Imperial College London

AI in Medicine I

Advanced Machine Learning: Class imbalance, data augmentation and transfer learning

Daniel Rueckert
I31 – Chair for AI in Medicine and Healthcare
Faculty of Informatics and Medicine

Introduction

- In machine learning, in particular deep learning, the performance of our methods is dependent on:
  - Machine learning model (e.g. deep neural network architecture, loss function, …)
  - **Quality** and **quantity** of training data

- We will look at a number of concepts:
  - Class imbalance
  - Data augmentation
  - Transfer learning

Class imbalance

- When training our machine learning models for a classification task, we may have **different number of samples for each class**.
  – E.g. in a clinical study with uneven distribution fo gender, age, disease status…

- This is called **class imbalance**, and may lead to the overprediction of the class which has more samples present.
  – This leads to a bias and lack of fairness

- One solution would be to simply add more samples of the underrepresented class(es), but that may not always be possible:
  – Small trials, rare diseases, lack of clinical annotations / cost / time-consuming…

Class imbalance: The challenge

- Looking at a binary classification problem only, recall definition of accuracy (ACC)

$$ACC = \frac{TP+TN}{P+N},\ P = TP + FN,\ N = TN + FP$$

  – If we have only few positive examples, *ie P << N*, accuracy reporting will be dominated by the negative examples.

Example: 1%, positive and 99% negative samples

➢ *Predicting *all* samples as negative, ACC = 99%*

- Precision $PPV = \frac{TP}{TP+FP}$

  – more sensitive to class imbalance because it considers the number of FP

  – *but provides not insight into number FN*

Class imbalance: sensitivity vs specificity

- In a clinical context, sensitivity is the ability of a test to **correctly identify those with the disease**, whereas test specificity is the ability of the test to correctly **rule out disease**

- **Sensitivity** evaluates a model's ability to predict true positives of each available category

$$TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$$

True positive rate
- does not relate to FP

- **Specificity** evaluates a model's ability to predict true negatives of each available category.

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP}$$

True negative rate
- does not relate to FN

Class imbalance: Balanced accuracy

- **Balanced accuracy** is a lot lower than the conventional ACC measure when either the sensitivity (TPR) or specificity (TNR) is low due to a bias in the classifier towards the dominant class:

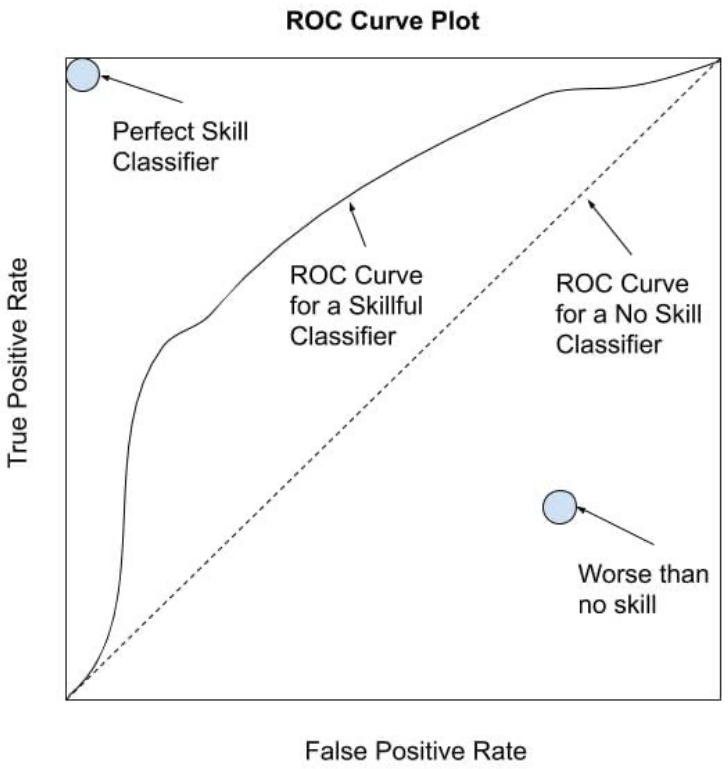
$$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)=\frac{1}{2}\left(\frac{TP}{P}+\frac{TN}{N}\right)=\frac{TPR+TNR}{2}$$

  – The weighting between TP/P and TN/N can be shifted

- This measure offers a good performance measure for imbalanced class problems

Class imbalance: ROC

- Receiver Operating Characteristic (**ROC):**
  - summarises the behaviour of a model by calculating the sensitivity and specificity for a set of predictions by the model under different thresholds
  - Area under the curve offers another good performance measure for imbalanced class problems
  - Want to maximise area under ROC curve



What can we do about class imbalance?

- The **bias towards a majority class** can be alleviated by:
  - Altering the training data to decrease imbalance
  - Modifying the loss function to increase sensitivity towards the minority group.

- Methods for **handling class imbalance**:
  - data-level techniques
  - algorithm-level methods
  - hybrid approaches

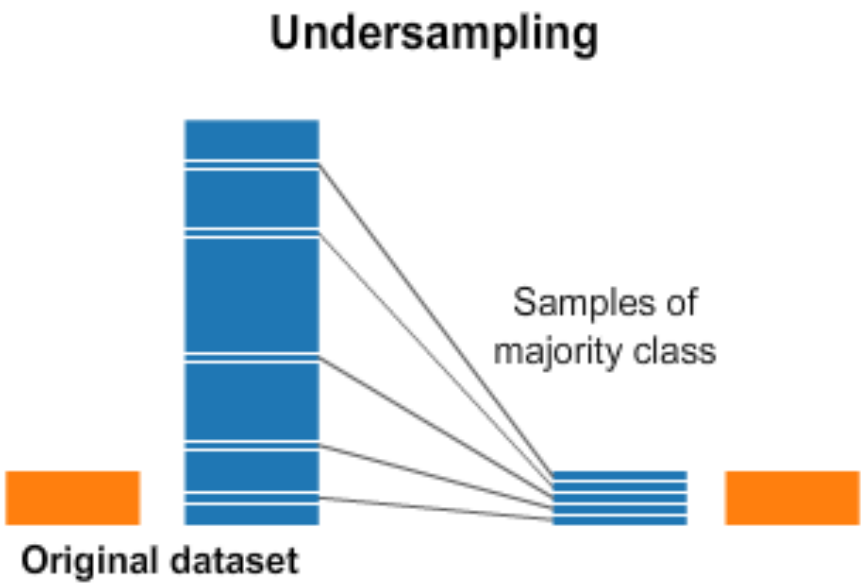
Data-level techniques for handling class imbalance

