

Resembling Better Lungs

Using Generative Adversarial Networks (GANs) to create CT-Scans of COVID-19 infected lungs

Dieter Hoogestraat, @opencampus GANs WiSe2021



Christian-Albrechts-Universität zu Kiel

Idea

Resemble CT lung scans of **Covid 19** infected persons as done in 2020 during the OpenCampus course “Machine Learning with **Tensorflow**”, using new techniques to get (perhaps) better results, i.e. getting better insights in the possibilities and practical issues of using GANs for reproducing detailed imaging products.

Agenda

- Idea (already done)
- Starting point: ML, TF and AF
- Pathways
- Results
- Entropy
- Conclusion & Possible Future Work

Start

OC SoSe202

Tensorflow

Ali Farooq

GAN

Covid19 CT

Resembling COVID-19

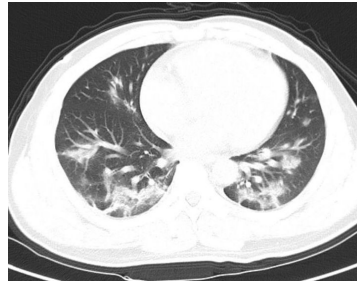
Using Generative Adversarial Networks (GAN's) to
create CT-Scans of COVID-19 infected lungs

Dieter Hoogestraat, Felix Facklam, Ali Farooq @opencampus Tensorflow SoSe2020

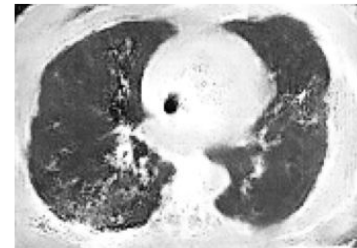


Start - former results

Real

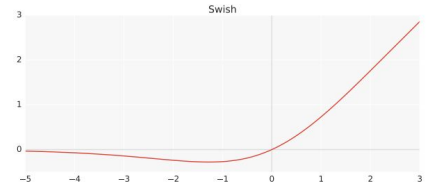


GAN



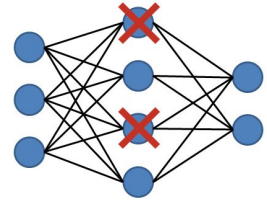
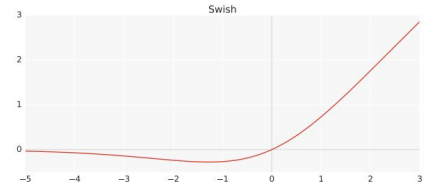
Methods

- Use another activation function
 - Transfer the job from TF to PyTorch
 - Implement a Wasserstein-GAN-GP
 - Add new CT-Scans, i.e. new data
- (added: examine the metalcover-results in relation to the initial Covid-GAN-results)



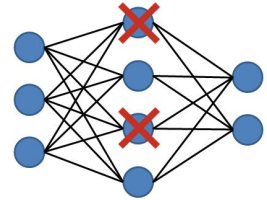
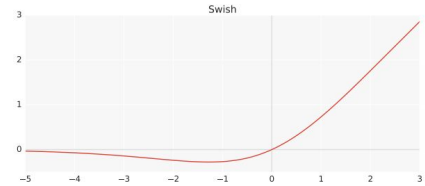
Methods

- Use another activation function
- Transfer the job from TF to PyTorch
- Implement a Wasserstein-GAN-GP
- Add new CT-Scans, i.e. new data
- (added: examine the metalcover-results in relation to the initial Covid-GAN-results)



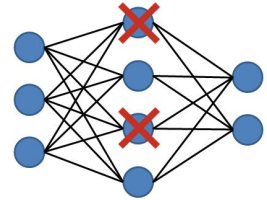
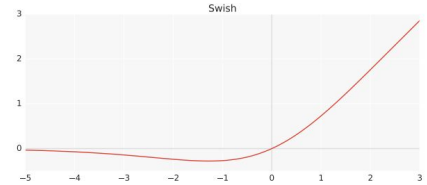
Methods

- Use another activation function
- Transfer the job from TF to PyTorch
- Implement a Wasserstein-GAN-GP
- Add new CT-Scans, i.e. new data
- (added: examine the metalcover-results in relation to the initial Covid-GAN-results)



Methods

- Use another activation function
 - Transfer the job from TF to PyTorch
 - Implement a Wasserstein-GAN-GP
 - Add new CT-Scans, i.e. new data
-
- (added: examine the metalcover-results in relation to the initial Covid-GAN-results)

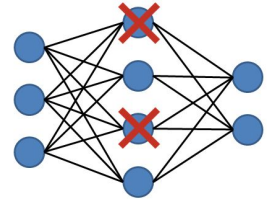
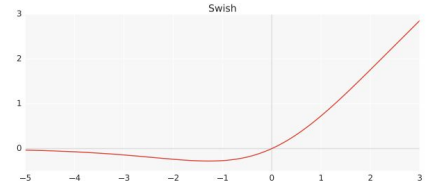
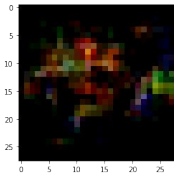


vs.



