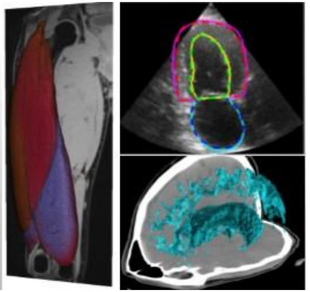
Generative Adversarial Networks (GANs)

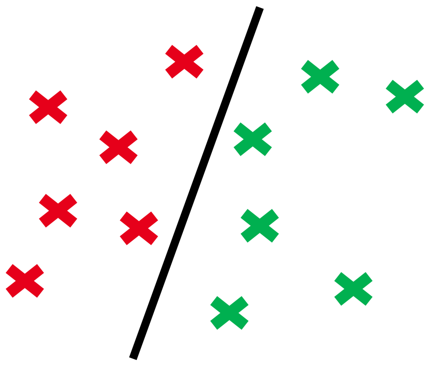
Anirban Mukhopadhyay

TU Darmstadt, Germany



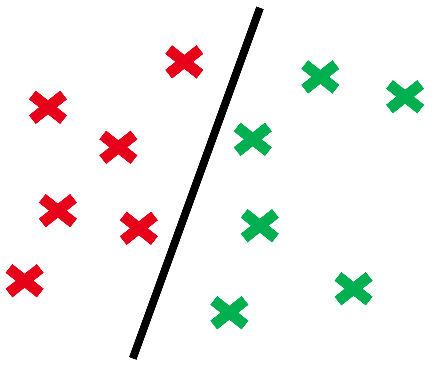
Introduction

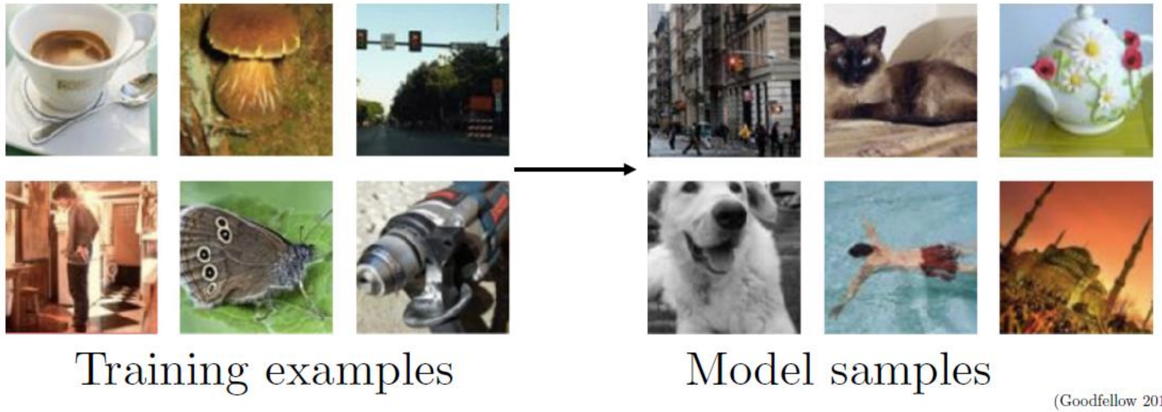
- Generative vs. Discriminative



Introduction

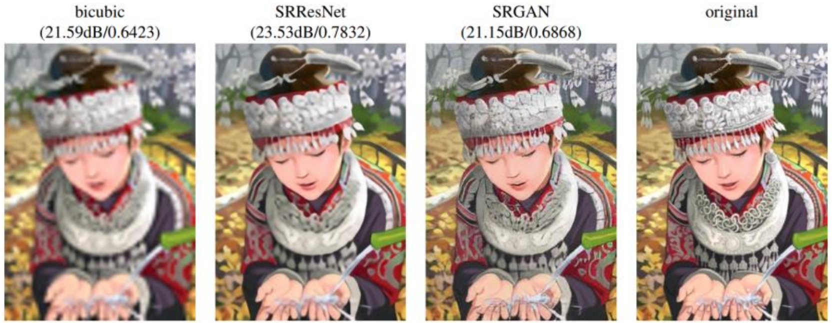
- Generative vs. Discriminative
  - Generating “realistic-looking” images – one step closer to understanding it





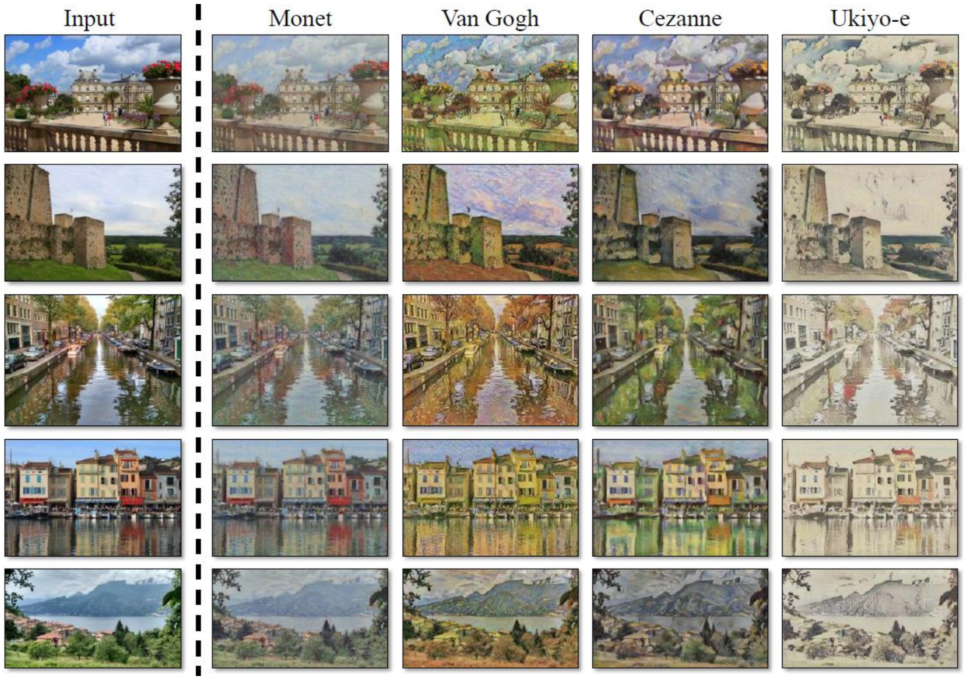
(Goodfellow 2016)

GAN Results



© SRGAN

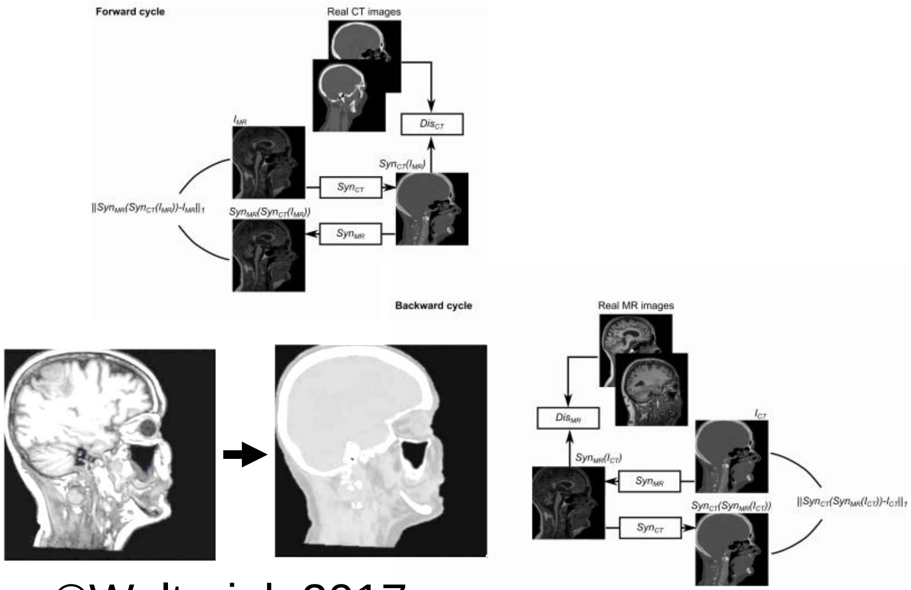
© Karras2018



© CycleGAN

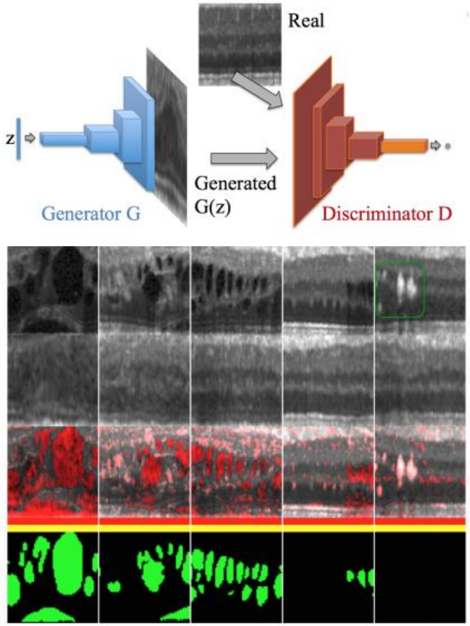
What is in it for me?

- MR to CT Reconstruction



©Wolterink 2017

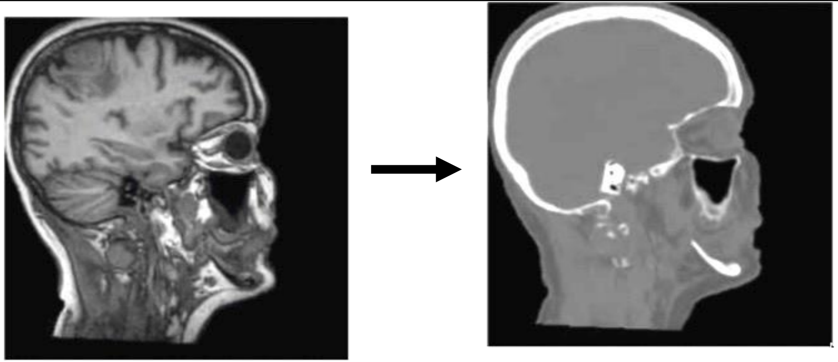
- Anomaly Detection



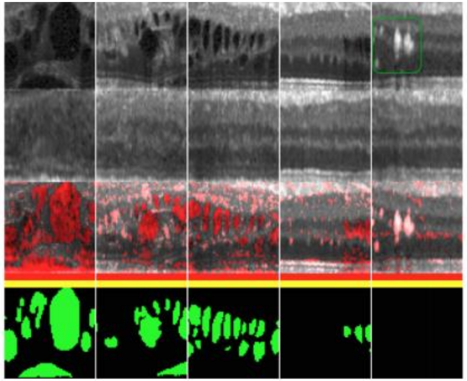
©Schlegl 2017

What is in it for me?

- Proxy for training data
  - Costly annotation
  - Imbalance



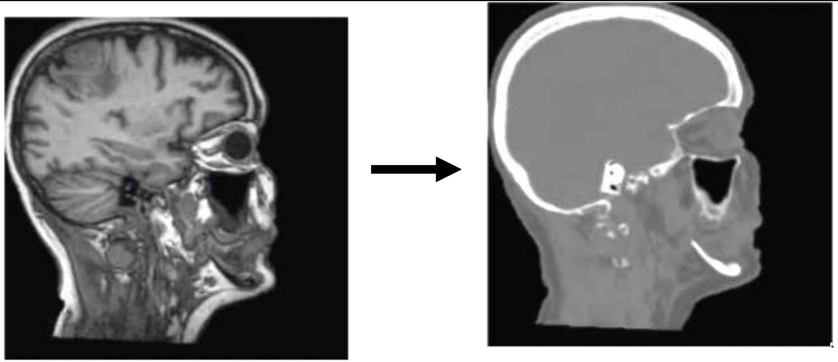
©Wolterink 2017



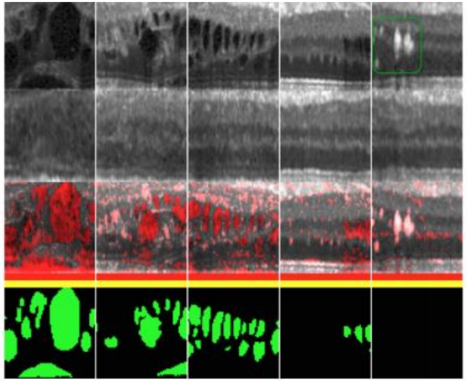
©Schlegl 2017

What is in it for me?