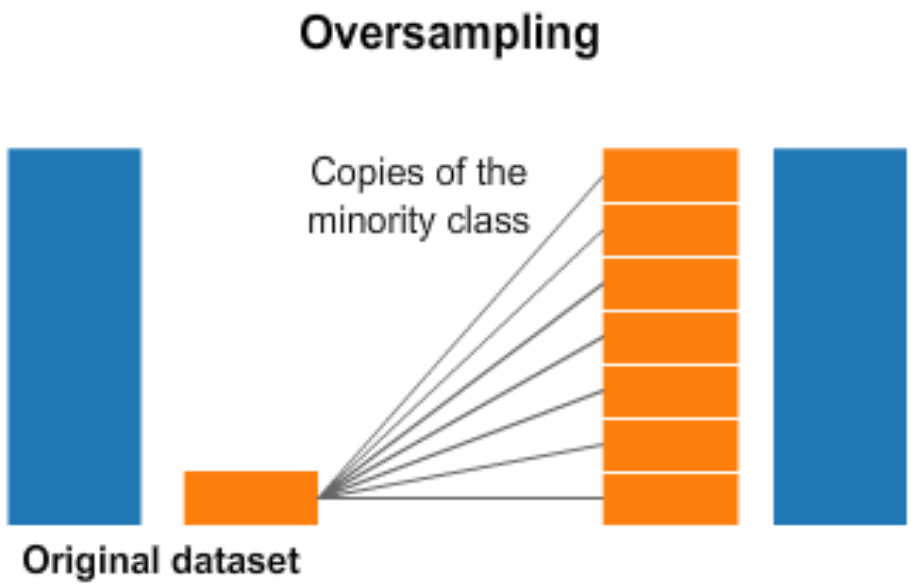
Over-sampling and **under-sampling** modify the training distributions to decrease the level of imbalance or reduce noise (mislabelled sample)

The most naïve form consists of **random resampling**:

- **Random undersampling** discards random samples from majority class
  - Reduces total amount of information model can learn from
- **Random oversampling** duplicates random samples from minority class
  - Increases training time and may lead to overfitting

Data-level techniques for handling class imbalance





https://www.analyticsvidhya.com

Data-level techniques for handling class imbalance

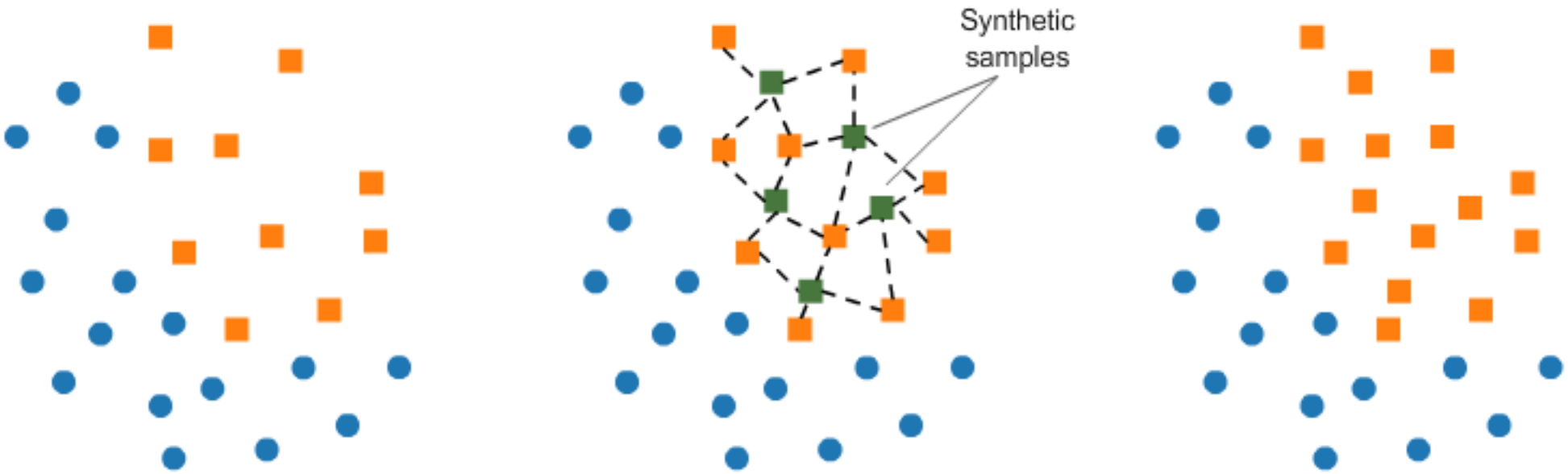
- **Synthetic Minority Over-sampling Technique (SMOTE**) is a more intelligent oversampling technique:

- Generates artificial minority samples by interpolating between existing minority samples and their nearest minority neighbour:

1. Choose a minority class as the input vector
2. Find its **k Nearest Neighbours**
3. Choose one of these neighbours and place a synthetic point anywhere on the line joining the point under consideration and its chosen neighbour
4. **Repeat** steps 1-3 **until data is balanced**

Data-level techniques for handling class imbalance

- **SMOTE** illustration:



https://www.analyticsvidhya.com

Algorithmic methods for handling class imbalance

- Instead of altering the training data distribution, the learning or decision process can be adjusted to increases the importance of minority class.

- Two main approaches exist:
  - Modify the loss function to take a class penalty or weight into consideration
    - E.g. include balanced accuracy in the loss function
  - Shift the decision threshold, in order to reduce bias towards the majority class
    - Upweighting of minority class

Hybrid approaches for handling class imbalance

- One can also combine data-level and algorithmic methods, e.g.:
  - Perform data sampling to reduce class noise and imbalance, followed by thresholding to further reduce the bias towards the majority group
  - Create ensemble of differently resampled datasets

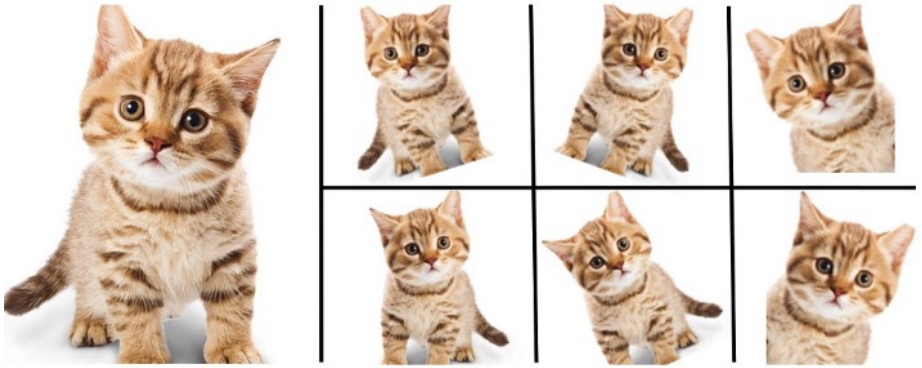
Data augmentation

- If there is not enough training data (balanced or not), we can also synthesise new training samples
  - Simple methods operate directly on the training data at hand
  - Generative models learn new instances from the existing training data
  - More realistic models use further prior knowledge

Data augmentation – simple techniques

- The simplest data augmentation techniques apply some random geometric transformations to the input image data
  - translation
  - rotation
  - scaling (zooming)

} Global (rigid body or affine) transformations



https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

Data augmentation – simple techniques

- Additionally, some very simple image operations can be applied:
  - Shearing and skewing
  - flipping
  - cropping
  - padding

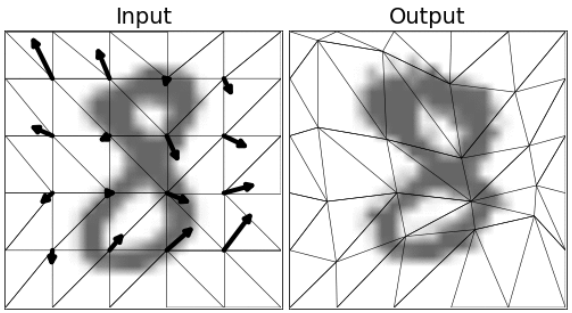
Still affine transformations!