Methods

- Use another activation function
 - Transfer the job from TF to PyTorch
 - Implement a Wasserstein-GAN-GP
 - Add new CT-Scans, i.e. new data
-
- (added: examine the metalcover-results in relation to the initial Covid-GAN-results)



vs.



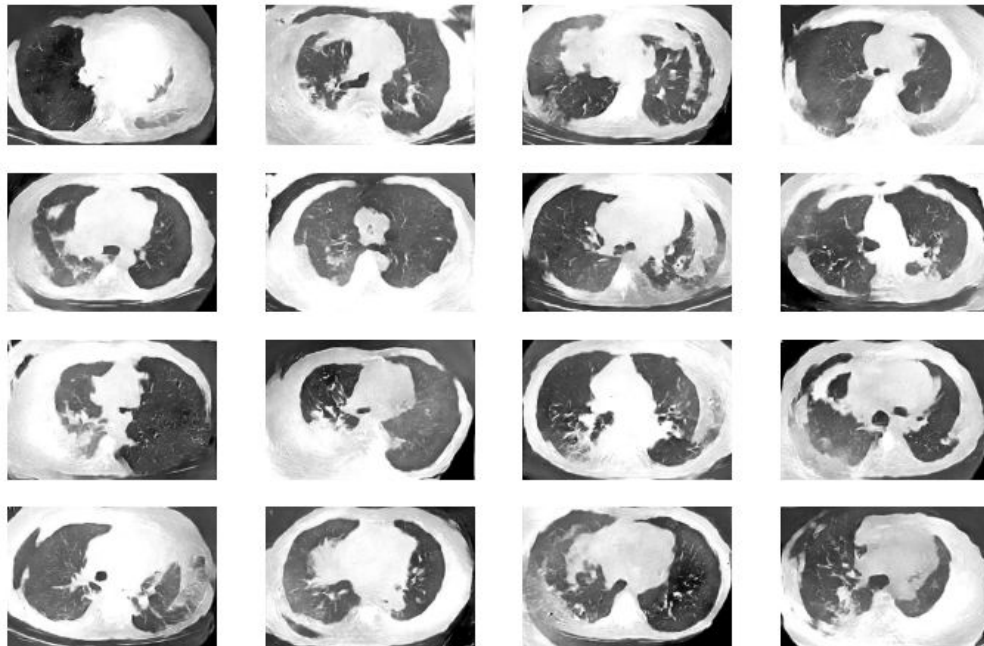