- Proxy for training data
  - Costly annotation
  - Imbalance
- Similarity metric
  - Discriminator



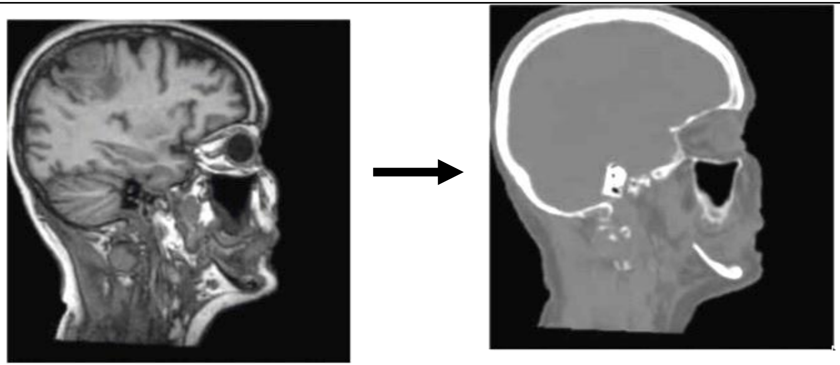
©Wolterink 2017



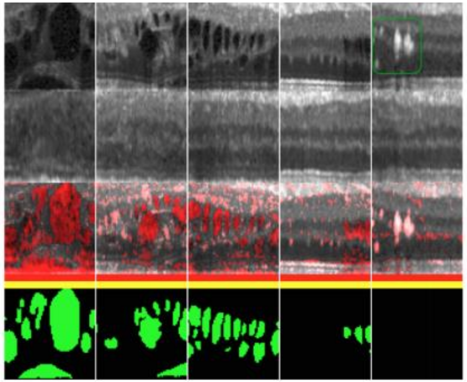
©Schlegl 2017

What is in it for me?

- Proxy for training data
  - Costly annotation
  - Imbalance
- Similarity metric
  - Discriminator
- Domain Shift
  - Adversarial training



©Wolterink 2017



©Schlegl 2017

Outline

- Theory
- Key GANs

- Medical Applications
- Adversarial Learning

- Limitations of GAN
- Summary

Tidying Up GAN – the Marie Kondo way

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

Closet

- Theory
- Key GANs

Kitchen

- Medical Applications
- Adversarial Learning

Emotional

- Limitations of GAN
- Summary

Tidying Up GAN – the Marie Kondo way

Theory

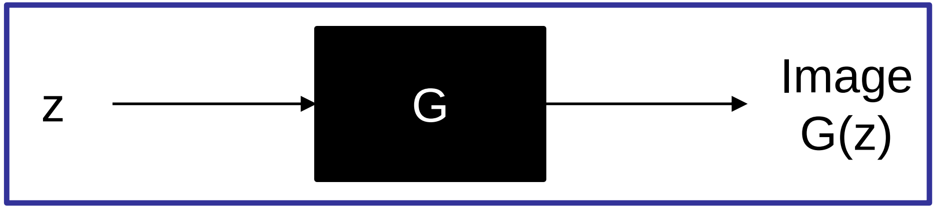
Theory

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- UNSUPERVISED Learning

Theory

- UNSUPERVISED Learning
- Perplexity
  - pdf for the generated distribution



Critique G – Calculating Perplexity

Theory

- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator
- **Idea 3:** Game of many moves

Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets

Similarity of P_{real} and P_{synth}

Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets

Similarity of P_{real} and P_{synth}
Deep Net D – maps images to [0,1]

Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets

Similarity of P_{real} and P_{synth}
Deep Net D – maps images to [0,1]
$E_x[D(x)]$ is high if $x \epsilon P_{real}$
$E_x[D(x)]$ is low if $x \epsilon P_{synth}$

Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets

Similarity of P_{real} and P_{synth}
Deep Net D – maps images to [0,1]
$E_x[D(x)]$ is high if $x \epsilon P_{real}$
$E_x[D(x)]$ is low if $x \epsilon P_{synth}$
Train using **Backpropagation**

Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator

Goal of generator G:
$E_z[D(G(z))]$ is as high as possible

Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator

Goal of generator G:
$E_z[D(G(z))]$ is as high as possible
Fooling Discriminator

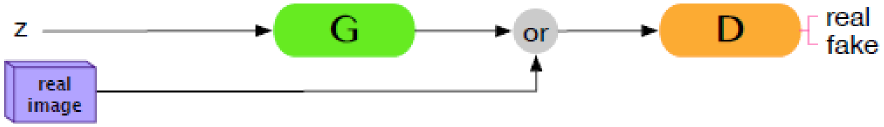
Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator

Goal of generator G:
$E_z[D(G(z))]$ is as high as possible
Fooling Discriminator
Backpropagation through D(G(.))

Theory

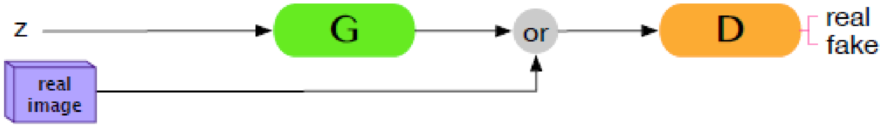
- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator
- **Idea 3:** Game of many moves



Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator
- **Idea 3:** Game of many moves

$$\min_G \max_D E_{x \sim P_{real}}[f(D(x))] + E_{z \sim P_{synth}}[1 - f(D(G(z)))]$$

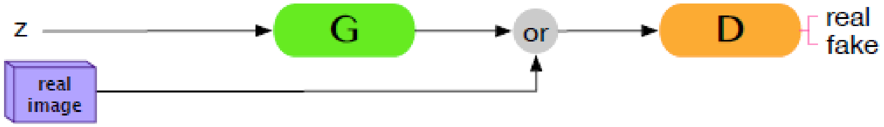


Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator
- **Idea 3:** Game of many moves

$$\min_G \max_D E_{x \sim P_{real}}[f(D(x))] + E_{z \sim P_{synth}}[1 - f(D(G(z)))]$$

For Goodfellow 2014
f(x) = log(x)

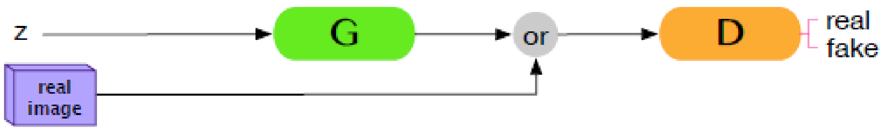


Theory

- Generative vs. Discriminative
- UNSUPERVISED Learning
- Perplexity
- **Idea 1:** Sidestep perplexity with deep nets
- **Idea 2:** Gradient feedback from discriminator
- **Idea 3:** Game of many moves

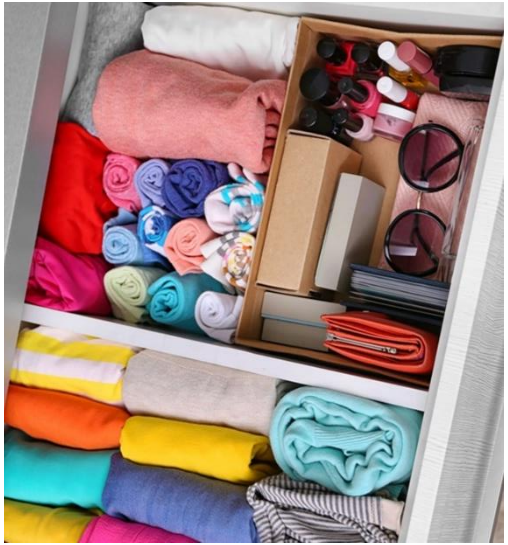
$$\min_G \max_D E_{x \sim P_{real}}[f(D(x))] + E_{z \sim P_{synth}}[1 - f(D(G(z)))]$$

For Goodfellow 2014
f(x) = log(x)
Derivative of log(x) = 1/x
Training **sensitive** to instances
that D finds **awful**



Understanding Key GANs

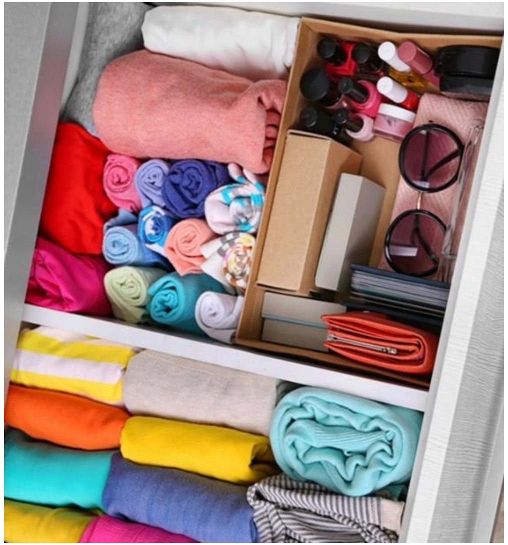
Theory | Engineering Recipe | Tidy GANs

© giphy.com

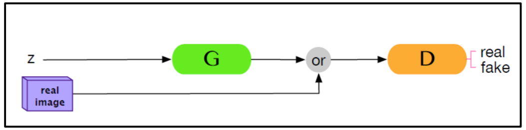
Understanding Key GANs

- Engineering Recipe
  - I/P, O/P
  - Architecture
  - Loss Function

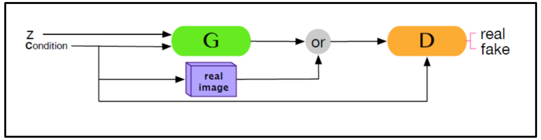


Understanding Key GANs

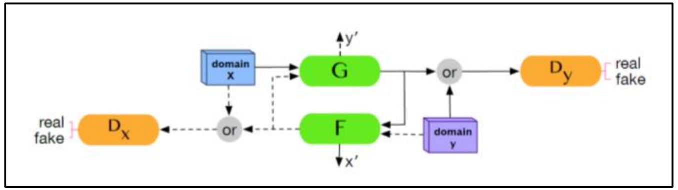
- Engineering Recipe
  - I/P, O/P
  - Architecture
  - Loss Function



DC-GAN



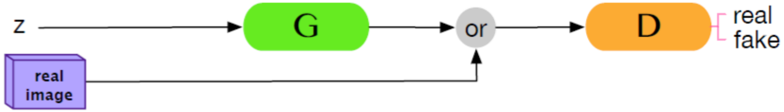
C-GAN



Cycle-GAN

Deep Convolutional GAN (DC-GAN)

- **Unsupervised**
- **Representation Learning**



Deep Convolutional GAN (DC-GAN)

- Unsupervised
- Representation Learning
- Latent space Interpolation

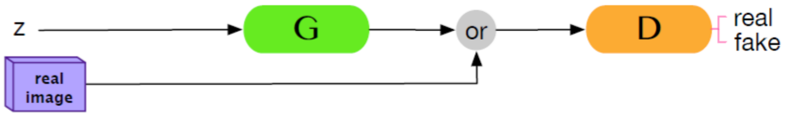
Deep Convolutional GAN

- I/P: Z (100-D multivariate Gaussian)
- O/P: Image

DC-GAN

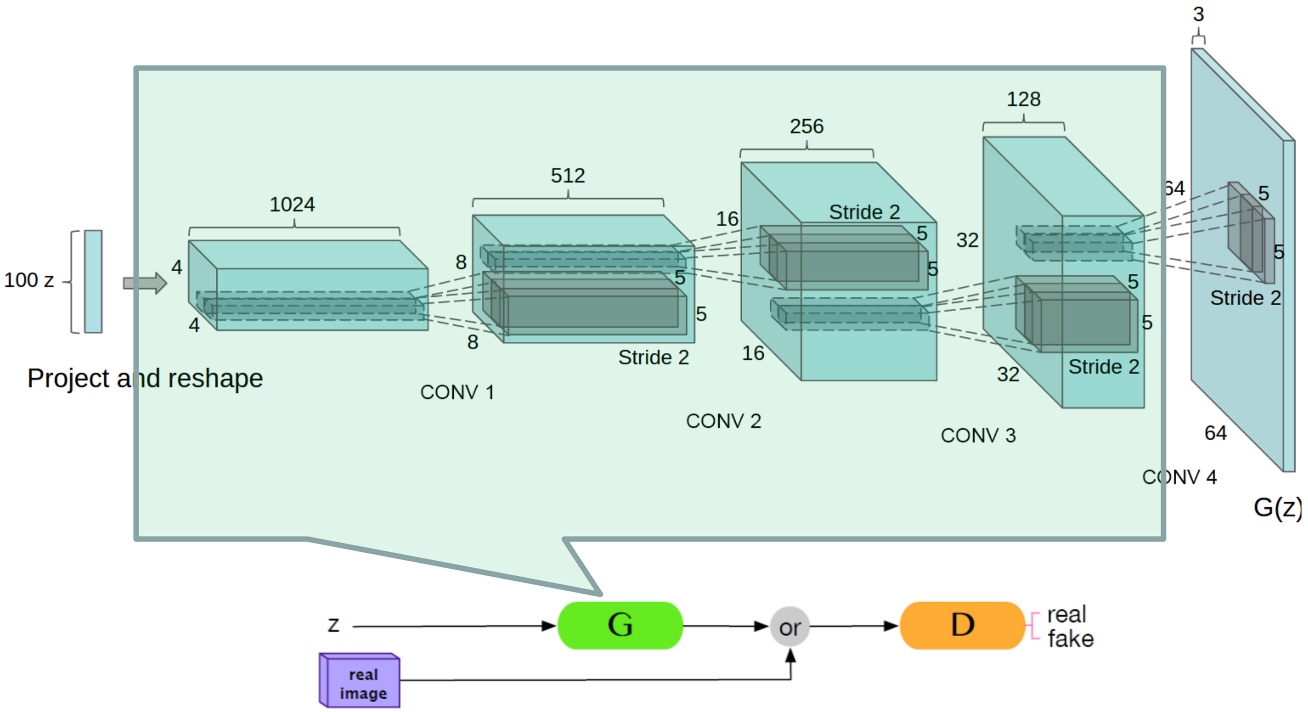
- I/P: Z (100-D multivariate Gaussian)
- O/P: Image

