. A healthy chest X-ray is taken from the CheXpert dataset. Inversion is performed using a previously trained Restyle Encoder. When this latent code is passed to the generator without any manipulation, it provides the exact image with the encoder input. On the other hand, predefined directions can be moved along this inverted latent code. In our representative illustration, we add two different components to the latent code and get two manipulated latent vectors. When these are given in sequence to the StyleGAN2 generator, two different X-ray images are generated. In the figure, we select the component as the direction that can add or remove Cardiomegaly. This observation is diagnosed by looking at the heart size. If the size of the heart is much larger than a healthy heart, this can be diagnosed as Cardiomegaly. As we move along this component in both positive and negative directions, the output images will show a smaller or larger heart than the original image. In the negative direction, the heart becomes smaller; in the positive direction, it becomes larger.

4.4. Dataset

As we mentioned earlier, there are not many options for medical imaging datasets. We chose the CheXpert [97] dataset because it consists of the most X-ray films with the best labeling system, which we will explain in detail later. The CheXpert dataset is introduced by a team of 20 people from the departments of computer science, medicine, and radiology at Stanford University. The dataset includes 224,316 chest radiographs

taken from 65,240 patients. These radiographs were taken at Stanford Hospital between October 2002 and July 2017. According to the general diagnosis in the reports, it is decided that each radiograph will be labeled with 14 different observations. While 12 of these labels include the presence or absence of an observation, one of the labels is "No finding" in order to indicate that none of the observations were found. The last label, "Support Devices", indicates whether any auxiliary devices were used on the patient during the radiography. The other labels can be seen in the Figure 4.11. The dataset also consists of a validation set containing 200 labeled radiographs taken to evaluate uncertainty labeling. These radiographs were labeled by 3 experts, unlike the other 224K radiographs.

Pathology	Positive (%)	Uncertain (%)	Negative (%)
No Finding	16627 (8.86)	0 (0.0)	171014 (91.14)
Enlarged Cardiom.	9020 (4.81)	10148 (5.41)	168473 (89.78)
Cardiomegaly	23002 (12.26)	6597 (3.52)	158042 (84.23)
Lung Lesion	6856 (3.65)	1071 (0.57)	179714 (95.78)
Lung Opacity	92669 (49.39)	4341 (2.31)	90631 (48.3)
Edema	48905 (26.06)	11571 (6.17)	127165 (67.77)
Consolidation	12730 (6.78)	23976 (12.78)	150935 (80.44)
Pneumonia	4576 (2.44)	15658 (8.34)	167407 (89.22)
Atelectasis	29333 (15.63)	29377 (15.66)	128931 (68.71)
Pneumothorax	17313 (9.23)	2663 (1.42)	167665 (89.35)
Pleural Effusion	75696 (40.34)	9419 (5.02)	102526 (54.64)
Pleural Other	2441 (1.3)	1771 (0.94)	183429 (97.76)
Fracture	7270 (3.87)	484 (0.26)	179887 (95.87)
Support Devices	105831 (56.4)	898 (0.48)	80912 (43.12)

Figure 4.11. All observations with presence, absence and uncertainty numbers of CheXpert dataset.

Because it takes too much time and effort to label 224,316 X-ray images, the researchers use a labeler to annotate each image. This labeling program is rule-based and finds the observations in the X-ray reports in 3 steps. These are mention extraction, mention classification, and mention aggregation [97].

The first step of the automatic labeling system finds mentions about the observations in the list in the impression section of the X-ray reports. This section contains the experts' comments on the findings in the report.

Mention classification step aims to determine whether found mentions are positive, negative or uncertain. To classify any of them, each sentence or phrase is processed. These processes are the rules that were previously defined. These processes include classification steps used in Natural Language Processing. First, the sentences are split and tokenized using the NLTK library. If a sentence consists of a assessment about presence of one of the pathologies, it is classified as positive for that pathology. If it does not contain an indication of a pathology, it is marked as negative. If there are expressions of possibility that could be sentences containing "may" or "might", they are classified as uncertain.

Finally, mention aggregation step is applied each mention to obtain the final annotations on the radiographs. The previous step determines the positive, negative or uncertain situations of each sentence. This step determines the overall classes by reviewing each sentence. The X-ray image is labeled positive if there is at least one positive mention for an observation. An uncertain label is assigned if there is no positive mention. In this situation, one uncertain mention is sufficient to be noted as uncertain. Similarly, if mention does not contain any of these, it is annotated as negative. If none of these three labels exist, *blank* is assigned for that observation. A positive label is indicated as "1", a negative label is indicated as "0", and uncertainty is indicated as "*u*" in the dataset. If all 12 pathologies are labeled as negative, "No Finding" is labeled as a positive "1". Table 4.1 shows the example labels for the radiographs given in Figure 4.12. Positive labels are "1.0", negatives are "0.0" and uncertain ones are "-1.0" to match the data type.



Figure 4.12. Different type of radiograph examples from CheXpert dataset.

Table 4.1. Label information of the radiographs in the Figure 4.12.

No Finding	**Enlarged Cardiomediastinum**	**Edema**	**Consolidation**	**Pneumothorax**	**Pleural Effusion**	**Support Devices**
1.0	0.0		0.0		0.0	
1.0	0.0		0.0		0.0	
	-1.0	1.0	-1.0	-1.0	-1.0	1.0

They also evaluate the labels to verify the success of the labeler. For this purpose, they randomly select 1000 radiographs from the dataset and 2 experts annotated each of these images separately. Then they become together and review the annotated images in different ways. For another approach to evaluate the auto labeler, they use another labeling program proposed by Peng et al. [98]. They compare these two labelers with the NIH dataset. The labeler presented by Stanford outperforms the other in all categories.

Since the dataset does not contain binary labels for the observation, the authors offer several approaches to the problem of uncertain labels. The first approach is U-Ignore, where they propose to ignore uncertainty labels during training. Another approach converts all uncertainty labels to positive (1), which they call U-Ones, or to negative (0) which they call U-Zeroes. In Self-Training approach, the U-Ignore approach is first applied and the system is trained accordingly. After training, the dataset is evaluated to fill the uncertain labels with 0s or 1s. The last approach is to treat the dataset with 3 classes and train it accordingly.

5. EXPERIMENTS AND RESULTS

This section will provide detailed information about the training settings for each method used and how the augmented datasets are created. Then we will give details about the classification method and its training settings. Finally, we show how the methods affect the classification results.

5.1. Visual Results of Image Inpainting Model

Our method was implemented based on RFR-Net [11]. The details of the network remain the same, as they are very well designed for the task of image inpainting. The Adam optimizer was used for the generator network. Training was performed using a mini-batch of size 6. Since our dataset contains 224k images, the model was trained with 37k iterations in an epoch. The model was trained for 10 epochs (370k iterations) with a learning rate of 0.0001 and the results were somewhat fuzzy. Therefore, we continued training by decreasing the learning rate to 0.00001. We trained 8 more epochs (296k iterations).