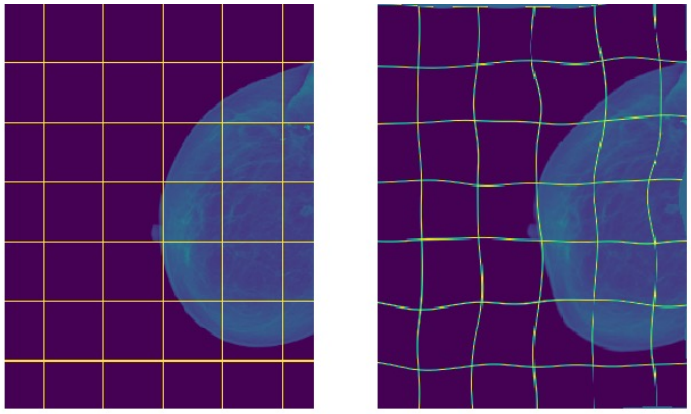
https://medium.com/analytics-vidhya/data-augmentation-in-deep-learning-3d7a539f7a28

Data augmentation – simple techniques

- The simplest data augmentation techniques apply some random geometric transformations to the input image data
  - Warping

Local (nonlinear or deformable) transformations





https://commons.wikimedia.org/wiki/File:An_Example_of_Data_Augmentation_via_Augmentor.png

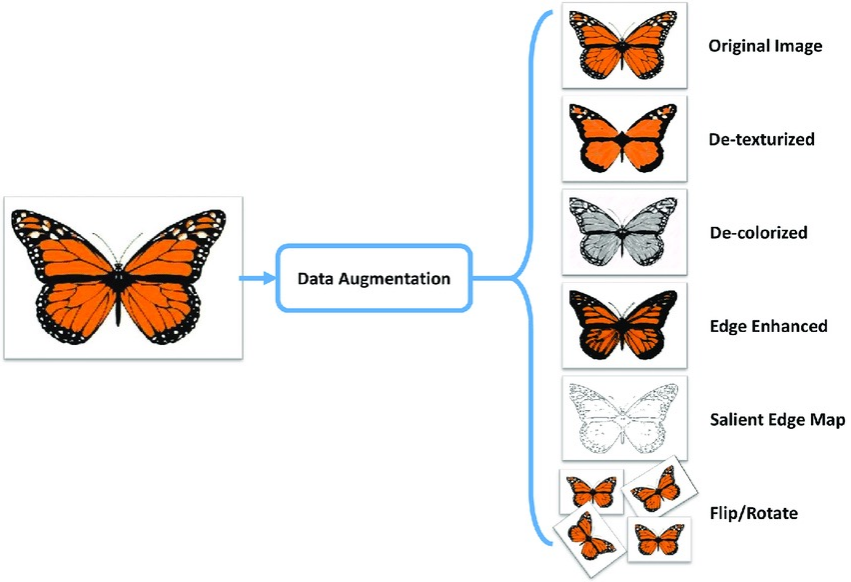
DOI: 10.1109/BHI.2018.8333411 • Corpus ID: 4708077

Elastic deformations for data augmentation in breast cancer mass detection

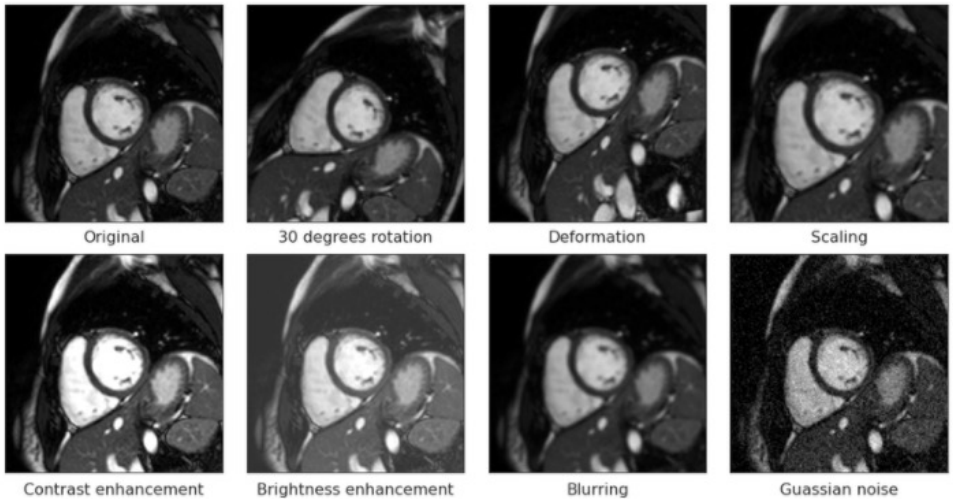
Eduardo Castro, Jaime S. Cardoso, J. C. Pereira • Published 1 March 2018 • Computer Science •
2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)

Data augmentation – simple techniques

- We can also operate directly on image intensities:
  - intensity modification (e.g. change contrast, brightness, …)
  - adding noise



https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722



IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. XX, NO. XX, XXXX 2020 1

Multi-Centre, Multi-Vendor and Multi-Disease Cardiac Segmentation: The M&Ms Challenge

Víctor M. Campello, Polyxeni Gkontra, Cristian Izquierdo, Carlos Martín-Isla, Alireza Sojoudi, Peter M. Full, Klaus Maier-Hein, Yao Zhang, Zhiqiang He, Jun Ma, Mario Parreño, Alberto Albiol, Fanwei Kong, Shawn C. Shadden, Jorge Corral Acero, Vaanathi Sundaresan, Mina Saber, Mustafa Elattar, Hongwei Li, Bjoern Menze, Firas Khader, Christoph Haarburger, Cian M. Scannell, Mitko Veta, Adam Carscadden, Kumaradevan Punithakumar, Xiao Liu, Sotirios A. Tsaftaris, Xiaoqiong Huang, Xin Yang, Lei Li, Xiahai Zhuang, David Viladés, Martín L. Descalzo, Andrea Guala, Lucia La Mura, Matthias G. Friedrich, Ria Garg, Julie Lebel, Filipe Henriques, Mahir Karakas, Ersin Çavuş, Steffen E. Petersen, Sergio Escalera, Santi Seguí, José F. Rodríguez-Palomares, and Karim Lekadir