(some) Results



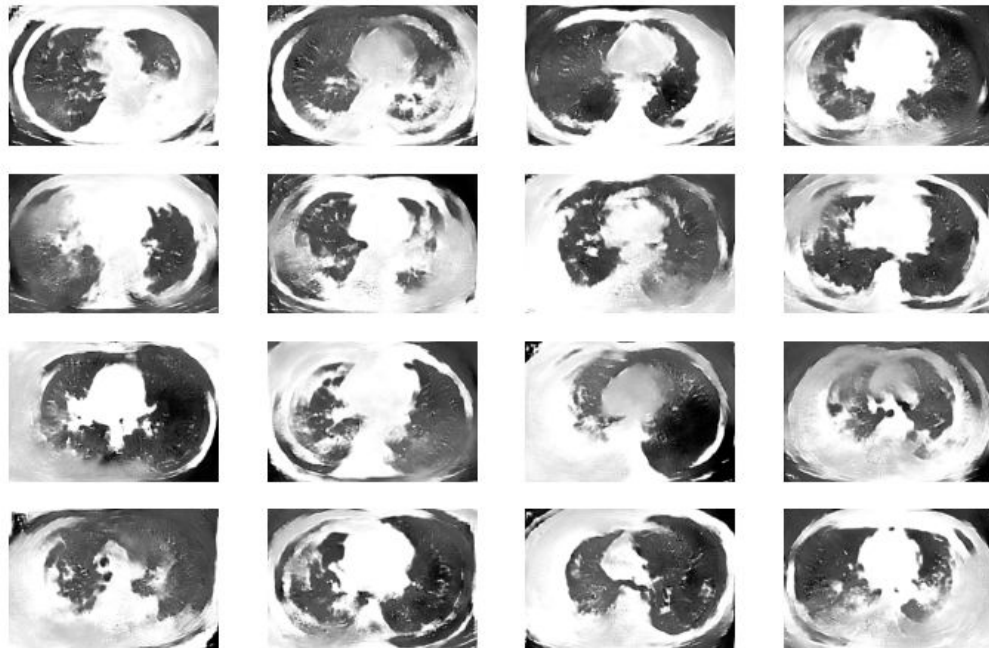
Christian-Albrechts-Universität zu Kiel

Resembling Better Lungs
Dieter Hoogstraat @opencampus GANs WS2021

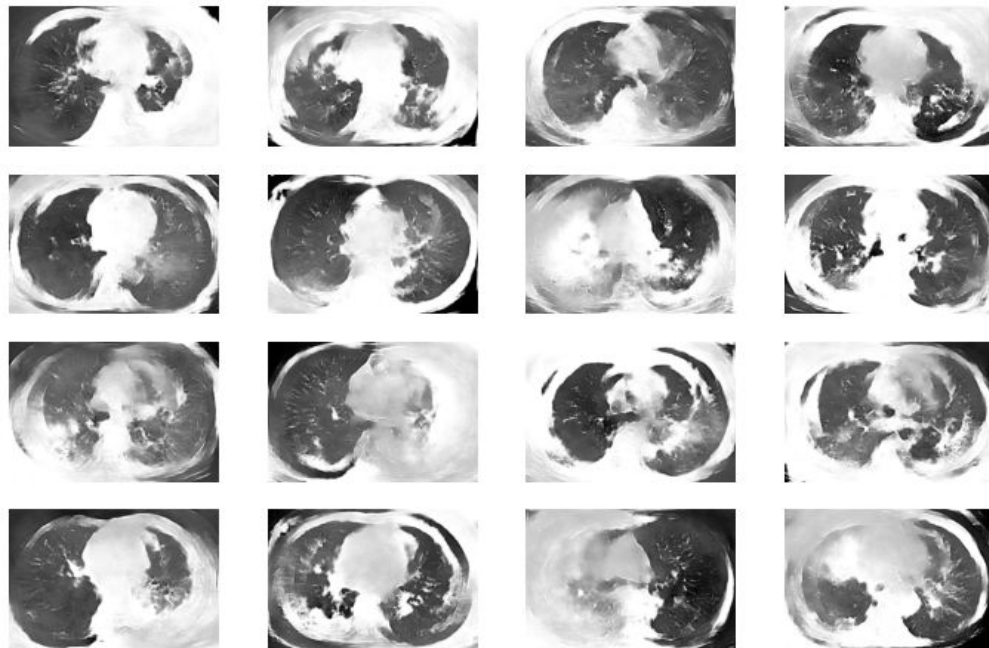
06-2020
LeakyRelu
no dropout
1.250 epochs