The size of the original X-ray images in the dataset is not square. The size of height and weight varies from 320 to 390. Since our inpainting method requires an input size of 224×224 and a square shape, we resized all images in the dataset to 224×224. Since the original size was not necessarily square, we also padded the resized images to preserve the original aspect ratio of the X-ray image itself.

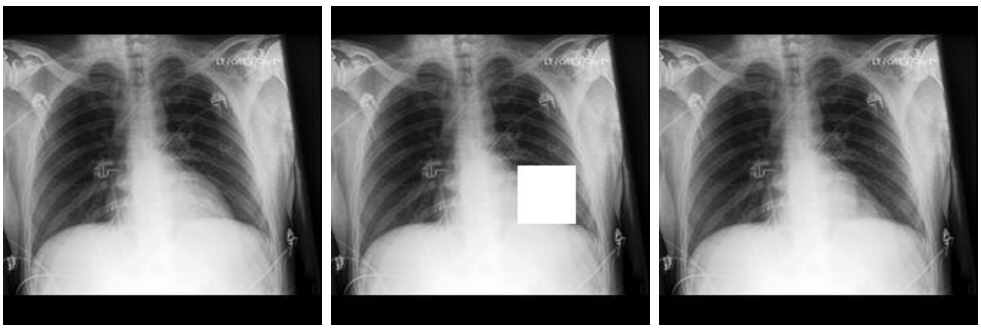


Figure 5.1. Original, randomly masked and inpainted radigraphs of patient00003.

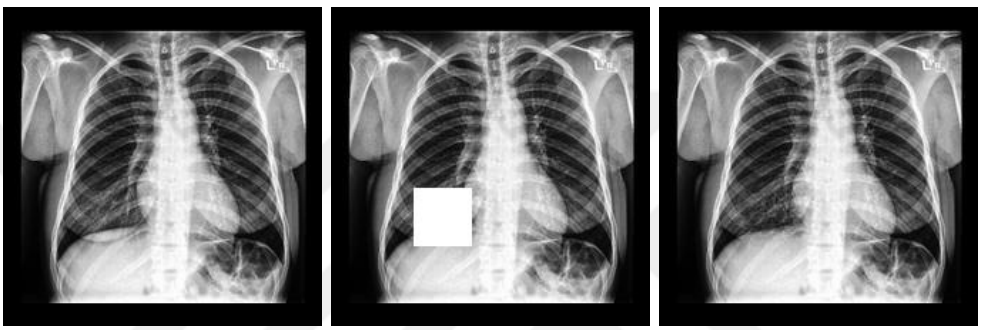


Figure 5.2. Original, randomly masked and inpainted radigraphs of patient00004.

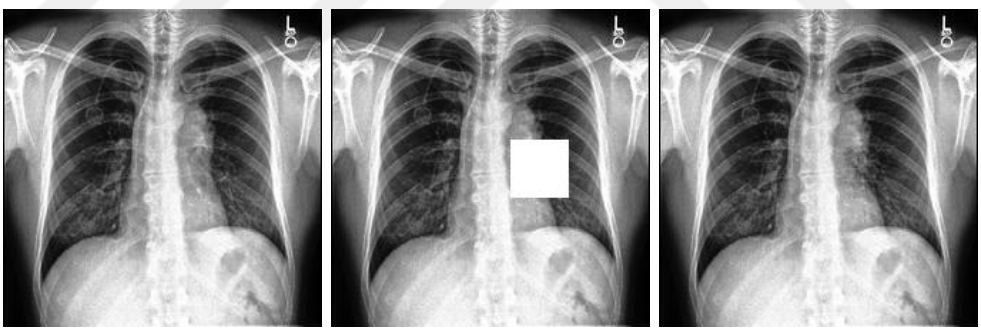


Figure 5.3. Original, randomly masked and inpainted radigraphs of patient00005.

We created random and square masks because other studies in the literature generally use square masks. Since our images were 224×224 in size, we thought a 64×64 mask would be sufficient. In general, the top, bottom, left, and right sides of the X-ray images are not useful for the inpainting task. These areas are usually black and just provide a blank background. Therefore, when creating random masks, we left 30 pixels of space on 4 sides of the X-ray images. This ensures that the masks are located on the chest and prevents irrelevant masks in empty areas.

We added another step to the masking process of the model. When creating random masks, we were careful not to mask the region where the observation takes place. This was necessary because inpainting these regions can cause the observation to be distorted and the X-ray image may no longer have the same label. Consequently, if an observation disappears due to inpainting, the entire labeling process must start over. This step guarantees that none of the labels on the X-ray images will change. For this purpose, we took the masks from the heatmaps mentioned in Section 4.1 and checked if there was any overlapping with the randomly generated mask. If there was more than 20%, we generated another one until we had more separate masks. If the overlapping was less than 20%, we removed the overlapping area from the generated mask and inpainted the remaining area. If we removed the overlapping area that was more than 20%, the remaining masked area became too small and inpainting was meaningless.

After obtaining the model, we created a new set of images by completely inpainting the original dataset. Using this method, we created 3 new datasets. In the next step, we measured whether these datasets helped to improve the classification results. First, we classified the original dataset, then we classified the augmented datasets by adding inpainted sets.

5.2. Evaluation Of Inpainted X-rays

The evaluation of this type of task can be done in 2 ways. One is based on purely human judgment, and the other is based on verification of improvement in other tasks such as classification. While the first method can be subjective and does not allow comparison with similar tasks, the result of the second method can provide the desired results. In addition, the inpainting task can also be used for data augmentation in many domains. The success of data augmentation can be tested in classification. For these reasons, we decide to use a good classification model for chest X-ray to test whether our inpainted images can improve the result.

Stanford University also held a competition [97] on the occasion of the release of the CheXpert dataset. The goal of this competition is to find the best model that can classify the X-rays in the CheXpert dataset according to the given labels. Since we are using the CheXpert dataset [97] from Stanford, we chose a model that has an open source code and gives one of the best results in this competition. This model was proposed by the authors of PCAM [10].

Because the project contains randomness, we ran each experiment 3 times. Then we averaged the 3 results. We first trained the original dataset without any augmentation and took the model with the highest AUC. Next, we created 3 different datasets by randomly inpainting the X-ray images. During the inpainting process, we controlled the area where the observation takes place as mentioned in the previous sections to preserve the labels. After we finished the inpainted sets, we trained our model with the new set. We improve the classification results by 1.6%. While the AUC of the original dataset is 86.1%, the AUC of the augmented set is 87.7%. The results can be seen in the Table 5.1.

5.3. Evaluation of Conditional GANSpace

In this section, we refer to the dataset augmented with StyleGAN2-ADA as ADA, the dataset augmented with our cGANSpace method as CGA, the dataset augmented with our inpainting based method as IA.

To provide a basis for our cGANSpace experiment, we first decided to analyze the generation of X-ray images using StyleGAN2-ADA. This experiment was performed to determine whether feature control during conditional image generation can improve the classification result of it. cGANSpace method generates X-ray images with controlled X-ray features, while StyleGAN2-ADA generates only conditioning with classes. To this end, we generated 40K X-ray images of each class using our pre-trained StyleGAN2-ADA generator without any manipulation. This means that 200K unobserved X-ray images with labels were added to the filtered CheXpert dataset containing 100K images.