- Architecture:



DC-GAN

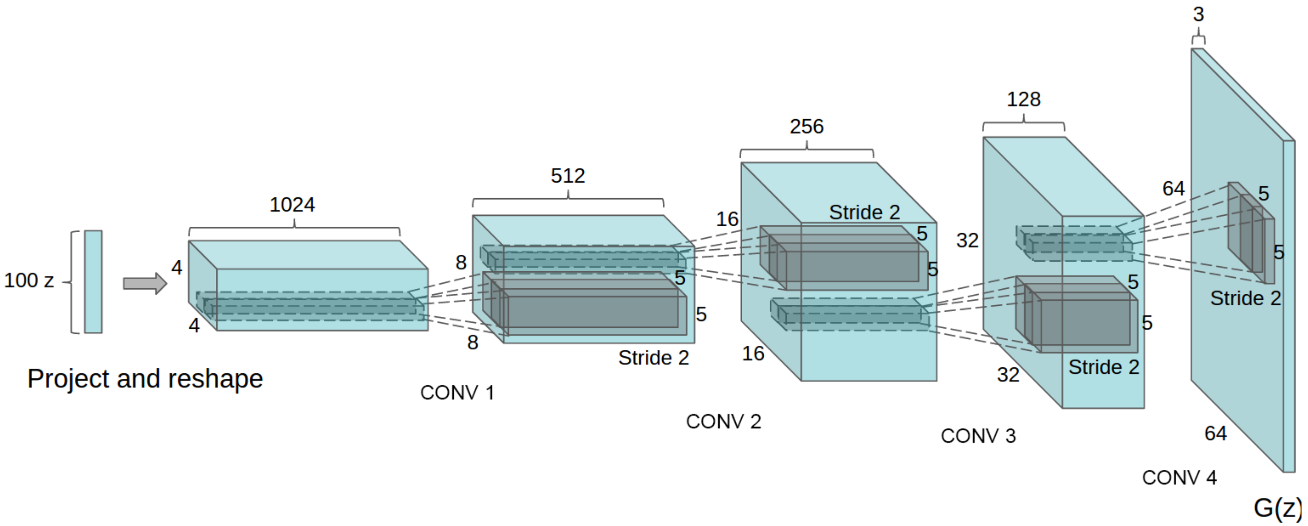
- **I/P: Z**
- **O/P: Image**
- **Architecture:**



DC-GAN

- **I/P: Z**
- **O/P: Image**

- **Architecture:**

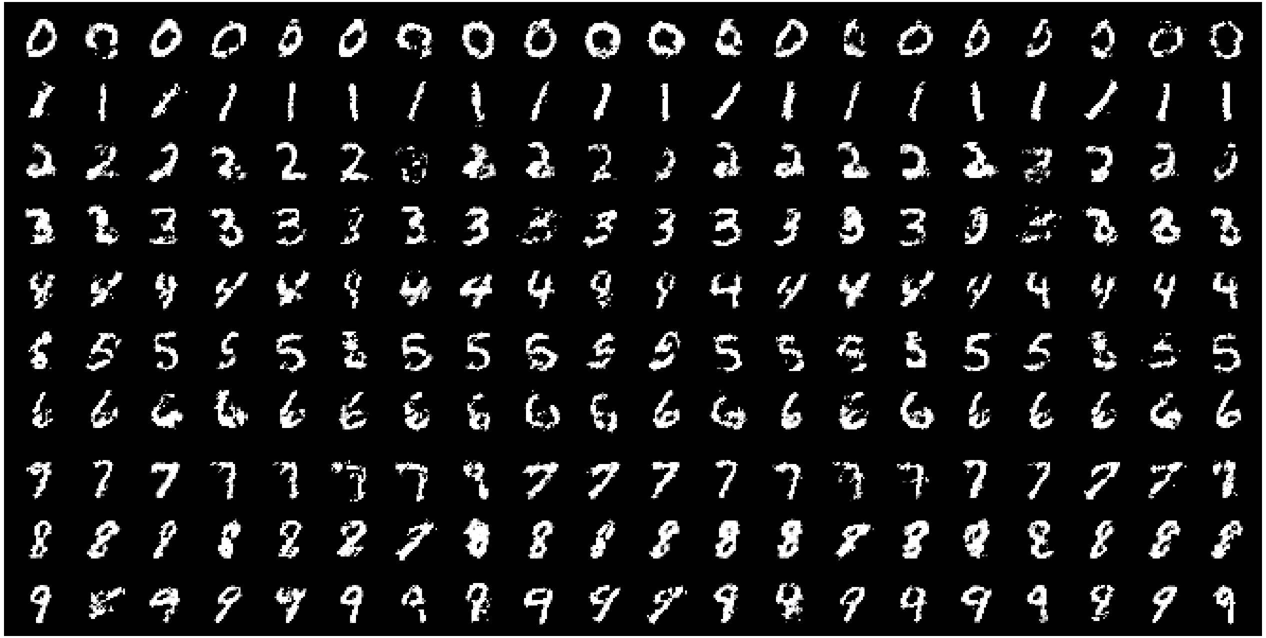


- **Loss Function:** Same as Goodfellow 2014

Conditional GAN (C-GAN)

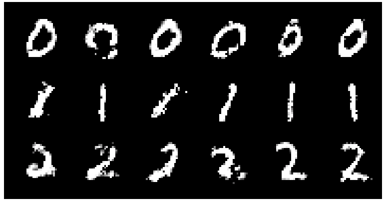
- How to bring in some supervision?

0
1
2
3
4
5
6
7
8
9



Conditional GAN (C-GAN)

- **I/P: Z, Condition** 0,1,2,…
- **O/P: Image**

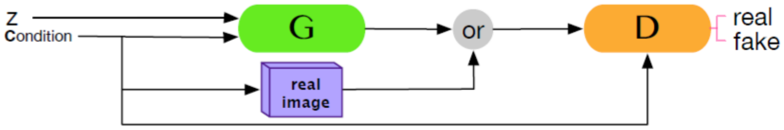


C-GAN

TECHNISCHE UNIVERSITÄT DARMSTADT

- **I/P: Z, Condition**
- **O/P: Image**

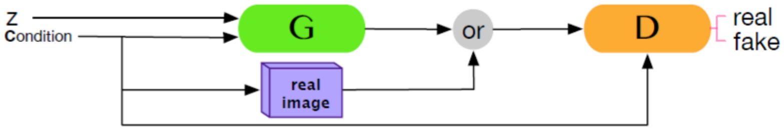
- **Architecture:**



C-GAN

- **I/P: Z, Condition(c)**
- **O/P: Image**

- **Architecture:**

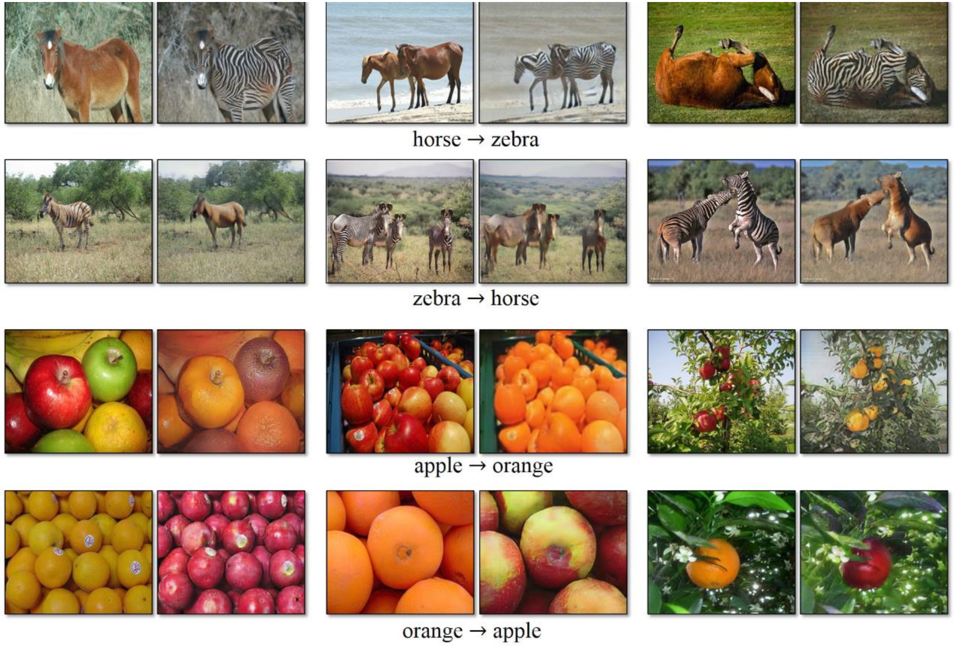


- **Loss Function:**

$$\min_G \max_D E_{x\sim P_{real}}[f(D(x|c))] + E_{z\sim P_{synth}}[1 - f(D(G(z|c)))]$$

Cycle-GAN

- How to incorporate unpaired images for style/ domain transfer?



©Cycle-GAN

Cycle-GAN

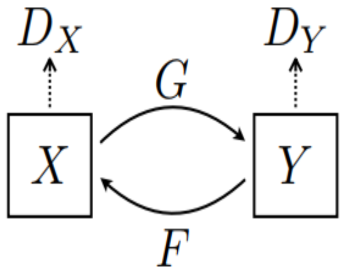
- **I/P: Image (Domain X)**
- **O/P: Image (Domain Y)**

UN-PAIRED

Cycle-GAN

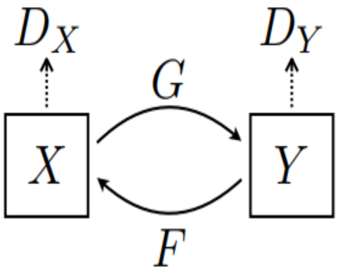
- I/P: Image (Domain X)
- O/P: Image (Domain Y)

- Architecture:

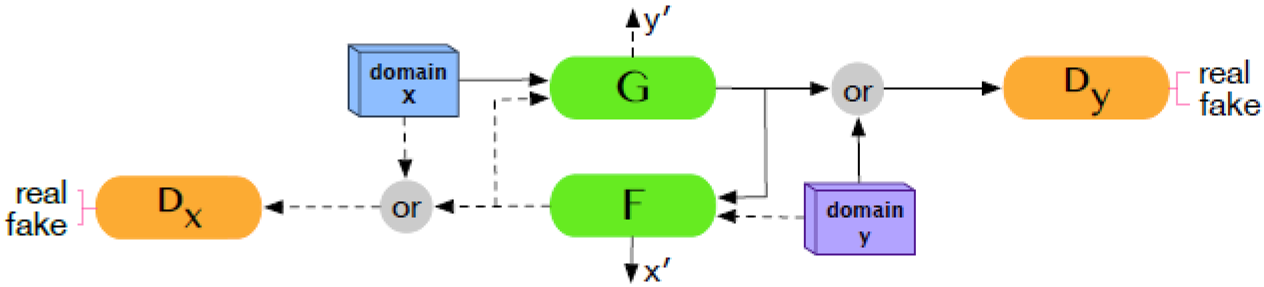


Cycle-GAN

- **I/P: Image (Domain X)**
- **O/P: Image (Domain Y)**

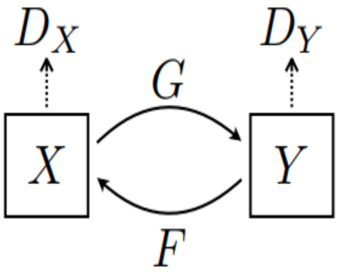


- **Architecture:**

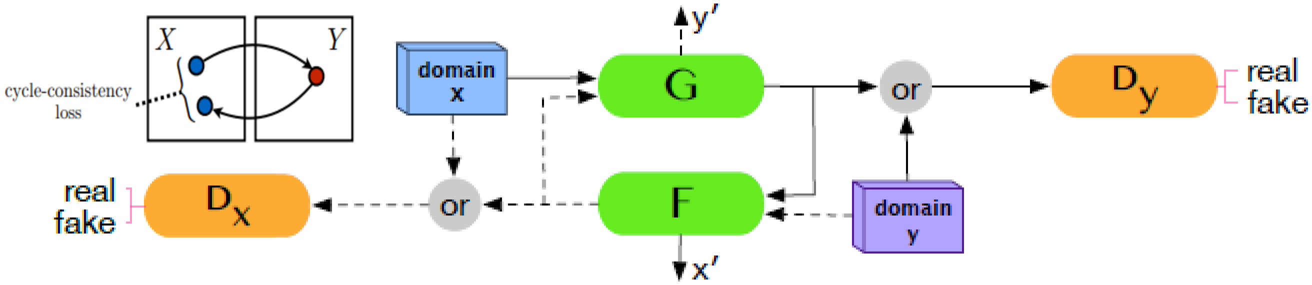


Cycle-GAN

- **I/P: Image (Domain X)**
- **O/P: Image (Domain Y)**



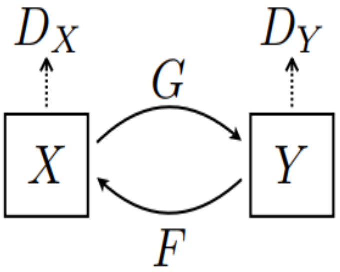
- **Architecture:**



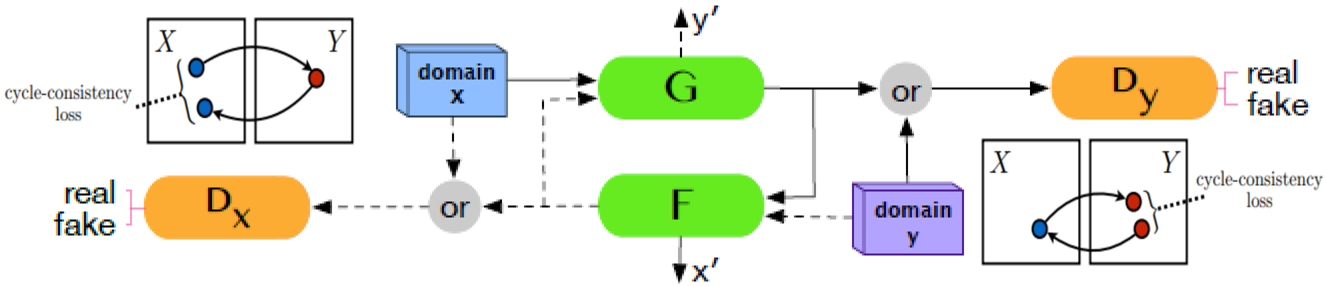
- **Loss Function**

Cycle-GAN

- **I/P: Image (Domain X)**
- **O/P: Image (Domain Y)**

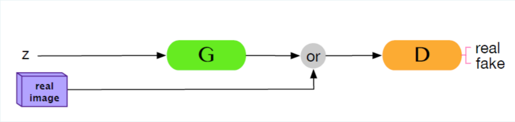
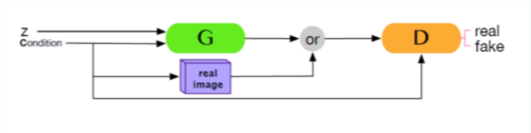
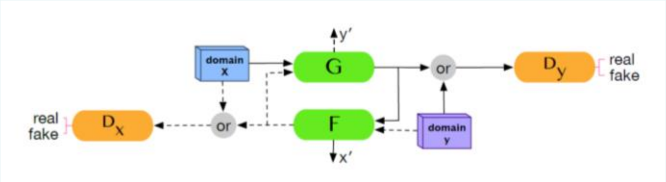


- **Architecture:**



- **Loss Function**

Engineering Recipe Summary

GANs	I/P	O/P	Architect.	Loss	Note
DC-GAN	z	Img		GAN	Unsup.
C-GAN	z,c	Img		Modif. GAN	Cond. Supervis.
Cycle-GAN	Img (X)	Img (Y)		Cycle Loss	Style Transfer

Medical Applications

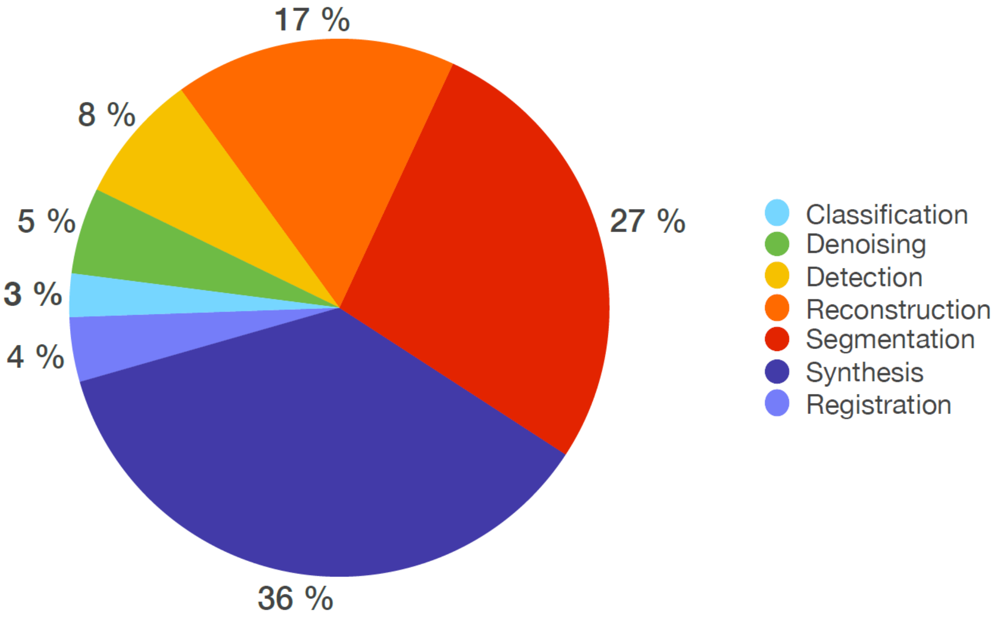
© popsugar

Medical Applications

- Review article
  - 77 papers are reviewed
  - Till end of 2018
  - Incl. MICCAI, MiDL, ISBI, TMI, MedIA etc.

Medical Applications

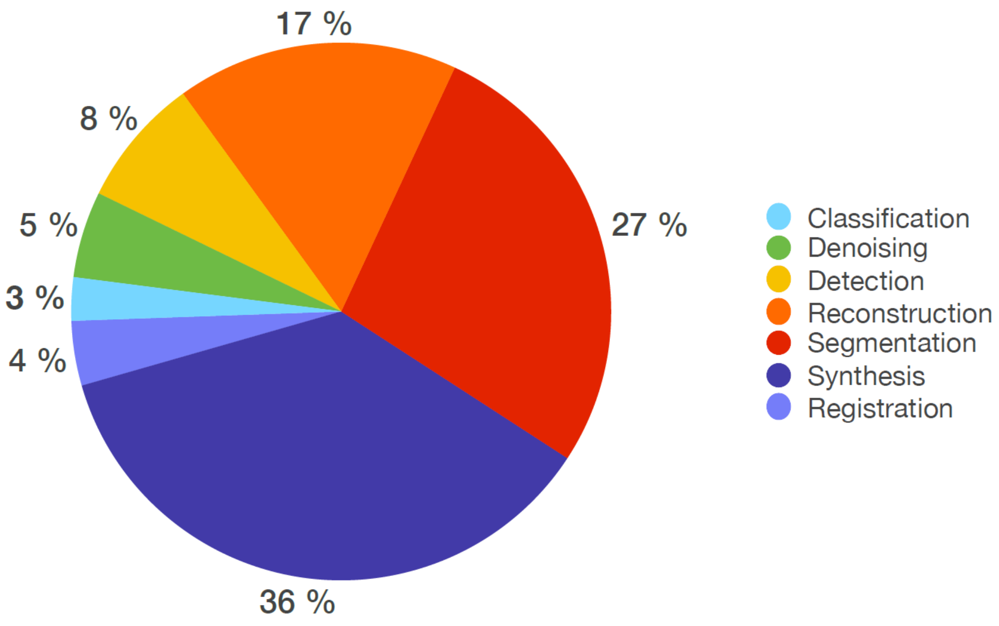
- Review article
  - 77 papers are reviewed
  - Till end of 2018
  - Incl. MICCAI, MiDL, ISBI, TMI, MedIA etc.
- Mostly applied in
  - Synthesis
  - Segmentation



https://arxiv.org/abs/1809.06222

Medical Applications

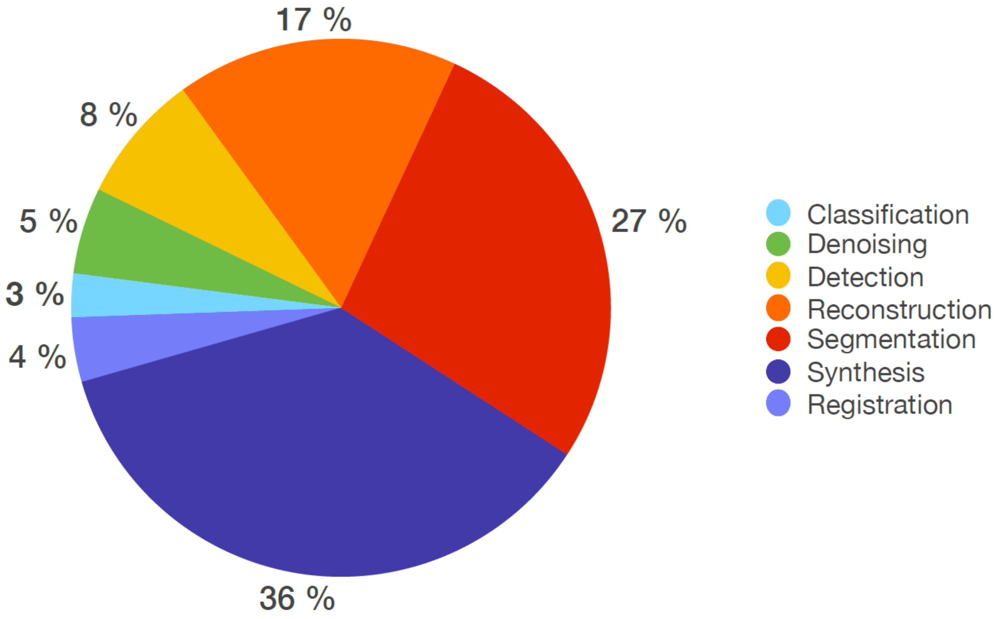
- Review article
  - 77 papers are reviewed
  - Till end of 2018
  - Incl. MICCAI, MiDL, ISBI, TMI, MedIA etc.
- Mostly applied in
  - Synthesis
  - Segmentation
- **Pattern**
  - Modify Architecture
  - Modify Loss



https://arxiv.org/abs/1809.06222

Medical Applications

- Review article
  - 77 papers are reviewed
  - Till end of 2018
  - Incl. MICCAI, MiDL, ISBI, TMI, MedIA etc.
- Mostly applied in
  - Synthesis
  - Segmentation
- **Pattern**
  - Modify Architecture
  - Modify Loss
- Re-apply the recipe



https://arxiv.org/abs/1809.06222

Synthesis - Unsupervised

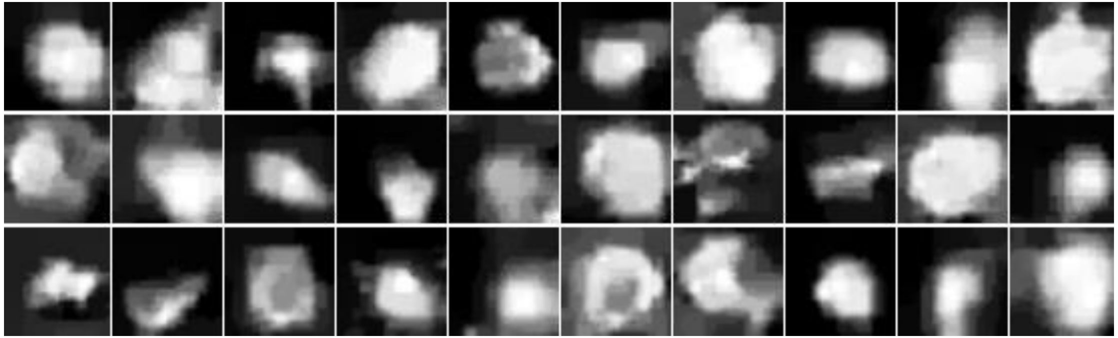
- Discriminating Lung Nodules
  - Benign
  - Malign