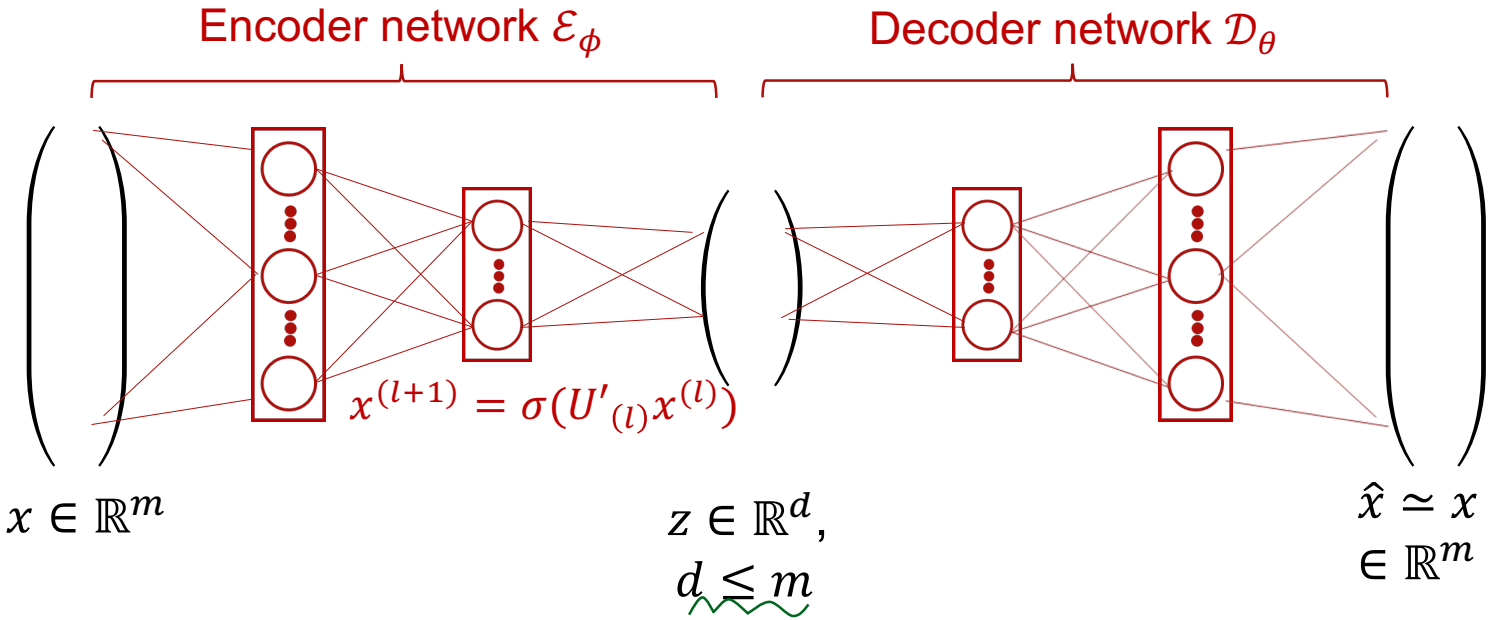
Data augmentation: Generative techniques

- Generative models lend themselves for synthesising new data from the training data

- Examples include:
  - Variational autoencoders (VAEs)
  - Generative adversarial networks (GANs)
  - Neural style transfer (styleGANs)

Autoencoders

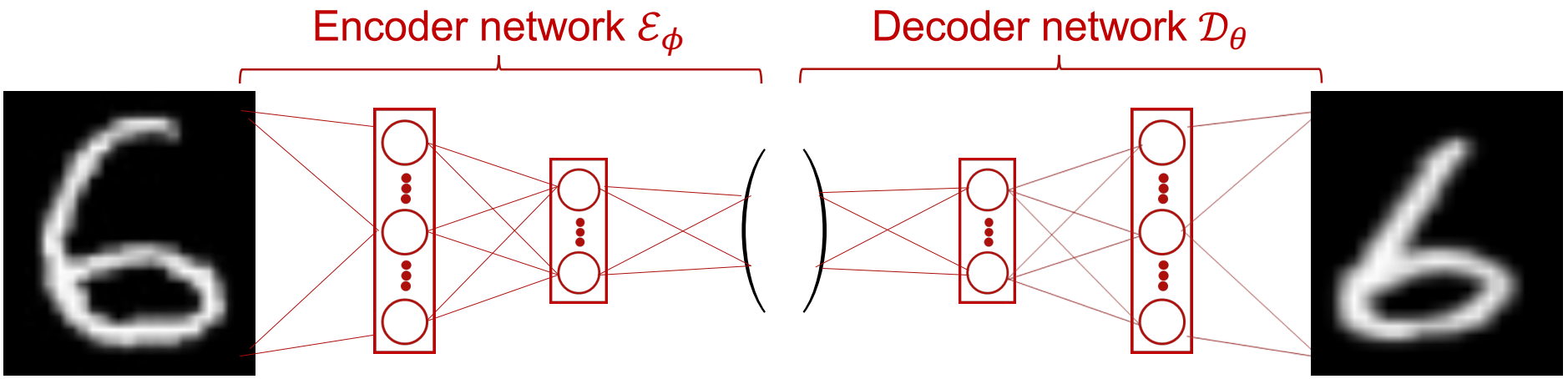
- Auto-Encoders are general encoder/decoder architectures with non-linearities



(not always, e.g. sparse autoencoders)

Reconstruction loss $\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2, \qquad \hat{x} = \mathcal{D}_\theta(\mathcal{E}_\phi(x))$

Autoencoders



I
$\in \mathbb{R}^{H \times W}$

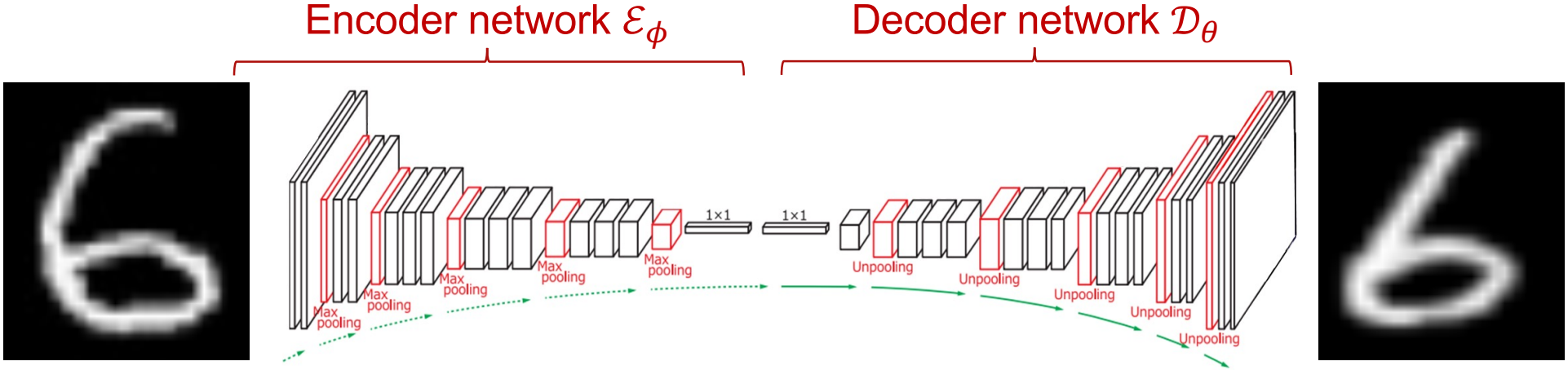
$z \in \mathbb{R}^d,$
$d \leq H \times \mathrm{W}$

$\hat{I} \simeq I$
$\in \mathbb{R}^{H \times W}$

Reconstruction loss $\mathcal{L}(I, \hat{I}) = \left\| I - \hat{I} \right\|^2,$ $\hat{I} = \mathcal{D}_\theta(\mathcal{E}_\phi(I))$

Autoencoders

- Many possible architectures (e.g. Convolutional Autoencoder)