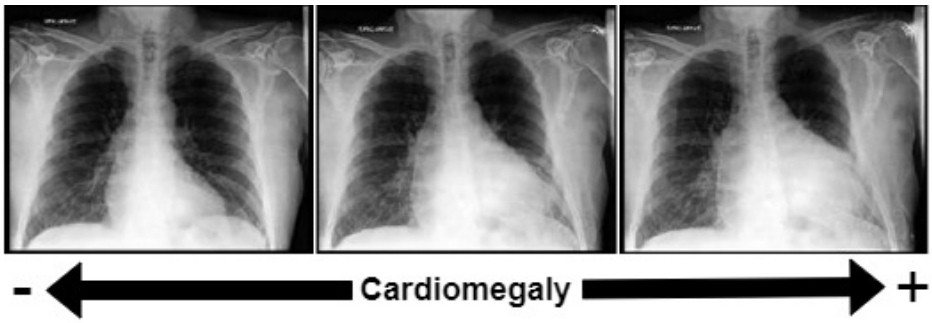


Figure 5.4. Manipulation of cardiomegaly on X-ray generated with cardiomegaly.

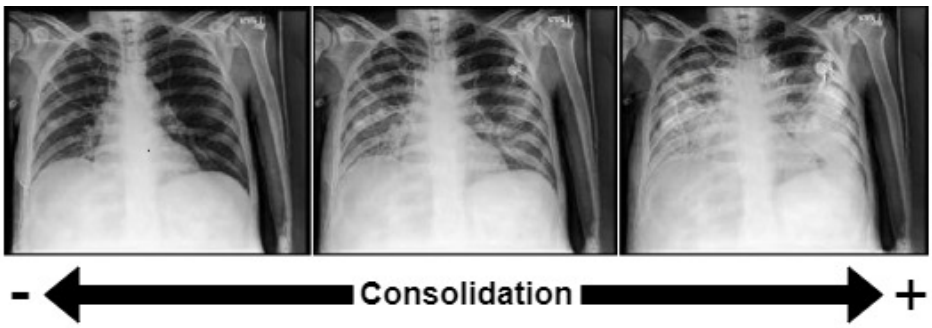


Figure 5.5. Manipulation of consolidation on X-ray generated with consolidation..

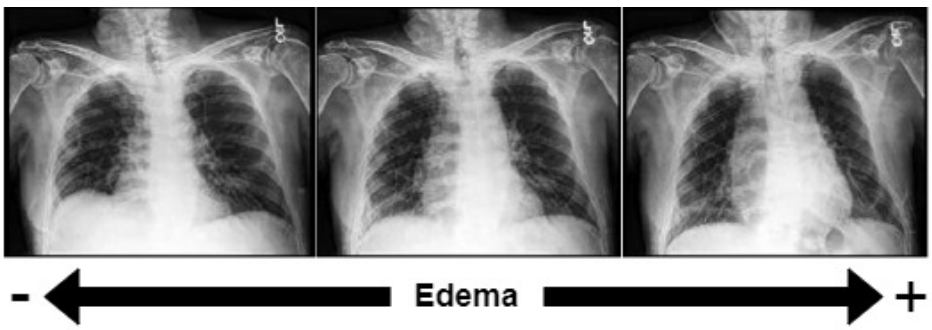


Figure 5.6. Manipulation of edema on X-ray generated with edema class.

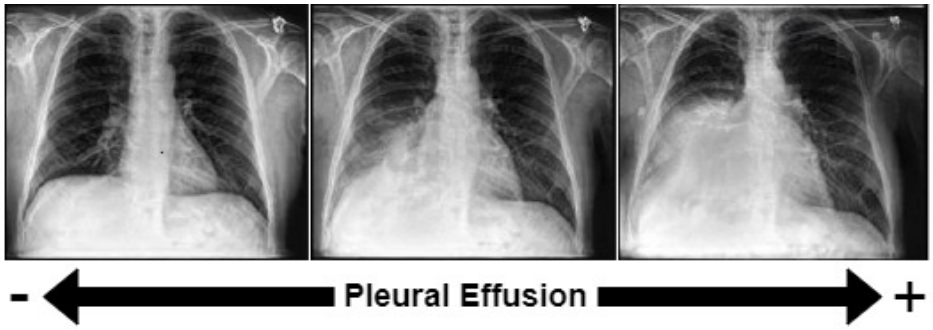


Figure 5.7. Manipulation of pleural effusion on X-ray generated with pleural effusion.

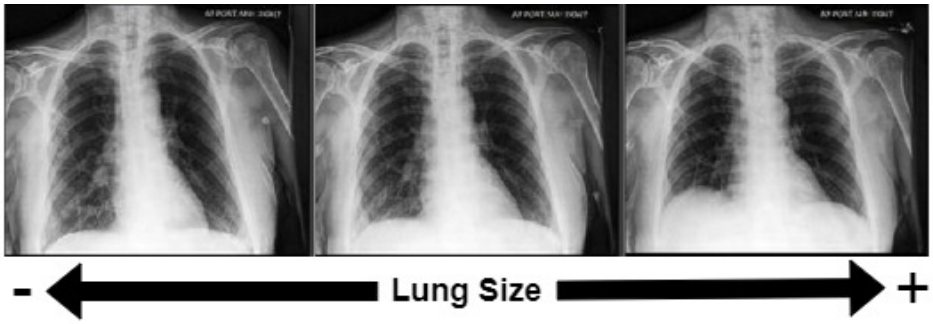


Figure 5.8. Manipulation of lung size on X-ray generated with atelectasis.

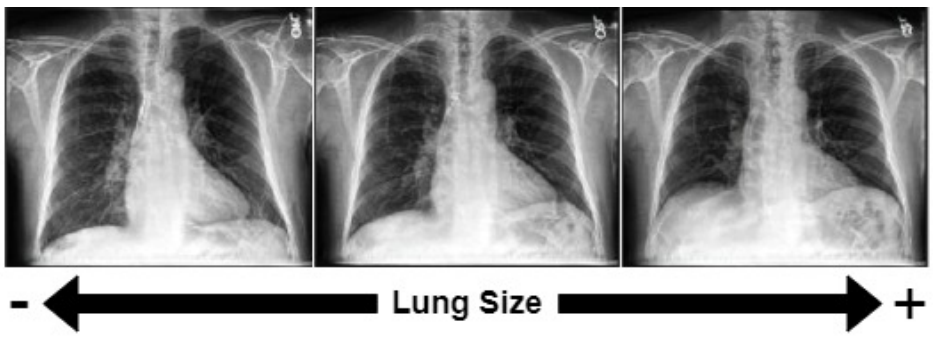


Figure 5.9. Manipulation of lung size on X-ray generated with cardiomegaly.