©Chuquicusma 2018

Synthesis - Unsupervised

- Discriminating Lung Nodules
  - Benign
  - Malign

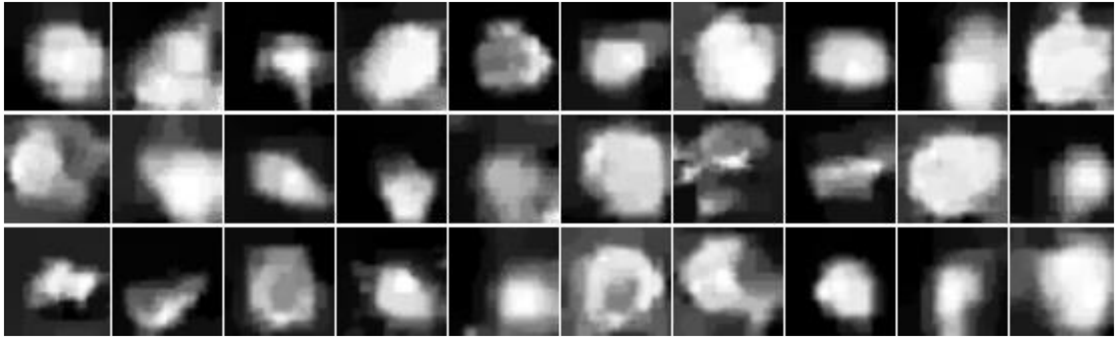
- Unsupervised synthesis
  - Modify DC-GAN



©Chuquicusma 2018

Synthesis - Unsupervised

- Discriminating Lung Nodules
  - Benign
  - Malign



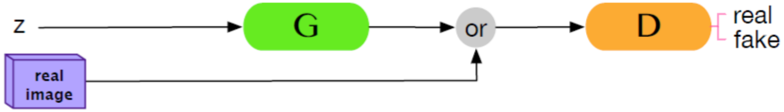
- Unsupervised synthesis
  - Modify DC-GAN
- Visual Turing Test
  - 2 radiologists

©Chuquicusma 2018

Synthesis - Unsupervised

- **I/P: Z**
- **O/P: Image (64X64X3)**

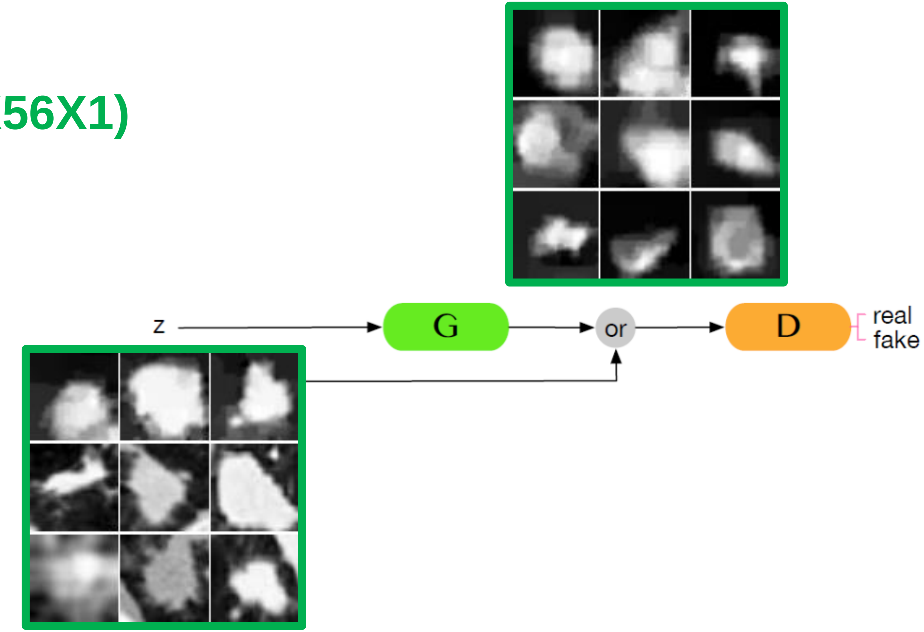
- **Architecture:**



- **Loss Function:** Same as Goodfellow 2014

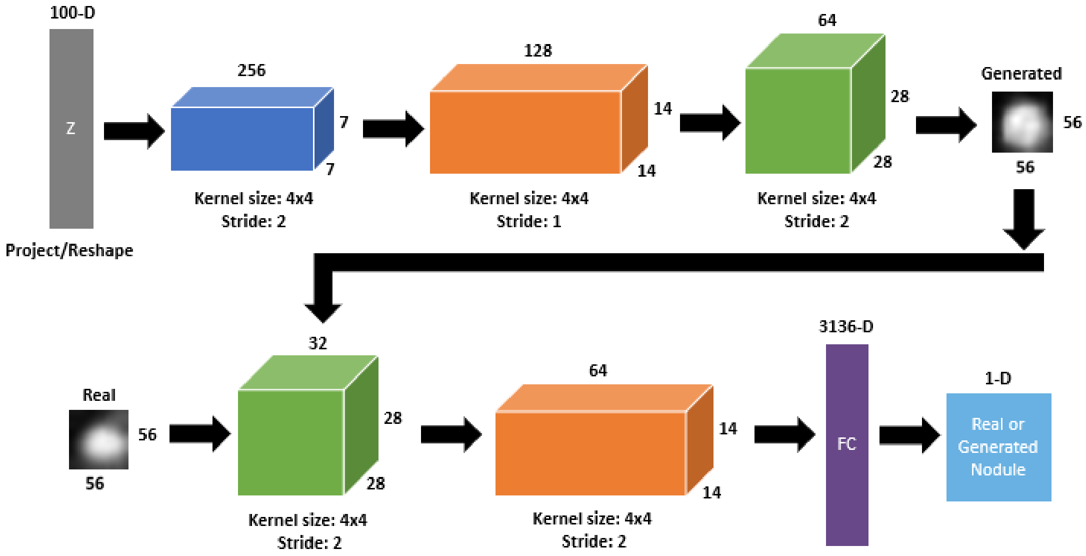
Synthesis - Unsupervised

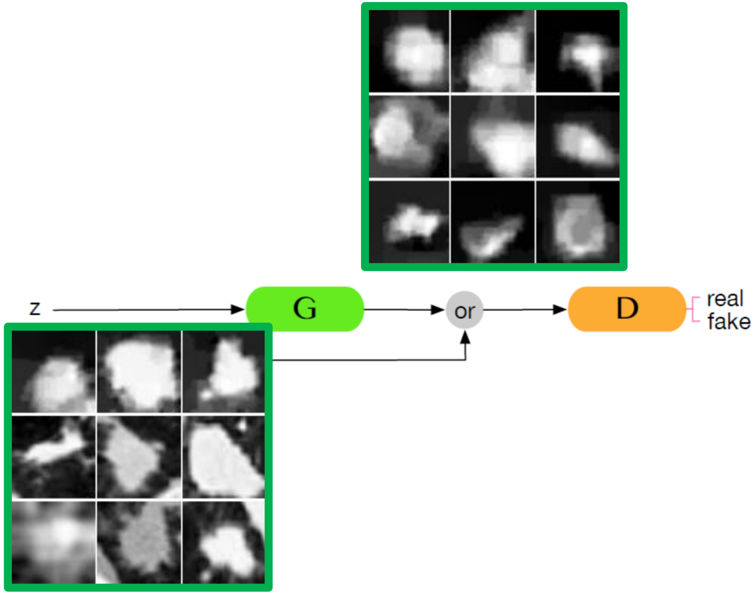
- **I/P: Z**
- **O/P: Lung Nodule image (56X56X1)**
- **Architecture:**



- **Loss Function:** Same as Goodfellow 2014

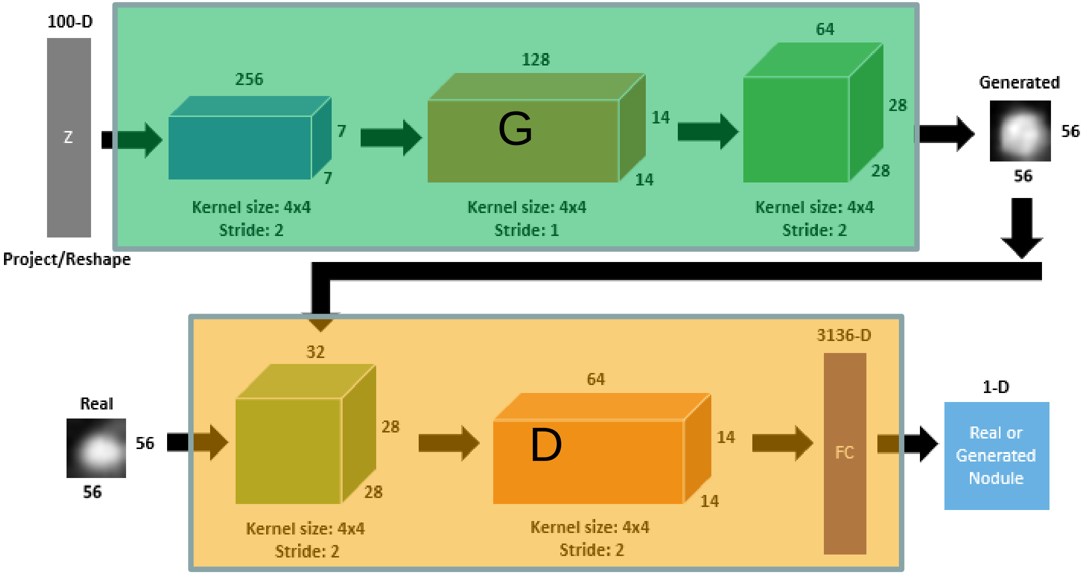
Synthesis - Unsupervised

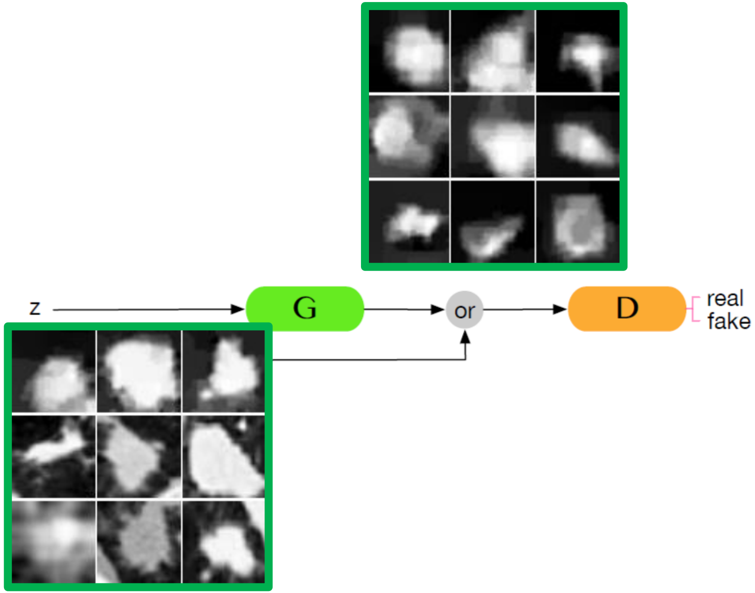




©Chuquicusma 2018

Synthesis - Unsupervised





©Chuquicusma 2018

Synthesis - Supervised

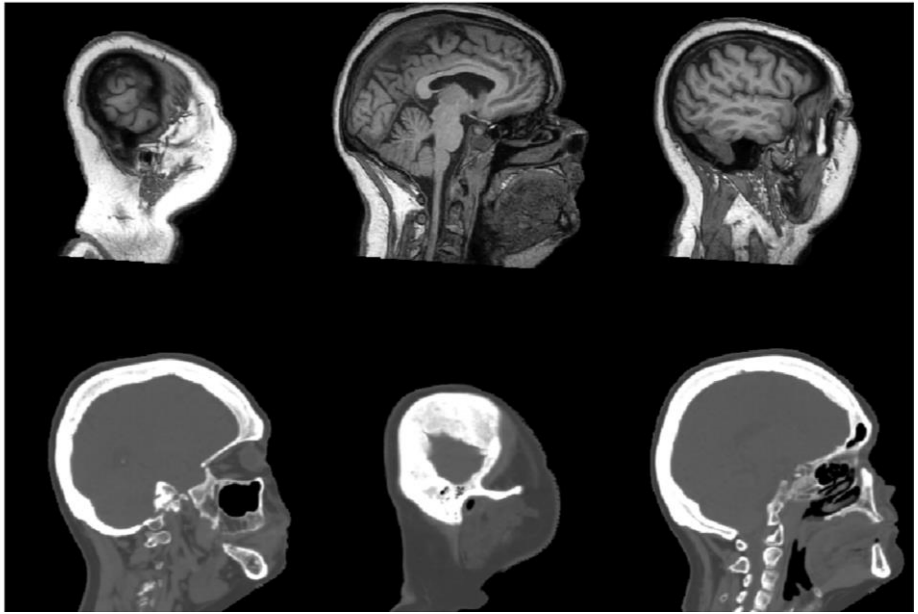
- Radiotherapy treatment planning
  - MR: Segmentation of tumor and organs
  - CT: Dose planning