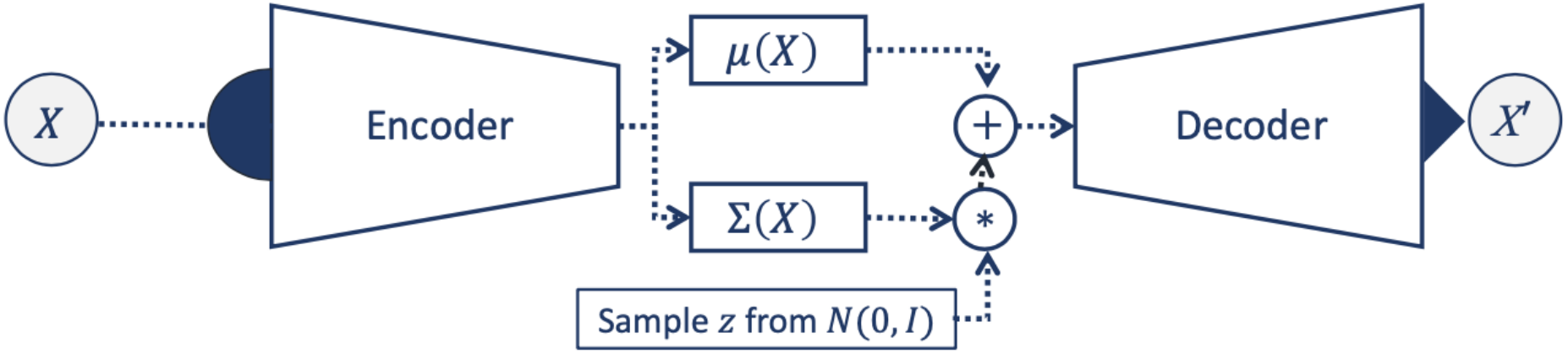
$I \in \mathbb{R}^{H \times W}$

$z \in \mathbb{R}^d$, $d \leq H \times \mathrm{W}$

$\hat{I} \simeq I \in \mathbb{R}^{H \times W}$

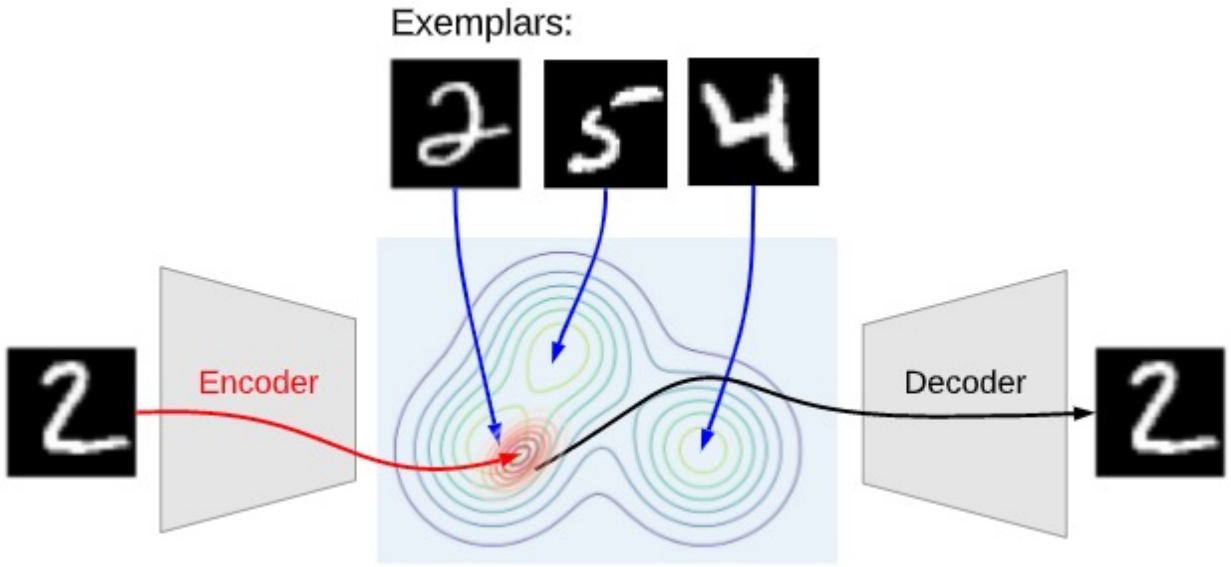
Reconstruction loss $\mathcal{L}(I, \hat{I}) = \left\| I - \hat{I} \right\|^2, \quad \hat{I} = \mathcal{D}_\theta(\mathcal{E}_\phi(I))$

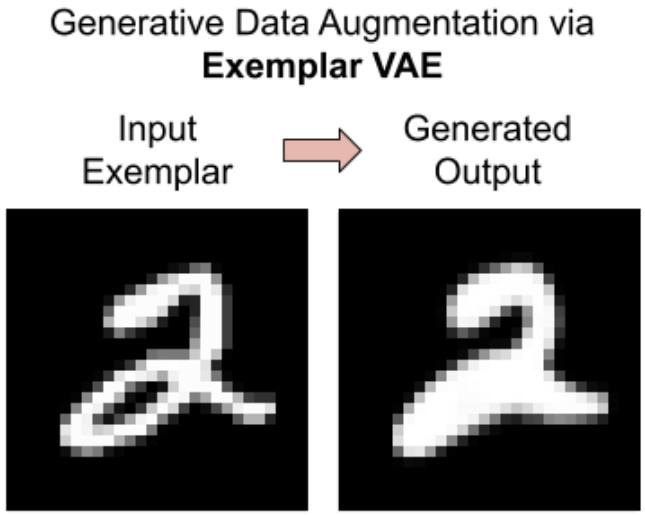
Variational autoencoders (VAE)



Loss Function $\|X - X'\| + \lambda \cdot KL\left[N\left(\mu(X), \Sigma(X)\right) \| N(0, I)\right]$

Data augmentation - VAEs

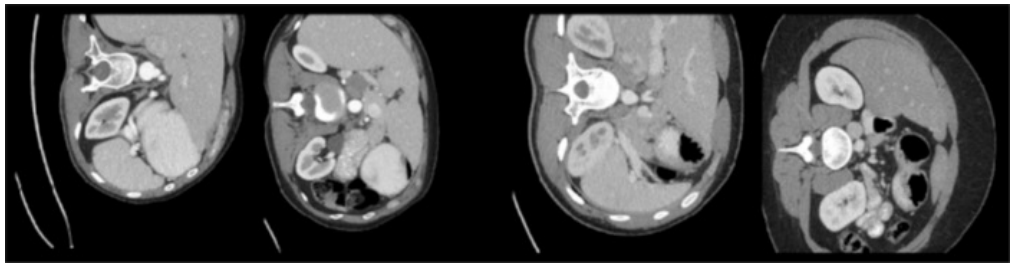




https://exemplar-vae.github.io/

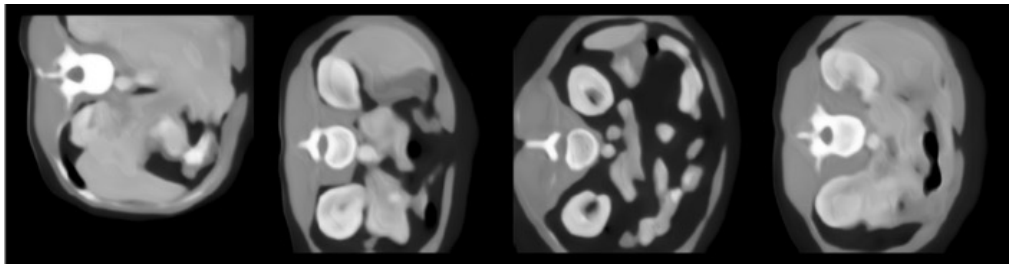
Data augmentation - VAEs

- Mathematically elegant and robust
  - But produce synthetic images of limited spatial sharpness