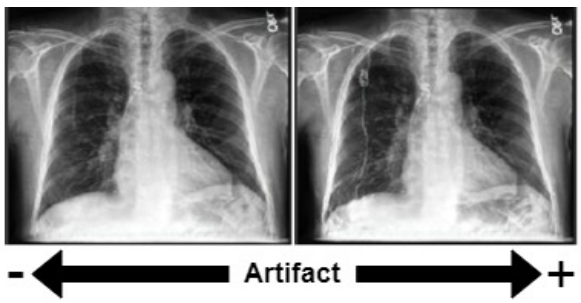


Figure 5.10. Manipulation of artifact on X-ray generated with cardiomegaly.

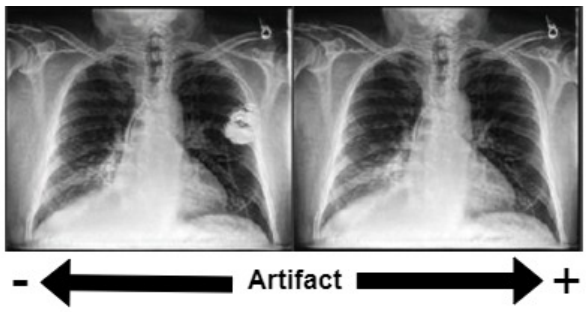


Figure 5.11. Manipulation of artifact on X-ray generated with consolidation.

Table 5.1. Quantitative results of experiments. CheXpert Classifier [10] is used for experiments. First column shows 5 different observations that are classified. Second column indicates classification experiments only with original CheXpert [97] dataset. Experiments in the third column includes dataset augmented with inpainted images. The forth column indicates the classification results of dataset augmented with StyleGAN2-ADA without any manipulation. Finally, last column shows the results of dataset augmented with Conditional GANSpace.

Class/Dataset	**Original Set**	**Augmentation Using Heatmap Based Inpainting**	**Augmentation Using StyleGAN2-ADA**	**Augmentation Using cGANSpace**
Atelectasis	0.8506	0.8416	0.8534	0.8636
Cardiomegaly	0.7598	0.8099	0.8046	0.8232
Consolidation	0.8947	0.9270	0.8991	0.9170
Edema	0.8993	0.9132	0.9002	0.9076
Pleural Effusion	0.9009	0.8925	0.9105	0.9224
Mean AUC	0.8611	0.8768	0.8736	0.8847

Thus, we had an augmented dataset with a total of 300K images. The experimental setup was the same as for the IA dataset. We trained the classifier [10] with this augmented set. We ran it three times. The experiment resulted in a mean AUC of 87.36%. This improves the result of the original set, which is 86.11%, by 1.25%. However, the result of the IA dataset cannot be surpassed. Its mean AUC, which is 87.68%, is higher than the ADA dataset. Detailed results can be seen in the Table 5.1.

Following the StyleGAN2-ADA experiment, we performed the experiment for the cGANSpace method. Our proposed method includes three steps for data augmentation. In the first step, PCA was applied to each class in our experiments. The number of classes is 5, corresponding to the observations of cardiomegaly, edema, atelectasis, pleural effusion, and consolidation. After PCA, we obtained 120 components for each of 5 different classes.

Table 5.2. The first column contains brief information about directional changes on X-ray images. The first number in the other columns indicates the component number and the numbers in parentheses indicate the range of layers where the directions are effective.

Explanation of Direction	**Cardiomegaly**	**Edema**	**Atelectasis**	**Pleural Effusion**	Consolidation
Add & Remove Artifact	1 (7-8)	4 (2-4)	2 (4-5)	1 (5-7)	2 (4-6)
Shift & Tilt X-ray	2 (2-4)	4 (3-4)	3 (1-5)	2 (1-3)	5 (4-6)
Add & Remove Artifact-2	6 (6-8)	8 (3-4)	4 (5-7)	5 (1-2)	6 (6-9)
Expand Lung	8 (5-7)	6 (3-5)	6 (4-8)	3 (4-6)	7 (1-2)
Edit Text	13 (0-4)	11 (6-7)	10 (1-2)	10 (1-3)	9 (8-9)
Expand & Narrow Shoulder	19 (2-4)	16 (5-6)	18 (1-3)	19 (3-5)	19 (2-3)
Edit Clavicle	20 (2-4)	21 (5-6)	28 (3-5)	13 (1-2)	18 (7-9)
Elongate Lung	21 (4-6)	29 (2-3)	23 (7-8)	22 (6-9)	25 (4-5)
Enlarge Lung	22 (2-5)	20 (2-4)	22 (3-4)	26 (1-2)	23 (1-2)
Edit Ribs	32 (1-2)	30 (2-3)	29 (7-9)	34 (4-6)	30 (1-2)
Cardiomegaly	10 (3-5)	-	-	-	-
Edema	-	29 (4-6)	-	-	-
Atelectasis	-	-	33 (6-7)	-	-
Pleural Effusion	-	-	-	18 (6-7)	-
Consolidation	-	-	-	-	11 (7-9)

In the second step of the experiment, medical student Yasin Durusoy and radiology specialist Dr. Görkem Durak helped us analyze the components. Since this process requires medical expertise, we needed their help to find and verify directions. We prepared an experimental setup on Google Colab so that Yasin could analyze each component to find directions. Each component was analyzed in a different range of layers. Finally, effective directions were found and 10 of them were selected to manipulate the random vectors in the X-ray generation phase. On the other hand, another direction was found for each class to manipulate its own observation. In other words, another direction was added to the predefined 10 directions to either increase or decrease the effect of the observation on the X-ray image. All of these directions, along with their functions and layers at which they are effective can be found in the Table 5.2. When we had created the images and videos of the manipulated X-ray images, we showed them to Dr. Görkem Durak. We asked his opinion as a specialist in radiology and were informed about which directions could be useful and what benefits they could have from a medical point of view.

In the final step, conditional image generation was used for 5 observations. As mentioned earlier, conditional image generation was performed using the extended version of GANSpace with StyleGAN2-ADA. The total number of X-ray images generated was 200K. For each class, 40K images were generated so that the data generation process was balanced. Random and different seeds were used for each latent vector. After obtaining the style vectors from the mapping network, we applied manipulations selected from 11 predefined directions. To create opposite samples from each direction, we moved along the components in a positive and a negative direction for each random vector. The negative direction can decrease the effect of a feature or make the feature disappear. On the other hand, a positive direction can increase the effect of a feature. It can add a feature even if it is not in the original latent vector. For example, the heart size component was applied in a negative direction to create a smaller heart, and applied in a positive direction to create a larger heart. We continued this process until we generated 40K X-ray images for each class.

Using the cGANSpace method, we were able to increase our dataset threefold. After generating 200K new labeled images as in the StyleGAN2-ADA experiment, we added 100K images from the filtered CheXpert dataset to obtain a dataset of 300K size. Our classification scenario was also the same as the previous experiments to achieve consistent results. Employed classifier was trained with the current augmented dataset. When we compare the results with the original dataset, we improved the mean AUC from 86.11% to 88.47%. The score increased by almost 2.4%. Similarly, we improved the mean AUC of the IA dataset by almost 0.8%. In this experiment, our main concern was whether we could improve the score of the ADA dataset. While its mean AUC score is 87.36%, the score of the CGA dataset is 88.47%. Controllable generation improves the score by 1.1%. All these results can be found in the Table 5.1.