©Wolterink 2017

Synthesis - Supervised

- Radiotherapy treatment planning
  - MR: Segmentation of tumor and organs
  - CT: Dose planning
- MR-only radiotherapy treatment planning
  - Synthesize CT

Unpaired data

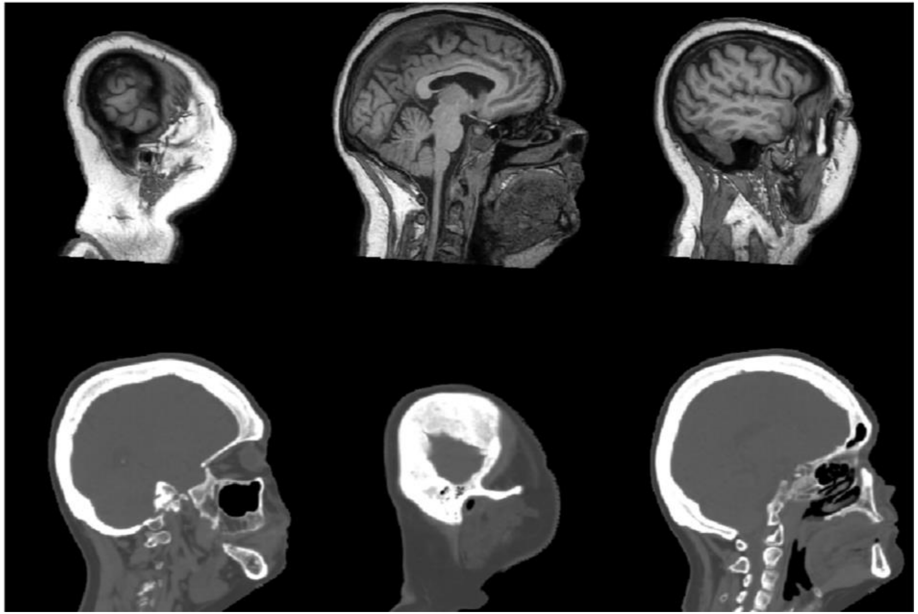


©Wolterink 2017

Synthesis - Supervised

- Radiotherapy treatment planning
  - MR: Segmentation of tumor and organs
  - CT: Dose planning
- MR-only radiotherapy treatment planning
  - Synthesize CT
- Re-purpose Cycle-GAN

Unpaired data

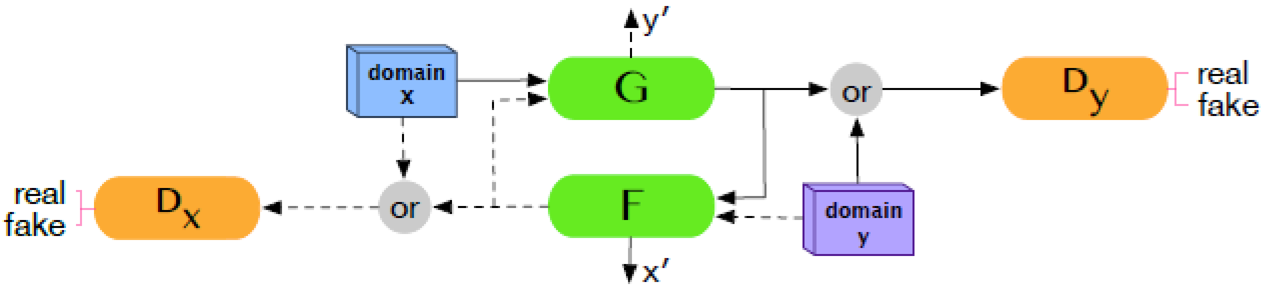


©Wolterink 2017

Synthesis - Supervised

TECHNISCHE UNIVERSITÄT DARMSTADT

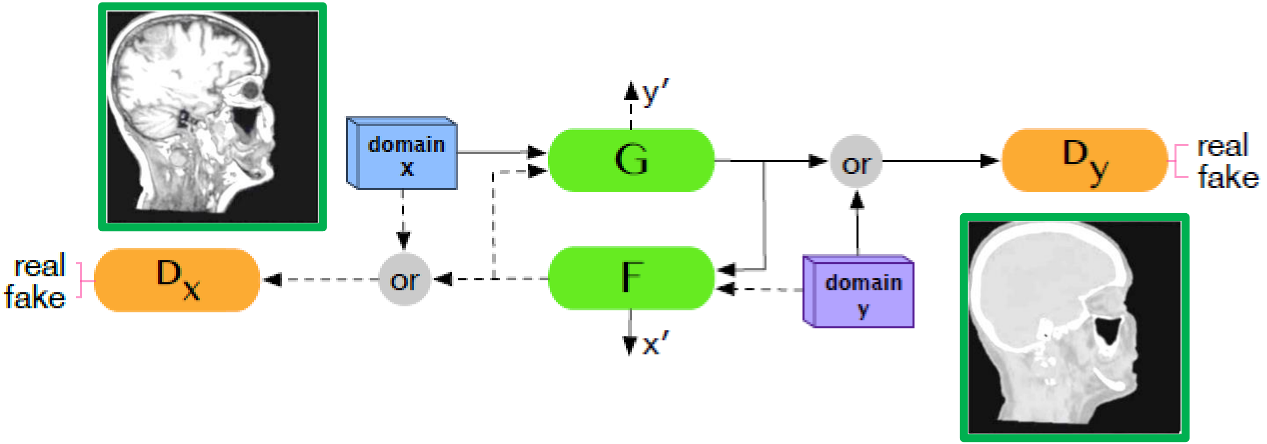
- **I/P: Image (Domain X)**
- **O/P: Image (Domain Y)**
- **Architecture:**



- **Loss Function: Cycle Loss**

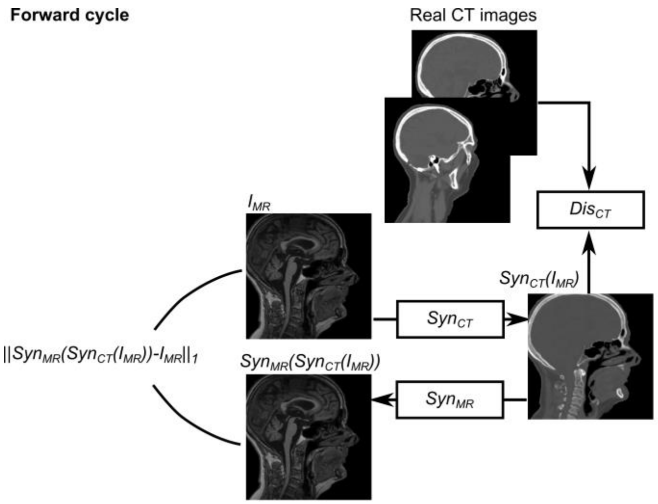
Synthesis - Supervised

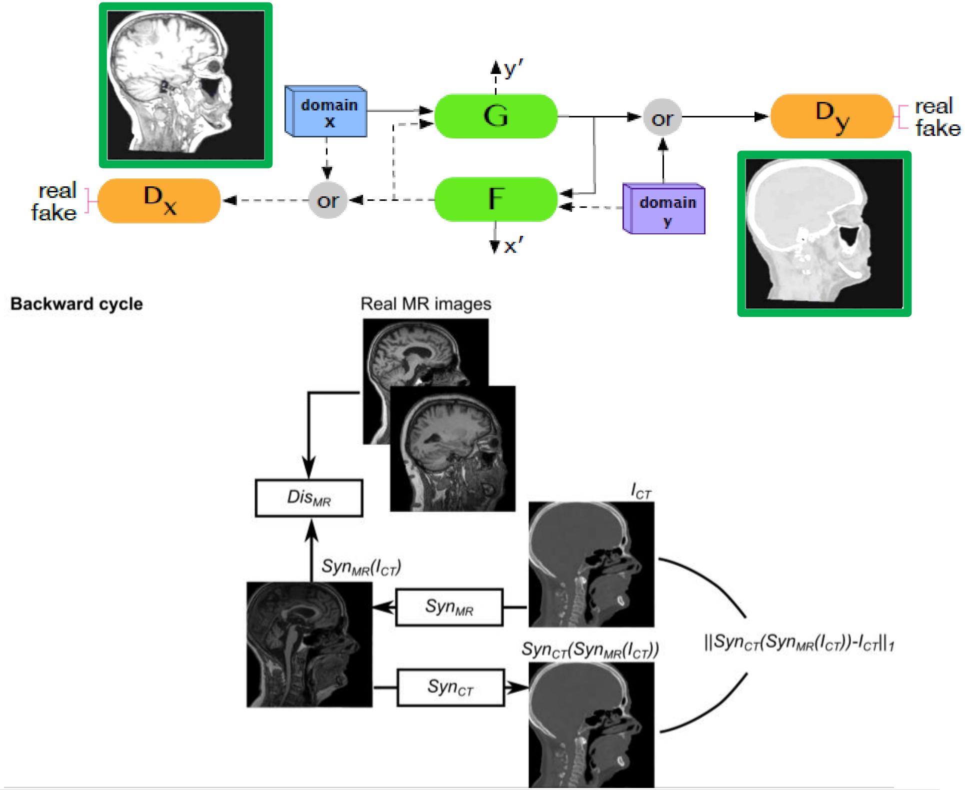
- **I/P: MR**
- **O/P: CT**
- **Architecture:**



- **Loss Function: Cycle Loss (Sum of L1 norms at MR and CT)**

Synthesis - Supervised





©Wolterink 2017

Domain Adaptation - Adversarial Learning

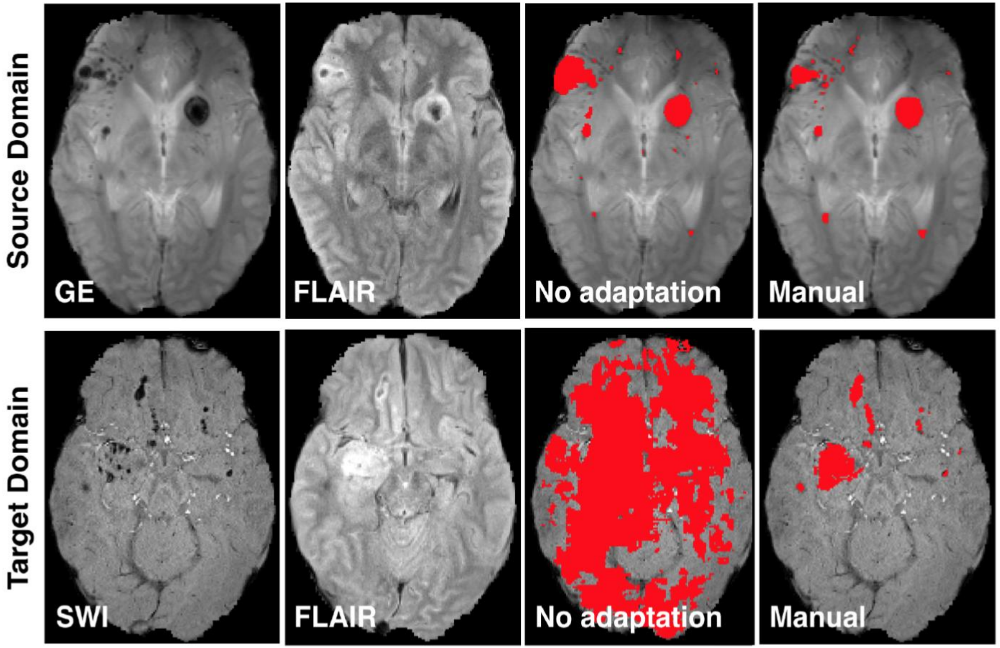
- Deep Learning Segmentation
  - Performs well in same domain
  - Degrades with new domain

©Kamnitsas 2017

Domain Adaptation - Adversarial Learning

- Deep Learning Segmentation
  - Performs well in same domain
  - Degrades with new domain