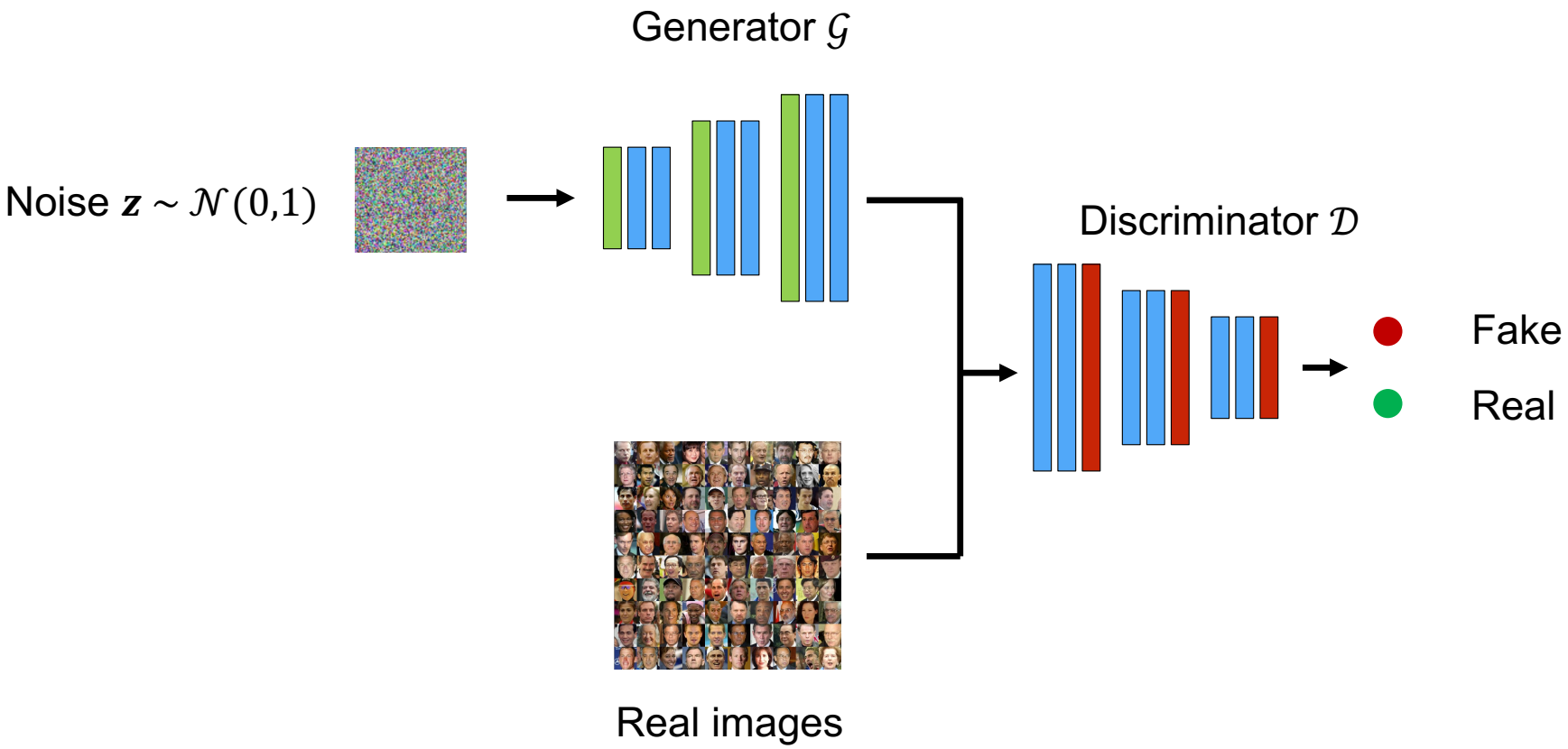
Exemplary abdominal CT image slices from the TCIA pancreas data set.

Synthetically generated abdominal CT image slices.



https://medium.com/miccai-educational-initiative/tutorial-abdominal-ct-image-synthesis-with-variational-autoencoders-using-pytorch-933c29bb1c90

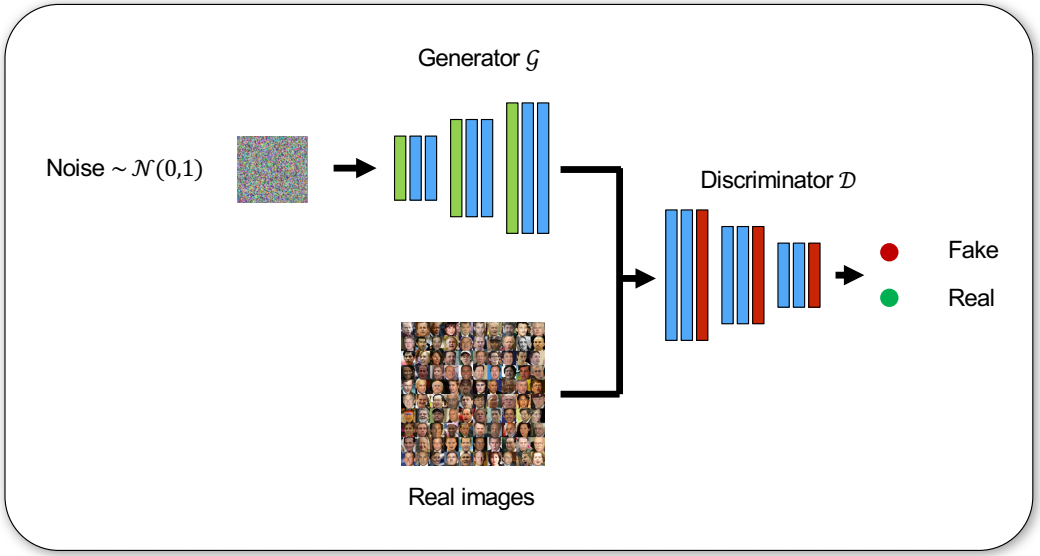
Data augmentation - GANs



I. Goodfellow, NIPS, 2014

Generative adversarial networks (GANs)

- $\mathcal{D}(x)$ represents the probability that x came from the real data rather than from $\mathcal{G}$.
- $\mathcal{D}$ is trained to maximize the probability of assigning the correct label to both real examples and samples from $\mathcal{G}$.