We provide details of our experimental setup and quantitative results for our cGANSpace method. We also show qualitative results for this method. We chose random seeds and generate X-ray images by manipulating predefined directions. While we can use conditions to determine the label when generating images, directions provide the ability to control the strength of the observation or other features of the images. In other words, we can increase the visibility of observations on X-ray images by moving along the appropriate directions. Figures from 5.4 to 5.11 show X-ray images created with different classes. We generated X-ray images with the classes cardiomegaly, consolidation, edema, and pleural effusion. Then we manipulated them with directions that change the effect of observation. The results for both directions are given for each observation in the Figures from 5.4 to 5.11. We also show that we can control other types of features of X-ray images such as lung size in cardiomegaly and atelectasis or the visibility of artifacts in cardiomegaly and consolidation.

5.4. Visual Results of Manipulation of Encoded Latent Vectors

As explained in Section 4.3, we combine two state-of-the-art methods to augment the current CheXpert dataset. Our goal is to manipulate healthy X-ray images by adding the desired observation in addition to other observation-independent manipu-

lations. Therefore, we use the Restyle encoder to invert X-ray images into their latent code. In its original version, Restyle uses ResNet for the MoCo loss function [14] in determining the similarity between input and synthesized image. Since ResNet was trained with ImageNet for general purposes and not for the X-ray domain, we decided to add DenseNet-121 model that we trained with CheXpert. In this way, we wanted to obtain lower loss values during training. After these changes, we trained our Restyle encoder and made it ready for X-ray image inversion.

On the manipulation side, we use GANSpace with the unconditional StyleGAN2 generator. First, we applied PCA with our pre-trained StyleGAN2 generator. Then we found useful and important directions. Some of these directions are those that increase heart size (an indication of cardiomegaly), lung size, shoulder height, or rib shape. By manually controlling each direction, we found disentangled directions for each of the 5 observations. In the next step, we manipulated the inverted vectors of the healthy X-ray images with these directions. After manipulation, we synthesized the final images using our generator.

In this method, we decided that the randomly synthesized thousands of images pose a problem in labeling. We cannot be sure that each direction is applied as desired and that the output images match the labels. Therefore, we did not perform a classification experiment on the augmented dataset. We report only qualitative results in this section.

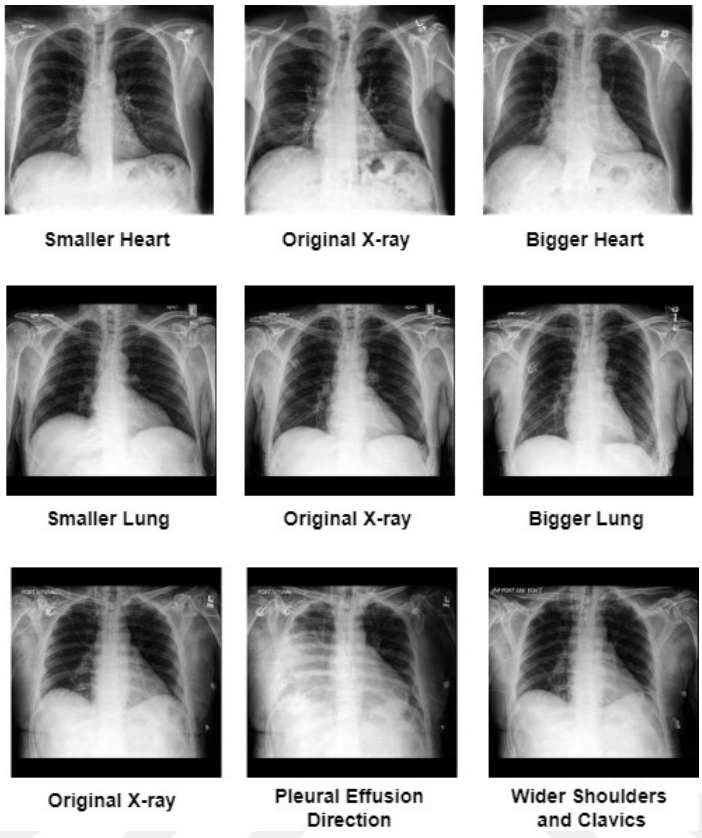


Figure 5.12. Qualitative results of Manipulation of Encoded Latent Vectors method. Four different directions are shown as example. First row shows changes in heart size, second one is example of lung size direction, next is pleural effusion direction and widening of shoulders-clavics.

6. CONCLUSION

In this thesis, we apply three methods for data augmentation on chest X-rays to improve pathology classification results. Our first method is based on Deep Image Inpainting, the second involves image synthesis with Conditional GANSpace, which is our extension in this thesis and the last is a manipulation of inverted healthy X-ray images.

For our first work, we create a heatmap-based inpainting method. We add the Probabilistic Class Activation Mapping [10] model to find the location of the observation on the X-ray image in the first step. This localization operation helps to prevent masking of critical areas, since we do not want to change the label of the X-ray image. After the location is determined, a random mask is chosen that does not overlap with the specified area. When the conditions are met, the masked image is passed to the RFR-Net [11] model, which is selected for the inpainting module.

GANSpace is a successful model for manipulating latent space, but it does not work with the conditional generator StyleGAN2-ADA. We extend the model so that it can work with it. This allows us to apply the conditional PCA operation and find directions accordingly. This means that the component groups of each observation cannot contain other observation specific directions when we analyze the components obtained by conditional generator. Thanks to this advantage, we can synthesize X-ray images manipulated by observation specific directions. A large set of data generation options is available for data augmentation.

Finally, we show the combination of image inversion and latent space manipulation. Unconditional latent space manipulation is meaningless because we do not know the label of the image synthesized from the random latent vector. To solve this problem, we invert healthy images into their latent vector and then use it for manipulation. If we find cardiomegaly or edema direction, we can manipulate the healthy

image accordingly. This creates freedom in synthesizing images with observation.

The results of the experiment show that improvement is achieved with two types of separate augmentation. The classification module is first trained with the original CheXpert data and achieved an mean AUC of 86.1%. The first augmentation method, heatmap-based inpainting, yields an improvement of almost 1.6% and the mean AUC increases to 87.7%. The conditional GANSpace method improves the mean AUC even more. The highest mean AUC value is obtained with this augmentation and is 88.5%. These results also show that data augmentation can always be an option for model training. It can make an important contribution to big data studies. Moreover, this study shows how powerful GANs are as synthesizers. All qualitative and quantitative results highly support this idea.

As future work, we can work on a new type of GAN inversion method that can function as a conditional method. This could allow the conditional generator to be used as a discriminator. Also, the conditional GAN inversion method can work with the conditional GANSpace.