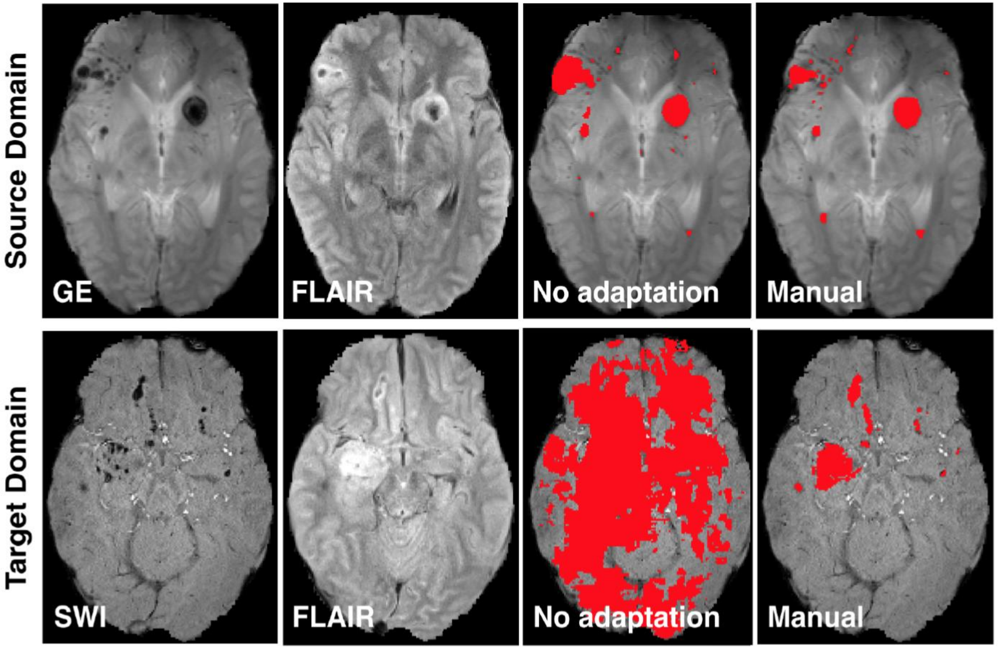
- Traumatic Brain Injury
  - Segment bleeding



©Kamnitsas 2017

Domain Adaptation - Adversarial Learning

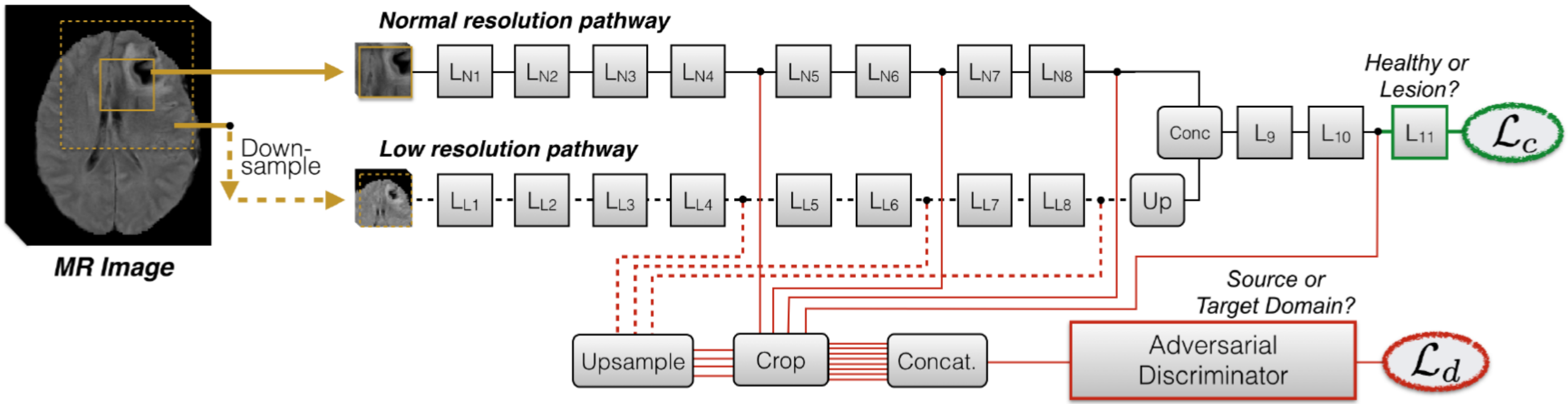
- Deep Learning Segmentation
  - Performs well in same domain
  - Degrades with new domain
- Traumatic Brain Injury
  - Segment bleeding
- Learn domain invariant features
  - Auxiliary task - Adversarial



©Kamnitsas 2017

Domain Adaptation - Adversarial Learning

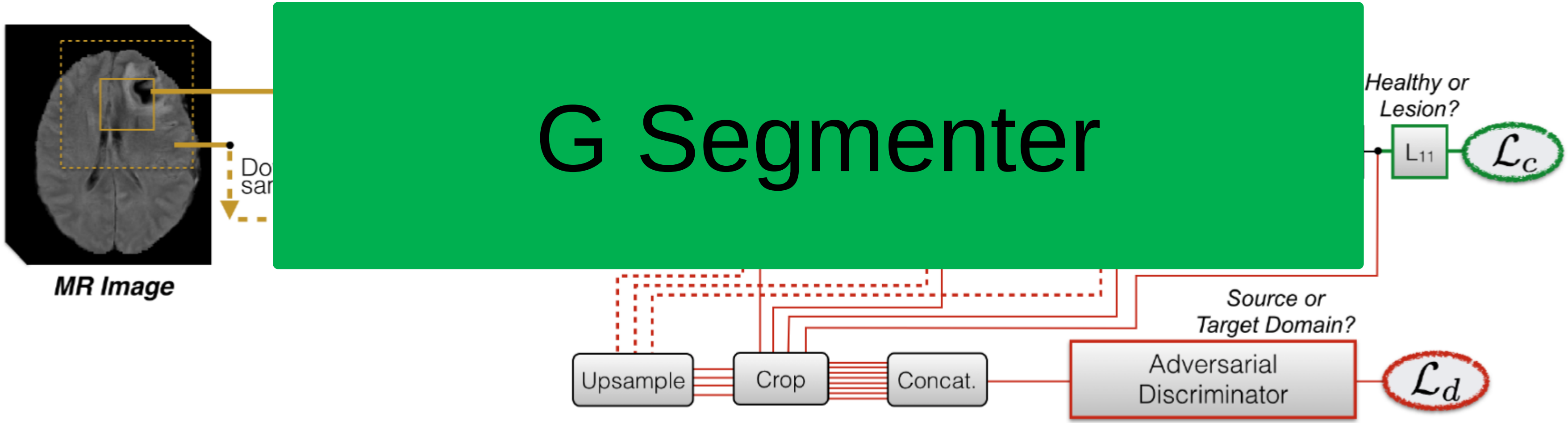
- Different MR sequences



©Kamnitsas 2017

Domain Adaptation - Adversarial Learning

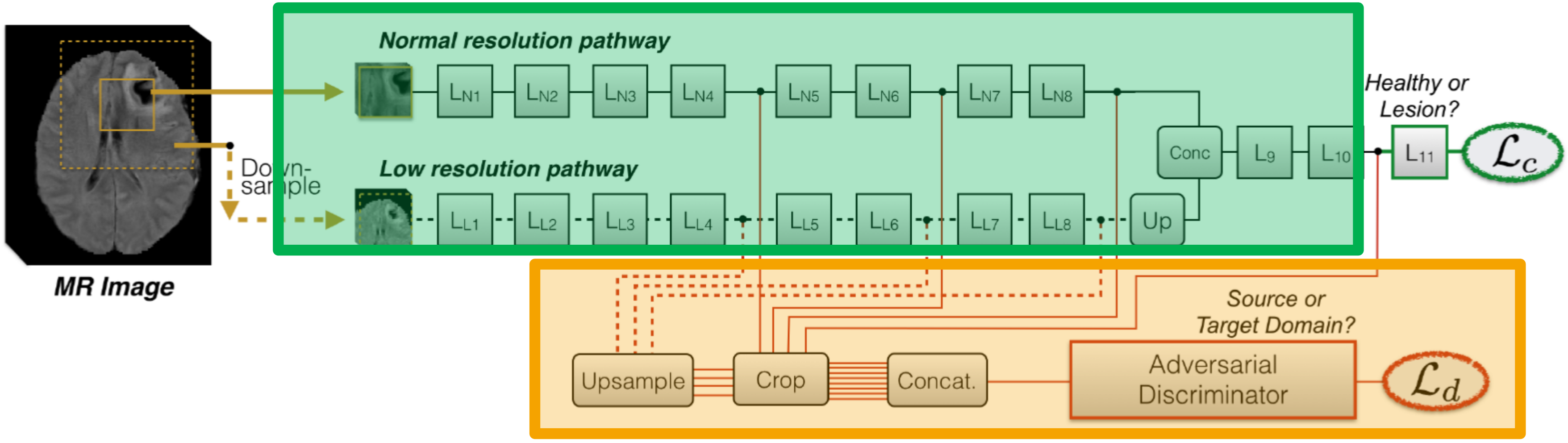
- Different MR sequences



©Kamnitsas 2017

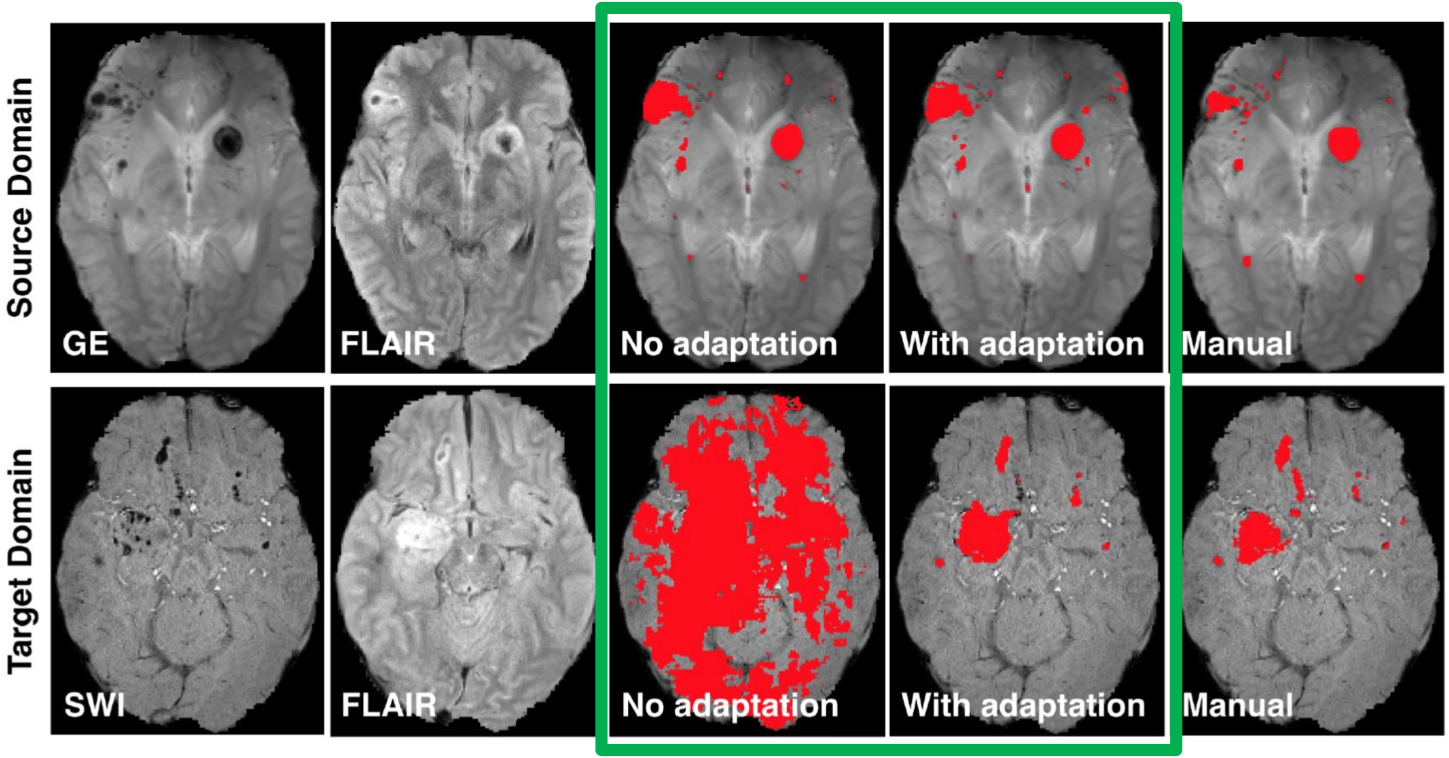
Domain Adaptation - Adversarial Learning

- Different MR sequences



©Kamnitsas 2017

Domain Adaptation - Adversarial Learning



©Kamnitsas 2017

Limitations of GAN

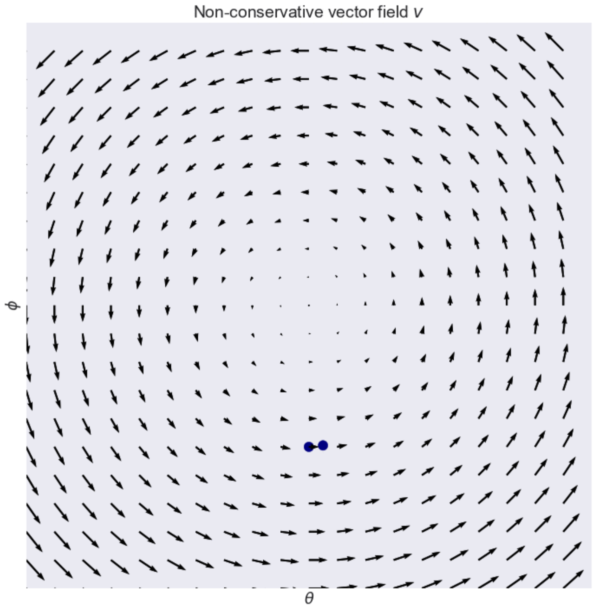
© slate.com

Limitations of GAN

- Numerical Instability
  - Mode Collapse

Extreme example

- Constant curl vector
  - Non-conservative
- Arises **naturally in zero-sum game**
  - Follow arrow like simultaneous gradient ascent
  - Though has equilibrium at (0,0)
- Initial Solution
  - Numerics of GAN (Reading List)