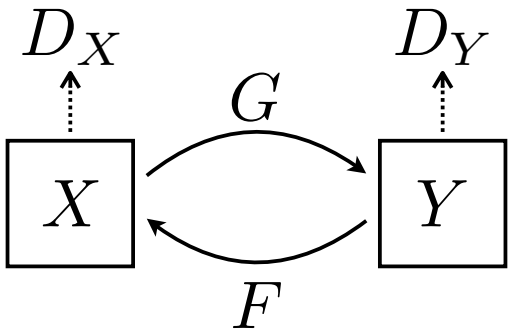
- Simultaneously train $\mathcal{G}$ to minimize $\log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z}))$, e.g. $\mathcal{D}$ and $\mathcal{G}$ play a two-player minimax game with value function $V(\mathcal{G}, \mathcal{D})$:

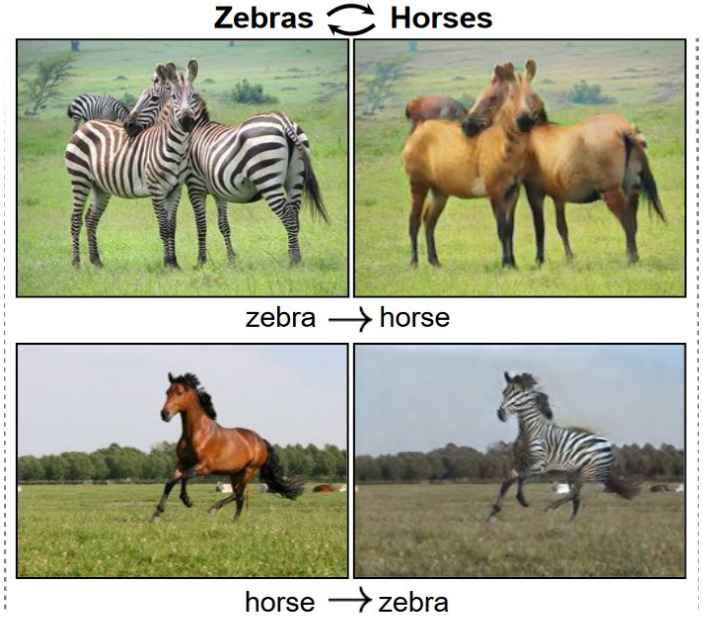
$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}) = \underbrace{\mathbb{E}_{x \sim p_{data}(x)}[\log \mathcal{D}(x)]}_{\text{real}} + \underbrace{\mathbb{E}_{z \sim p_z(z)}[\log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z}))]}_{\text{synthetic (aka “fake”)}}$$

Data augmentation - GANs

- **Generative adversarial networks (GANs)**: GAN algorithms can learn patterns from input datasets and automatically create new examples which resemble the training data.

- E.g. CycleGan:



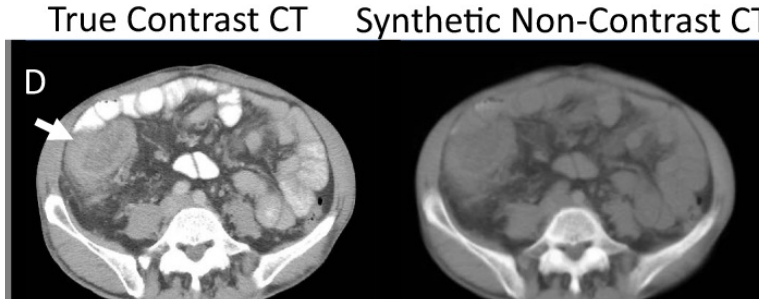


Data augmentation - GANs

- **Generative adversarial networks (GANs)**: GAN algorithms can learn patterns from input datasets and automatically create new examples which resemble the training data.

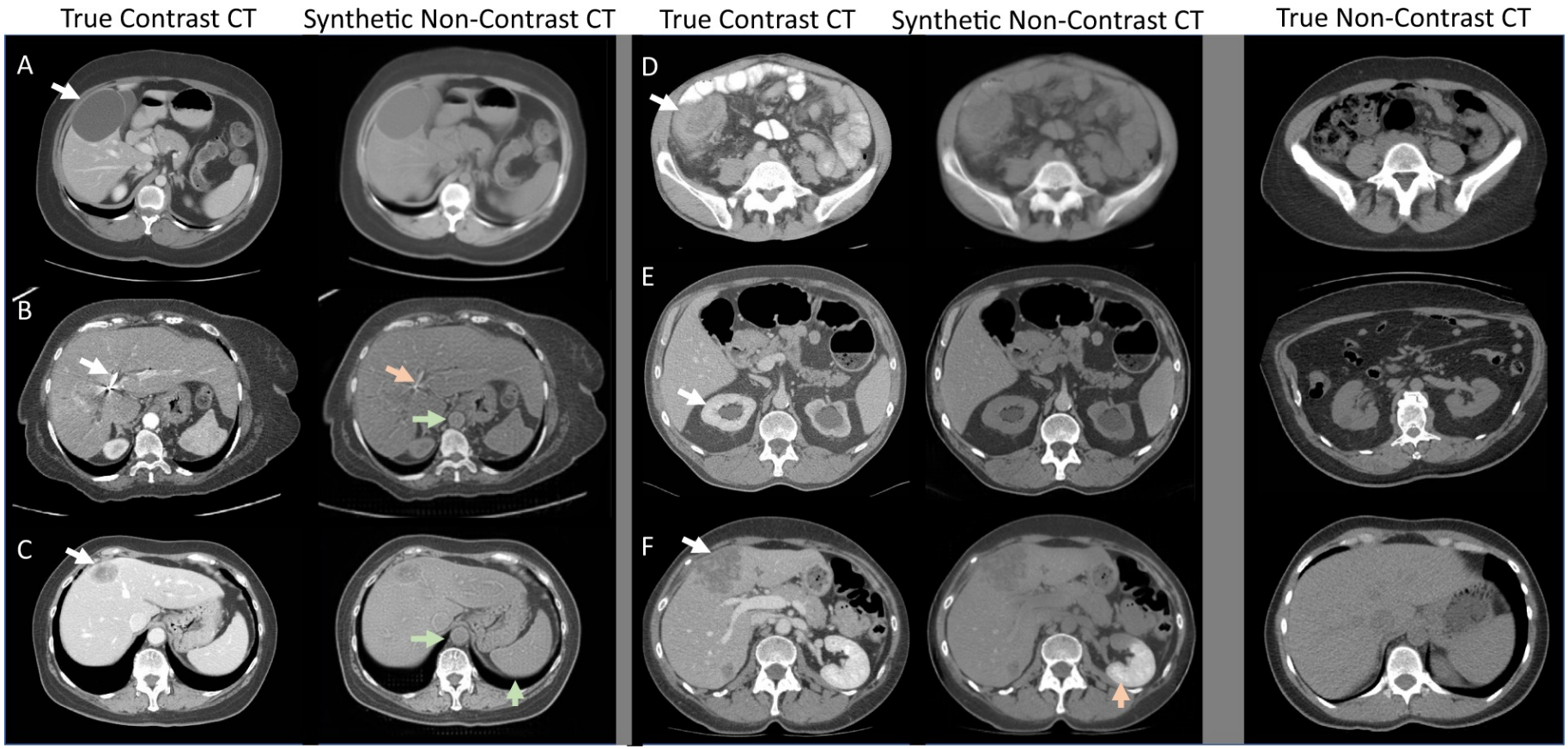
- E.g. CycleGan:

  1. Generate a synthetic version of training data
  2. Train on the original data while using the synthetic data for data augmentation



Sandfort, V., Yan, K., Pickhardt, P.J. *et al.* Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Sci Rep* **9,** 16884 (2019). https://doi.org/10.1038/s41598-019-52737-x

Example: Data augmentation using CycleGAN to improve generalizability in CT segmentation tasks*



Sandfort, V., Yan, K., Pickhardt, P.J. *et al.* Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Sci Rep* **9,** 16884 (2019). https://doi.org/10.1038/s41598-019-52737-x

Data augmentation – Neural style transfer

- **Neural style transfer**: Neural style transfer models can blend content image and style image and separate style from content.