Another future work could be to use computed tomography as another type of medical data instead of X-ray images for deep learning models. Computed tomography is one of the most commonly used techniques in healthcare. It could be interesting to apply inpainting or latent space manipulation techniques to tomography images. Working on the same patient with many images from different angles could bring further benefits.

REFERENCES

1. Karras, T., S. Laine and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2018.

2. Sogancioglu, E., S. Hu, D. Belli and B. van Ginneken, "Chest X-ray Inpainting with Deep Generative Models", *arXiv preprint arXiv:1809.01471*, 2018.

3. Ortega, S., M. Halicek, H. Fabelo, R. Guerra Hernández, C. Lopez, M. Lejeune, F. Godtliebsen, G. Marrero Callico and B. Fei, "Hyperspectral Imaging and Deep Learning for the Detection of Breast Cancer Cells in Digitized Histological Images", *Proceedings of SPIE–the International Society for Optical Engineering*, Vol. 11320, p. 30, 2020.

4. Sun, W., B. Zheng and W. Qian, "Computer Aided Lung Cancer Diagnosis with Deep Learning Algorithms", *Medical Imaging 2016: Computer-Aided Diagnosis*, Vol. 9785, pp. 241 – 248, 2016.

5. Yu-Dong Zhang, C. T. W. Z., Vishnu Varthanan Govindaraj and J. Sun, "High Performance Multiple Sclerosis Classification by Data Augmentation and AlexNet Transfer Learning Model", *Journal of Medical Imaging and Health Informatics*, Vol. 9, pp. 2012–2021, 2019.

6. Lakhani, P. and B. Sundaram, "Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks", *Radiology*, Vol. 284, No. 2, pp. 574–582, 2017.

7. Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Networks", *Advances in Neural Information Processing Systems*, Vol. 2, p. 2672–2680, 2014.

8. Radford, A., L. Metz and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", *International Conference on Learning Representations, (ICLR)*, 2016.

9. Arjovsky, M., S. Chintala and L. Bottou, "Wasserstein GAN", *International Conference on Machine Learning (ICML)*, 2017.

10. Ye, W., J. Yao, H. Xue and Y. Li, "Weakly Supervised Lesion Localization With Probabilistic-CAM Pooling", *arXiv preprint arXiv:2005.14480*, 2020.

11. Li, J., N. Wang, L. Zhang, B. Du and D. Tao, "Recurrent Feature Reasoning for Image Inpainting", *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7760–7768, June 2020.

12. Härkönen, E., A. Hertzmann, J. Lehtinen and S. Paris, "GANSpace: Discovering Interpretable GAN Controls", *Advances in Neural Information Processing Systems*, Vol. 33, pp. 9841–9850, 2020.

13. Karras, T., M. Aittala, J. Hellsten, S. Laine, J. Lehtinen and T. Aila, "Training Generative Adversarial Networks with Limited Data", *Advances in Neural Information Processing Systems*, Vol. 33, pp. 12104–12114, 2020.

14. Alaluf, Y., O. Patashnik and D. Cohen-Or, "ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6711–6720, 2021.

15. Bertalmío, M., G. Sapiro, V. Caselles and C. Ballester, "Image inpainting", *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pp. 417–424, 2000.

16. Ružić, T. and A. Pižurica, "Context-Aware Patch-Based Image Inpainting Using Markov Random Field Modeling", *IEEE Transactions on Image Processing*, Vol. 24, No. 1, pp. 444–456, 2015.

17. Alilou, V. and F. Yaghmaee, "Exemplar-Based Image Inpainting Using SVD-Based Approximation Matrix and Multi-scale Analysis", *Multimedia Tools and Applications*, Vol. 76, p. 13795–13809, 2016.

18. Lu, H., Q. Liu, M. Zhang, Y. Wang and X. Deng, "Gradient-Based Low Rank Method and Its Application in Image Inpainting", *Multimedia Tools and Applications*, Vol. 77, No. 5, pp. 5969–5993, 2017.

19. Pathak, D., P. Krahenbuhl, J. Donahue, T. Darrell and A. Efros, "Context Encoders: Feature Learning by Inpainting", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016.

20. Yang, C., X. Lu, Z. Lin, E. Shechtman, O. Wang and H. Li, "High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6721–6729, 2016.

21. Yu, J., Z. Lin, J. Yang, X. Shen, X. Lu and T. Huang, "Generative Image Inpainting with Contextual Attention", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5505–5514, 2018.

22. Yu, J., Z. Lin, J. Yang, X. Shen, X. Lu and T. Huang, "Free-Form Image Inpainting with Gated Convolution", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4471–4480, 2018.

23. Hogeweg, L., C. I. Sánchez, J. Melendez, P. Maduskar, A. Story, A. Hayward and B. van Ginneken, "Foreign Object Detection and Removal to Improve Automated Analysis of Chest Radiographs", *Medical Physics*, Vol. 40, No. 7, p. 071901, 2013.

24. Yeh, R., C. Chen, T. Lim, A. Schwing, M. Hasegawa-Johnson and M. Do, "Semantic Image Inpainting with Deep Generative Models", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6882–6890,

2017.

25. Armanious, K., Y. Mecky, S. Gatidis and B. Yang, "Adversarial Inpainting of Medical Image Modalities", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3267–3271, 2018.

26. Shah, S., P. Ghosh, L. S. Davis and T. Goldstein, "Stacked U-Nets: A No-Frills Approach to Natural Image Segmentation", *arXiv preprint arXiv:1804.10343*, 2018.

27. Armanious, K., V. Kumar, S. Abdulatif, T. Hepp, S. Gatidis and B. Yang, "ipA-MedGAN: Inpainting of Arbitrarily Regions in Medical Modalities", *IEEE International Conference on Image Processing (ICIP)*, pp. 3005–3009, 2019.

28. Ibtehaz, N. and M. S. Rahman, "MultiResUNet : Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation", *Neural Networks*, Vol. 121, pp. 74–87, 2020.

29. Le, H. X., P. D. Nguyen, T. H. Nguyen, K. N. Q. Le and T. T. Nguyen, "A Novel Approach to Remove Foreign Objects from Chest X-ray Images", *arXiv preprint arXiv:2008.06828*, 2020.

30. Tran, M.-T., S. Kim, G.-S. Lee and H.-J. Yang, "Deep Learning-Based Inpainting for Chest X-ray Image", *The 9th International Conference on Smart Media and Applications*, pp. 267–271, 2020.

31. Ronneberger, O., P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

32. Goetschalckx, L., A. Andonian, A. Oliva and P. Isola, "GANalyze: Toward Visual Definitions of Cognitive Image Properties", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5744–5753, 2019.

33. Brock, A., J. Donahue and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis", *7th International Conference on Learning Representations (ICLR)*, 2019.

34. Shen, Y., C. Yang, X. Tang and B. Zhou, "InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PP, pp. 1–1, 2020.

35. Karras, T., T. Aila, S. Laine and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", *Sixth International Conference on Learning Representations (ICLR)*, 2018.

36. Shen, Y. and B. Zhou, "Closed-Form Factorization of Latent Semantics in GANs", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1532–1540, 2021.