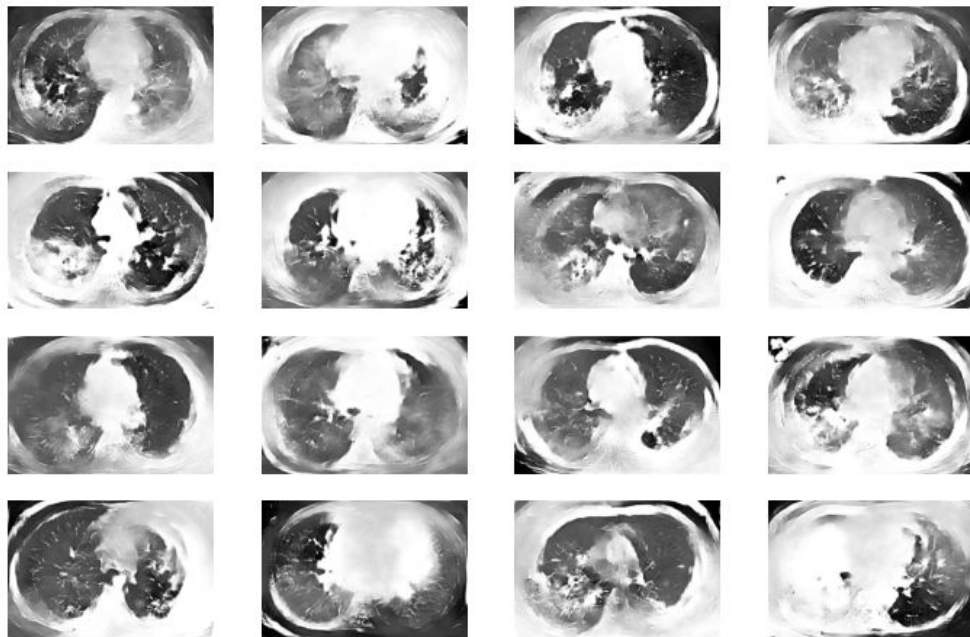
Swish
no dropout
1.250 epochs



Swish
30% dropout
1.250 epochs

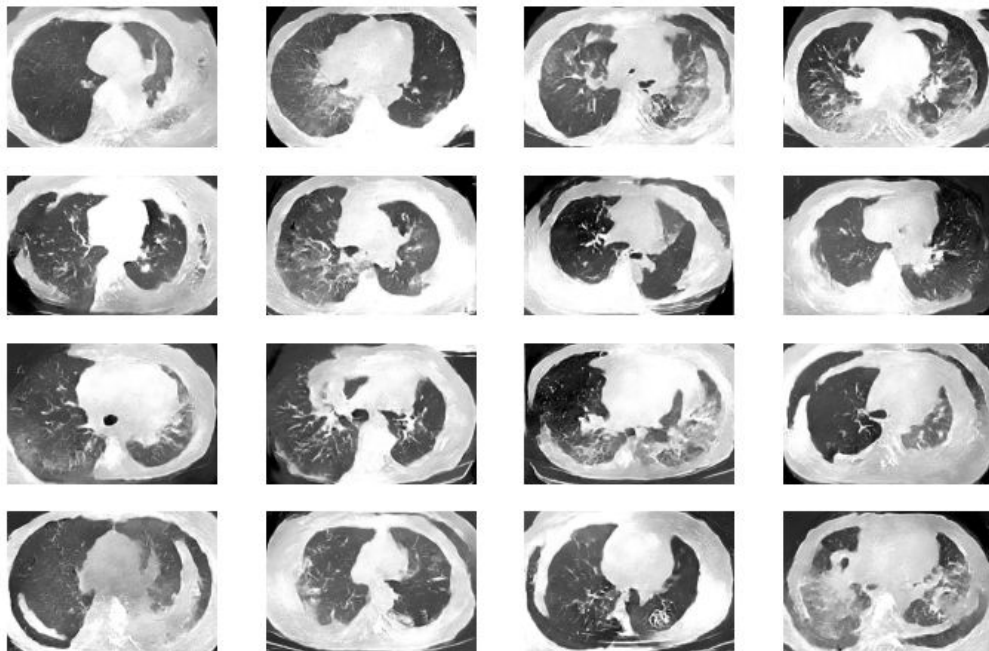


Swish
50% dropout
1.250 epochs



**LeakyRelu
no dropout
14.720 epochs**

**with lots of
instability**

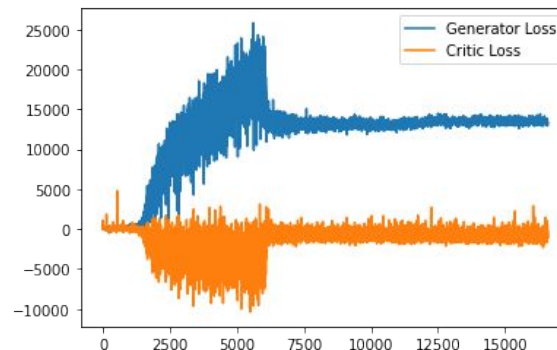
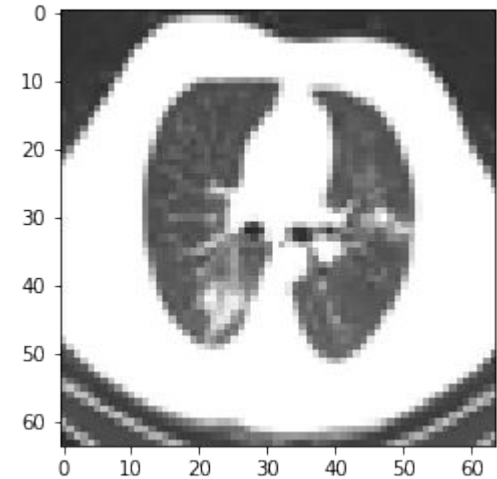
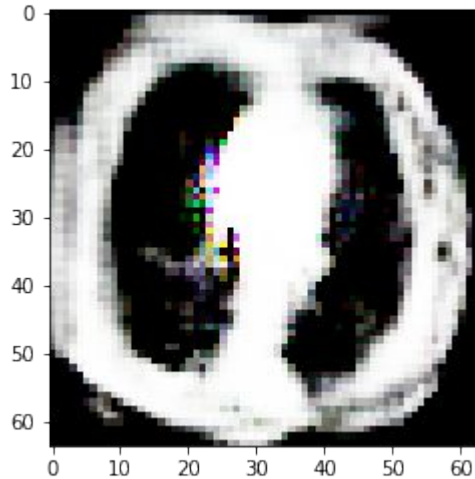