© inFERENCe

Limitations of GAN

- Numerical Instability
  - Mode Collapse

- Evaluation
  - Metrics

Limitations of GAN - Practical

Counting

Medical Equivalent
Cell Images

Limitations of GAN - Practical

Counting

Medical Equivalent
Cell Images

Perspective

Medical Equivalent
Cross domain synthesis

Limitations of GAN - Practical

Counting

Medical Equivalent
Cell Images

Perspective

Medical Equivalent
Cross domain synthesis

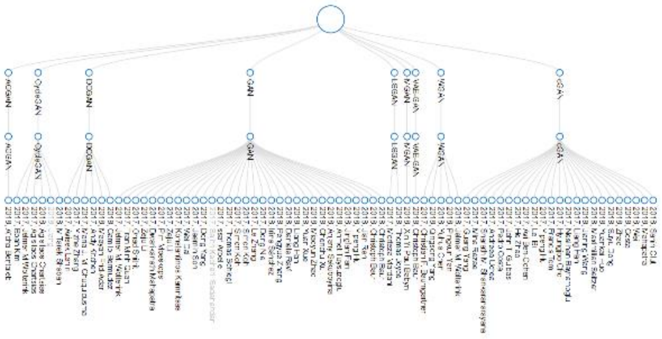
Global Structure

Medical Equivalent
Reconstruction

Reading List

Review: **GANs for Medical Image Analysis**

- Engineering
  - DC-GAN
  - C-GAN
  - CycleGAN
- GAN applications
  - Living Review



  - Wolterink 2017
  - Kamnitsas 2017
  - Chuquicusma 2018

- Theory
  - Numerics of GANs
  - Are GANs Created Equal?
  - f-GANs
- Blogs
  - Off the convex Path
  - GAN Open Problems
- MICCAI 2019 Tutorial
  - Lecturers: Me, J. Wolterink, K. Kamnitsas

Summary

- GANs – Unsupervised generative models with adversarial twist
- When done correctly
  - Realistic-looking images of unprecedented quality

- Medical Imaging
  - Synthesis - proxy for training data
  - Domain shift

- Issues
  - Numerical Instability
  - Evaluation metric

https://arxiv.org/abs/1809.06222

GANs for Medical Image Analysis

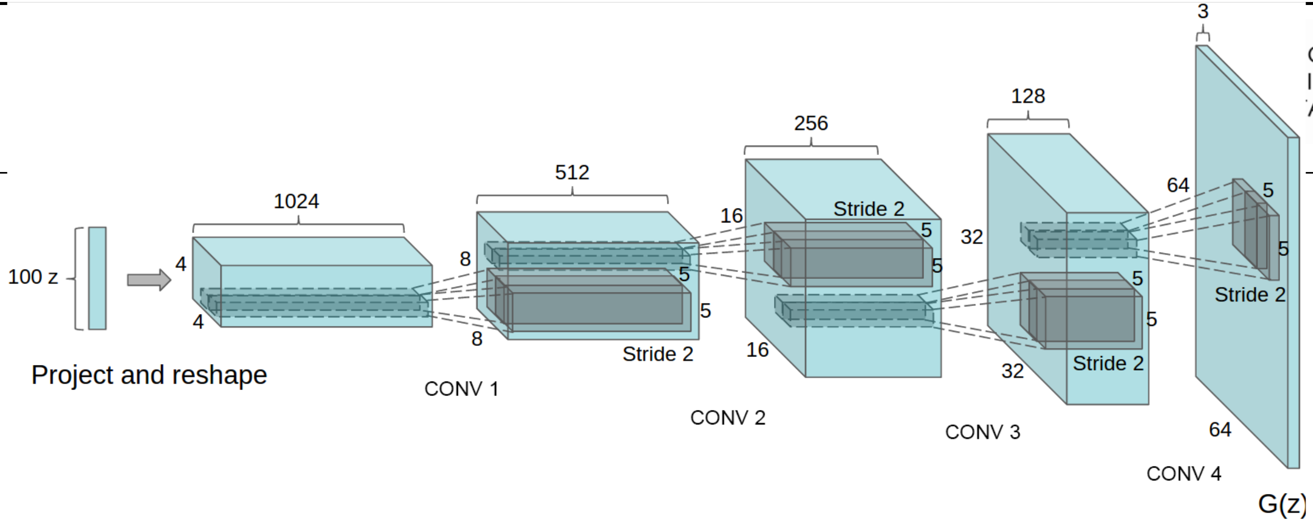
Salome Kazeminia[a,1], Christoph Baur[b,1], Arjan Kuijper[c], Bram van Ginneken[d], Nassir Navab[b], Shadi Albarqouni[b], Anirban Mukhopadhyay[a]

[a]*Department of Computer Science, TU Darmstadt, Germany*
[b]*Computer Aided Medical Procedures (CAMP), TU Munich, Germany*
[c]*Fraunhofer IGD, Darmstadt, Germany*
[d]*Radboud University Medical Center, Nijmegen, The Netherlands*

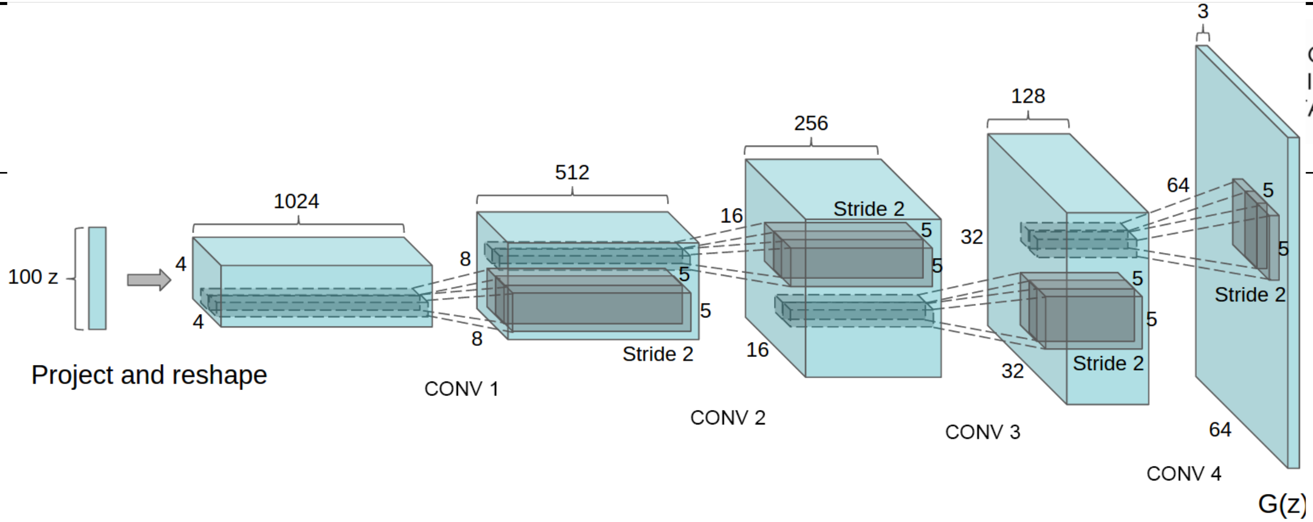
Thank You!

Backup Slides

DC-GAN


100 z
Project and reshape
1024
4
4
CONV 1
512
8
8
5
5
Stride 2
CONV 2
256
16
16
Stride 2
5
5
CONV 3
128
32
32
5
5
Stride 2
CONV 4
3
64
64
5
5
Stride 2
G(z)

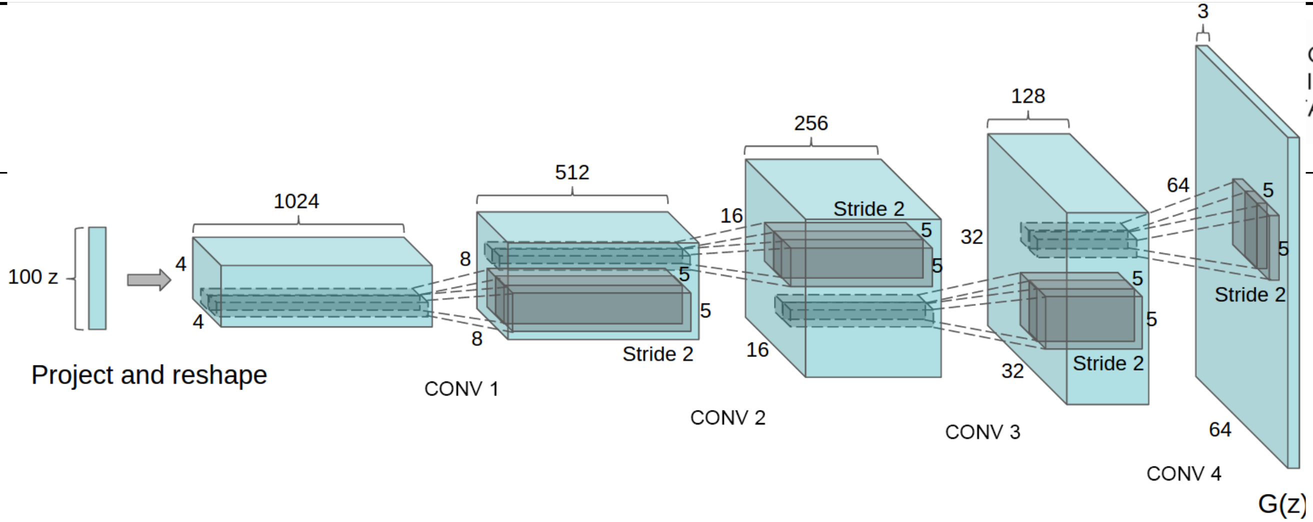
DC-GAN



Recipe

- Replace pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).

DC-GAN



Recipe

- Replace pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm
- Use LeakyReLU in discriminator

Synthesis

Unconditnl. (DC-GAN)

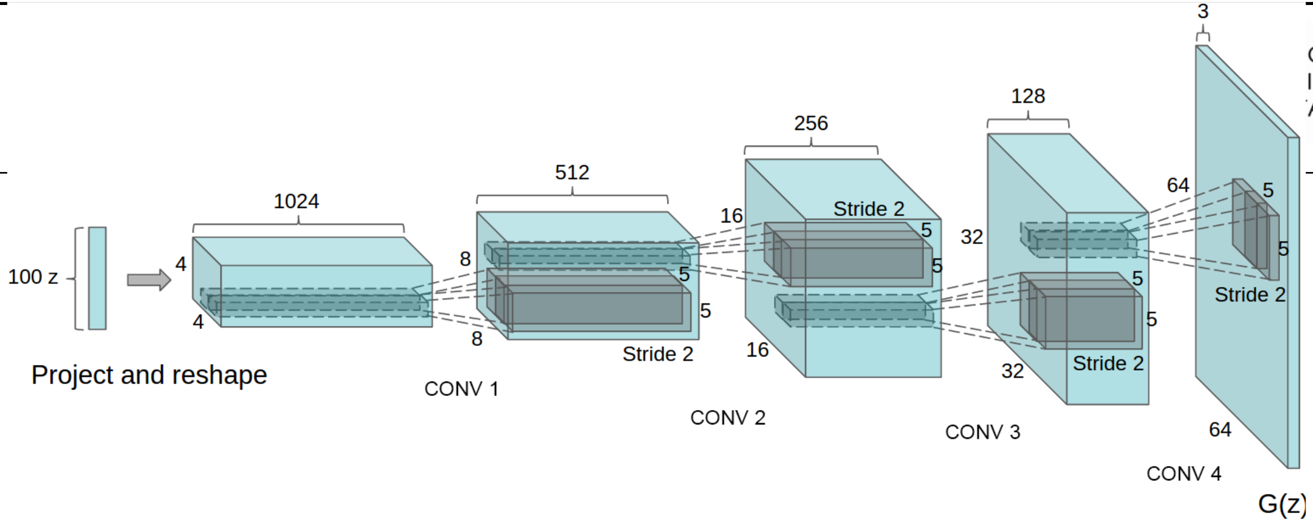
- Data Simulation
  - Class Imbalance
  - Data Augmentation

- Prostate Lesions
- Retina Patches
- Skin Lesions

Conditnl. (C-/ Cycle-GAN)

- CT from MR
- PET from CT/ MRI
- Stain Normalization

DC-GAN



Recipe

- Replace pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm
- Use LeakyReLU in discriminator
- Use ReLU in generator for all layers except output, which uses Tanh.