37. Abdal, R., Y. Qin and P. Wonka, "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4432–4441, 2019.

38. Abdal, R., Y. Qin and P. Wonka, "Image2StyleGAN++: How to Edit the Embedded Images?", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8296–8305, 2020.

39. Abdal, R., P. Zhu, N. J. Mitra and P. Wonka, "StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows", *ACM Transactions on Graphics*, Vol. 40, No. 3, p. 1–21, 2021.

40. Richardson, E., Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro and D. Cohen-Or, "Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2287–2296, 2021.

41. Lin, T.-Y., P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, 2017.

42. Tov, O., Y. Alaluf, Y. Nitzan, O. Patashnik and D. Cohen-Or, "Designing an Encoder for StyleGAN Image Manipulation", *ACM Transactions on Graphics (TOG)*, Vol. 40, No. 4, pp. 1–14, 2021.

43. Moradi, M., A. Madani, A. Karargyris and T. Syeda-Mahmood, "Chest X-ray Generation and Data Augmentation for Cardiovascular Abnormality Classification", *Medical Imaging 2018: Image Processing*, p. 57, 2018.

44. Kora, S., "Evaluation of Deep Convolutional Generative Adversarial Networks for Data Augmentation of Chest X-ray Images", *Future Internet*, Vol. 13, No. 1, pp. 1–13, 2020.

45. Kermany, D. S., M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia and K. Zhang, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning", *Cell*, Vol. 172, No. 5, pp. 1122 – 1131.e9, 2018.

46. Waheed, A., M. Goyal, D. Gupta, A. Khanna, F. Al-Turjma and P. R. Pinheiro, "CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection", *IEEE Access*, Vol. 8, pp. 91916–91923, 2020.

47. IEEE, "IEEE Covid Chest X-Ray Dataset, 2020", `https://github.com/ieee8023/covid-chestxray-dataset`, accessed in Mar. 7, 2020.

48. Kaggle, "Covid19 Radiography Database, 2020", https://www.kaggle.com/tawsifurrahman/covid19- radiography-database, accessed in Mar. 7, 2020.

49. DarwinAI Corp., C., Vision and C. Image Processing Research Group, University of Waterloo, "COVID-19 Chest X-Ray Dataset Initiative, 2020", https://github.com/agchung/Figure1-COVID-chestxray-dataset, accessed in Mar. 7, 2020.

50. Albahli, S., "Efficient GAN-Based Chest Radiographs (CXR) Augmentation to Diagnose Coronavirus Disease Pneumonia", *International Journal of Medical Sciences*, Vol. 17, No. 10, p. 1439, 2020.

51. Bao, J., D. Chen, F. Wen, H. Li and G. Hua, "CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training", *IEEE International Conference on Computer Vision (ICCV)*, pp. 2764–2773, 2017.

52. Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2015.

53. Zagoruyko, S. and N. Komodakis, "Wide Residual Networks", *arXiv preprint arXiv:1605.07146*, 2017.

54. Guendel, S., A. A. A. Setio, S. Grbic, A. Maier and D. Comaniciu, "Extracting and Leveraging Nodule Features with Lung Inpainting for Local Feature Augmentation", *Machine Learning in Medical Imaging*, pp. 504–512, 2020.

55. Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

56. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recog-

nition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

57. Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2014.

58. Krizhevsky, A., I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, Vol. 25, 2012.

59. Tan, M. and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.

60. Pham, H., Z. Dai, Q. Xie, M.-T. Luong and Q. V. Le, "Meta Pseudo Labels", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11557–11568, 2021.

61. Ioffe, S. and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37, pp. 448–456, 2015.

62. Mirza, M. and S. Osindero, "Conditional Generative Adversarial Nets", *arXiv preprint arXiv:1411.1784* , 2014.

63. Odena, A., C. Olah and J. Shlens, "Conditional Image Synthesis With Auxiliary Classifier GANs", *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Vol. 70, p. 2642–2651, 2017.

64. Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*,

Vol. 115, pp. 211 – 252, 2015.

65. Bodla, N., G. Hua and R. Chellappa, "Semi-supervised FusedGAN for Conditional Image Generation", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 669–683, 2018.

66. Mino, A. and G. Spanakis, "LoGAN: Generating Logos with a Generative Adversarial Neural Network Conditioned on Color", *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 965–970, 2018.

67. Stap, D., M. Bleeker, S. Ibrahimi and M. ter Hoeve, "Conditional Image Generation and Manipulation for User-Specified Content", *arXiv preprint arXiv:2005.04909*, 2020.

68. Perarnau, G., J. van de Weijer, B. Raducanu and J. M. Álvarez, "Invertible Conditional GANs for Image Editing", *arXiv preprint arXiv:1611.06355*, 2016.

69. Wang, T.-C., M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8798–8807, 2018.

70. Antipov, G., M. Baccouche and J.-L. Dugelay, "Face Aging With Conditional Generative Adversarial Networks", *IEEE Internation Conference on Image Processing (ICIP)*, pp. 2089–2093, 2017.

71. Schroff, F., D. Kalenichenko and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.

72. Lin, Y.-J., P.-W. Wu, C.-H. Chang, E. Chang and S.-W. Liao, "RelGAN: Multi-Domain Image-to-Image Translation via Relative Attributes", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5913–5921,

2019.

73. Shen, Y., J. Gu, X. Tang and B. Zhou, "Interpreting the Latent Space of GANs for Semantic Face Editing", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9243–9252, 2020.

74. Zhu, J.-Y., T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, 2017.

75. Huang, X., M.-Y. Liu, S. Belongie and J. Kautz, "Multimodal Unsupervised Image-to-Image Translation", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–189, 2018.

76. Dar, S. U. H., M. Yurt, L. Karacan, A. Erdem, E. Erdem and T. Çukur, "Image Synthesis in Multi-Contrast MRI with Conditional Generative Adversarial Networks", *IEEE Transactions on Medical Imaging*, Vol. 38, No. 10, pp. 2375–2388, 2018.

77. Osokin, A., A. Chessel, R. E. C. Salas and F. Vaggi, "GANs for Biological Image Synthesis", *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2233–2242, 2017.

78. Xiao, T., J. Hong and J. Ma, "ELEGANT: Exchanging Latent Encodings with GAN for Transferring Multiple Face Attributes", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 168–184, 2018.

79. Emami, H., M. M. Aliabadi, M. Dong and R. B. Chinnam, "SPA-GAN: Spatial Attention GAN for Image-to-Image Translation", *IEEE Transactions on Multimedia*, Vol. 23, pp. 391–401, 2020.

80. Dolhansky, B. and C. C. Ferrer, "Eye In-Painting with Exemplar Generative Adversarial Networks", *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), pp. 7902–7911, 2017.

81. Chen, Y., W. Chen, C. Wei and Y. F. Wang, "Occlusion-Aware Face Inpainting via Generative Adversarial Networks", *IEEE International Conference on Image Processing (ICIP)*, pp. 1202–1206, 2017.