From TF-DCGAN to PyTorch-WGAN+GP

- Wasserstein-GAN with Gradient Penalty from week 3
 - running WGAN-GP on old data: some images caused problems
 - new data (-)
 - working on images took a lot of time => using Google-Colab Pro+
-
- Additionally: after metalcover-talk, I decided to use exactly that model, what took even more time

- PyTorch
- LeakyRelu
- no dropout
- Wasserstein
- GP

- LeakyRelu
- no dropout
- Wasserstein
- GP



Methods

“Batch Normalization in the discriminator breaks Wasserstein GANs with gradient penalty. The authors themselves advocate the usage of layer normalization instead, but this is clearly written in bold in their paper (<https://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf>). [...]

By the way transposed convolutions produce stairway artifacts in your output images. Use image resizing instead. For an indepth explanation of that phenomenon I can recommend the following resource (<https://distill.pub/2016/deconv-checkerboard/>).”

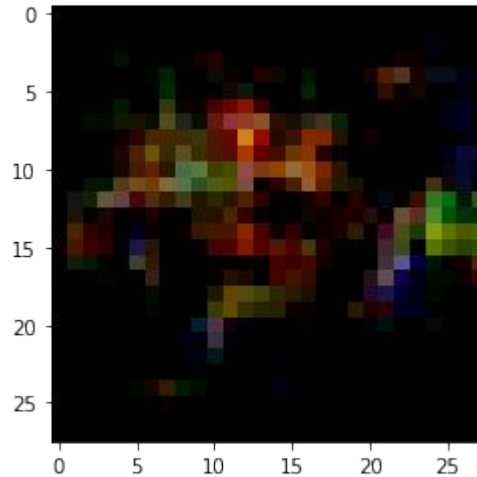
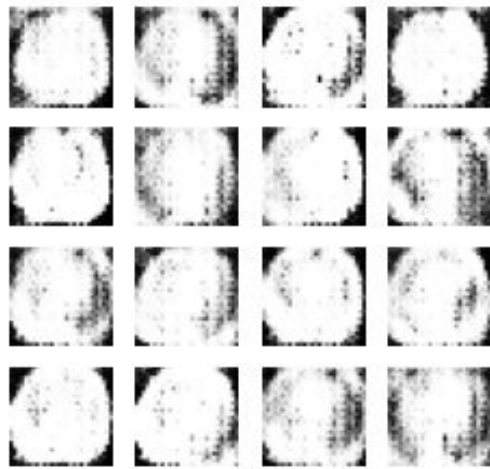
Thomas Peinitz

<https://stackoverflow.com/users/3994824/thomas-pinetz>

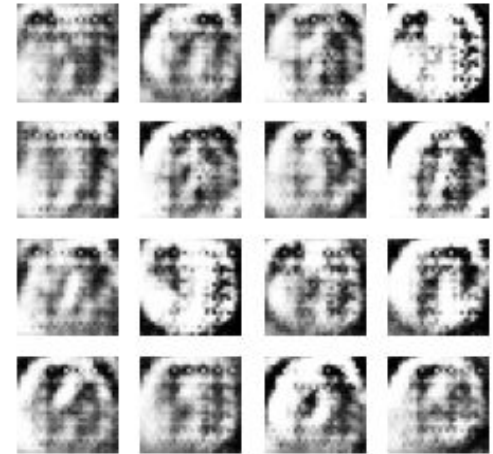
Results from 2020 - Déjà-vu

500 epochs, learning rate 1E-4, resolution 28x28

ReLu



LeakyReLu



Conclusion and Future Work

- Wasserstein-GANs don't work better just because they are Wasserstein GANs
- Batch Normalization could be a problem for Wasserstein GANs
- Easily extensible code is available, easy data handling not
- Generated images can be used for future covid-19 projects