82. Yuan, Z., H. Li, J. Liu and J. Luo, "Multiview Scene Image Inpainting Based on Conditional Generative Adversarial Networks", *IEEE Transactions on Intelligent Vehicles*, Vol. 5, No. 2, pp. 314–323, 2020.

83. Chen, Y., H. Zhang, L. Liu, X. Chen, Q. Zhang, K. Yang, R. Xia and J. Xie, "Research on Image Inpainting Algorithm of Improved GAN Based on Two-Discriminations Networks", *Applied Intelligence*, pp. 1–15, 2020.

84. Salimans, T., I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, "Improved Techniques for Training GANs", *Proceedings of the 30th International Conference on Neural Information Processing Systems*, p. 2234–2242, 2016.

85. Heusel, M., H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 6629–6640, 2017.

86. Lopez-Paz, D. and M. Oquab, "Revisiting Classifier Two-Sample Tests", *International Conference on Learning Representations (ICLR)*, 2017.

87. Borji, A., "Pros and Cons of GAN Evaluation Measures", *Computer Vision and Image Understanding*, Vol. 179, pp. 41–65, 2018.

88. Xu, Q., G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu and K. Weinberger, "An Empirical Study on Evaluation Metrics of Generative Adversarial Networks", *arXiv preprint arXiv:1806.07755*, 2018.

89. Liu, G., F. Reda, K. Shih, T.-C. Wang, A. Tao and B. Catanzaro, "Image Inpainting for Irregular Holes Using Partial Convolutions", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 85–100, 2018.

90. Zhou, B., A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.

91. Ilse, M., J. M. Tomczak and M. Welling, "Attention-based Deep Multiple Instance Learning", *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Vol. 80, pp. 2127–2136, 2018.

92. Zhu, J.-Y., P. Krähenbühl, E. Shechtman and A. A. Efros, "Generative Visual Manipulation on the Natural Image Manifold", *The 14th European Conference on Computer Vision (ECCV)*, pp. 597–613, 2018.

93. Bau, D., H. Strobelt, W. Peebles, J. Wulff, B. Zhou, J.-Y. Zhu and A. Torralba, "Semantic Photo Manipulation with a Generative Image Prior", *ACM Transactions on Graphics*, Vol. 38, No. 4, p. 1–11, 2019.

94. Kulkarni, T. D., W. Whitney, P. Kohli and J. B. Tenenbaum, "Deep Convolutional Inverse Graphics Network", *Advances in Neural Information Processing Systems*, Vol. 28, 2015.

95. Jahanian, A., L. Chai and P. Isola, "On the Steerability of Generative Adversarial Networks", *arXiv preprint arXiv:1907.07171*, 2020.

96. Karras, T., S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8110–8119, 2020.

97. Irvin, J., P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K.

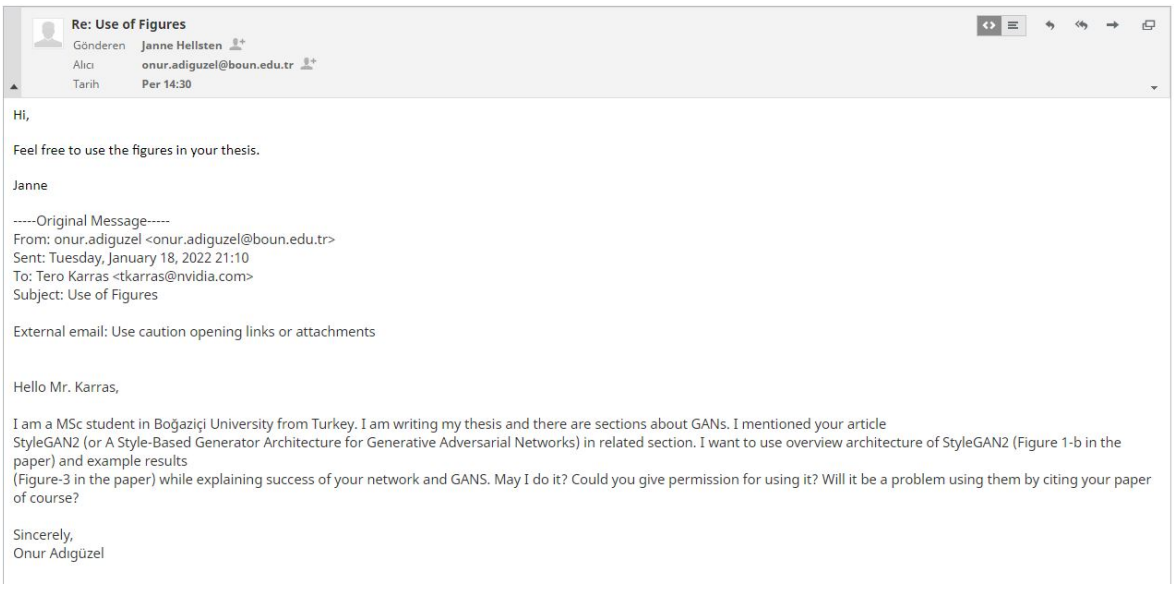
Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren and A. Y. Ng, "CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 590–597, 2019.

98. Peng, Y., X. Wang, L. Lu, M. Bagheri, R. Summers and Z. Lu, "NegBio: A High Performance Tool for Negation and Uncertainty Detection in Radiology Reports", *AMIA Summits on Translational Science Proceedings*, p. 188, 2017.

APPENDIX A: Permission for Copyright of Visuals Used In Thesis

The images that emerged within the scope of this thesis work and whose copyrights were transferred to the publishing house, were used in the thesis book in accordance with the publication policy of the publisher, which is valid for the reuse of the texts and graphics produced by the author, on his own web page.

Because we have used some figures from other publications, we have asked permission from the copyright holders, who are the authors of the articles. Figures 3.3 and 3.4 are from the publication "A Style-Based Generator Architecture for Generative Adversarial Networks" [1] by Karras et al. To get permission, we sent an email to the author and he said, "Feel free to use the figures in your thesis". Our email and their response can be found in Figure A.1.



Re: Use of Figures

Gönderen **Janne Hellsten**
Alıcı **onur.adiguzel@boun.edu.tr**
Tarih **Per 14:30**

Hi,

Feel free to use the figures in your thesis.

Janne

-----Original Message-----
From: onur.adiguzel <onur.adiguzel@boun.edu.tr>
Sent: Tuesday, January 18, 2022 21:10
To: Tero Karras <tkarras@nvidia.com>
Subject: Use of Figures

External email: Use caution opening links or attachments

Hello Mr. Karras,

I am a MSc student in Boğaziçi University from Turkey. I am writing my thesis and there are sections about GANs. I mentioned your article StyleGAN2 (or A Style-Based Generator Architecture for Generative Adversarial Networks) in related section. I want to use overview architecture of StyleGAN2 (Figure 1-b in the paper) and example results (Figure-3 in the paper) while explaining success of your network and GANS. May I do it? Could you give permission for using it? Will it be a problem using them by citing your paper of course?

Sincerely,
Onur Adıgüzel

Figure A.1. Permission to use visuals from official publication of StyleGAN2 [1].