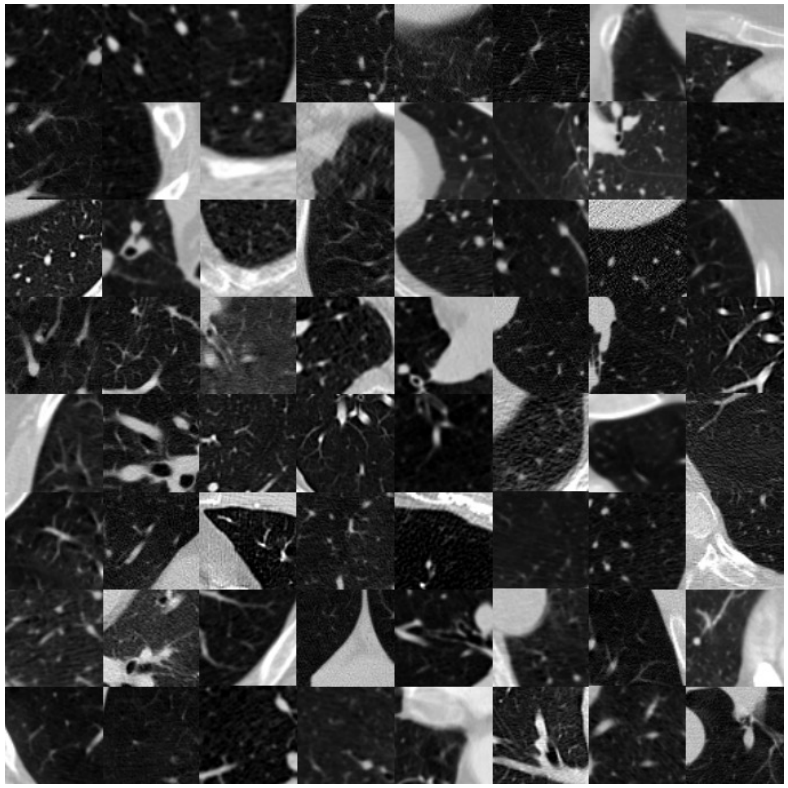
A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras
NVIDIA
tkarras@nvidia.com

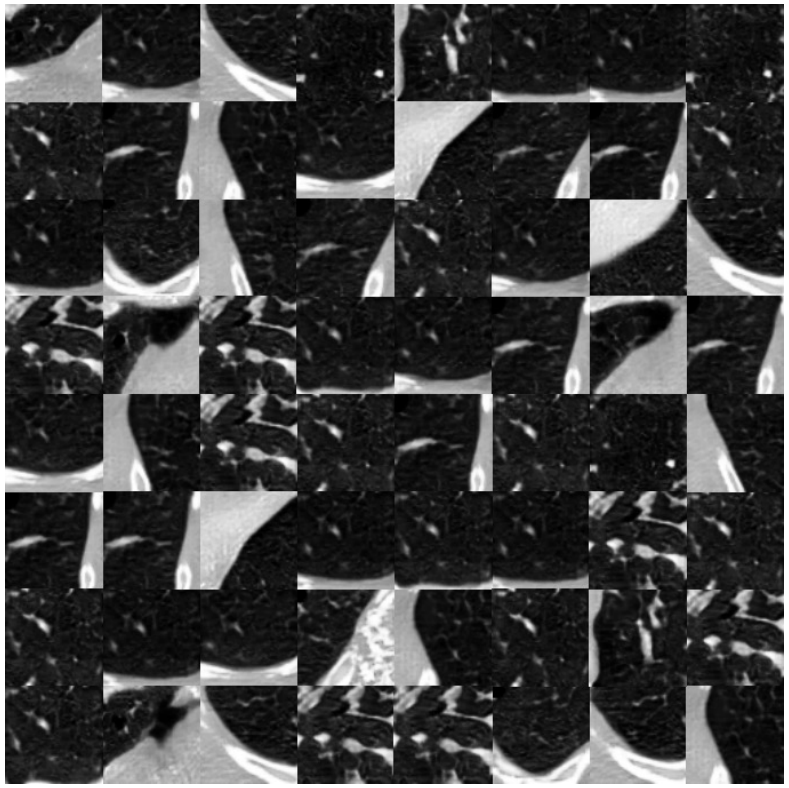
Samuli Laine
NVIDIA
slaine@nvidia.com

Timo Aila
NVIDIA
taila@nvidia.com

Data Augmentation – Lung CT*



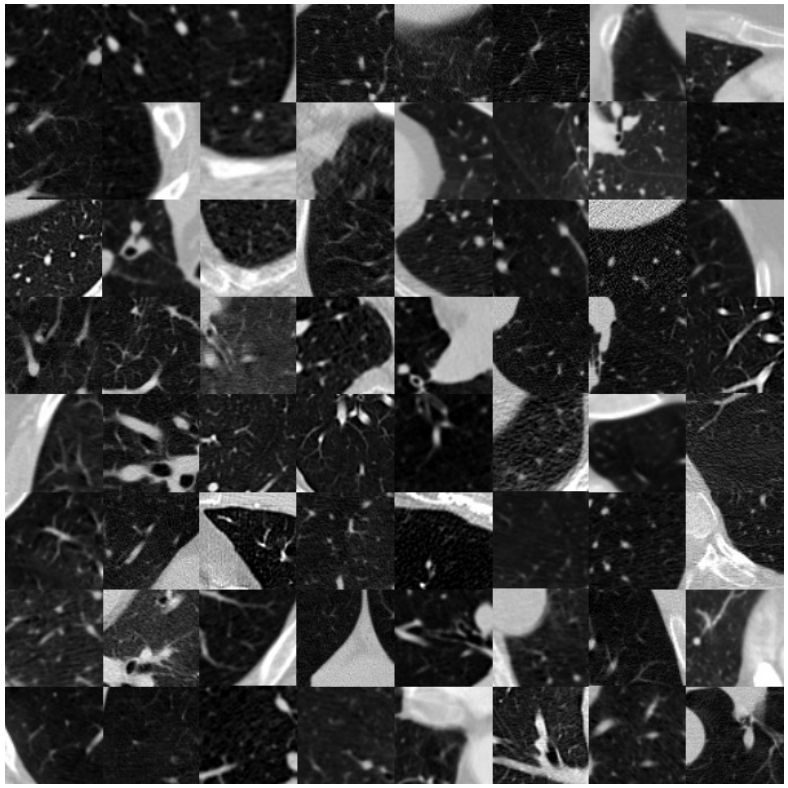
Real samples



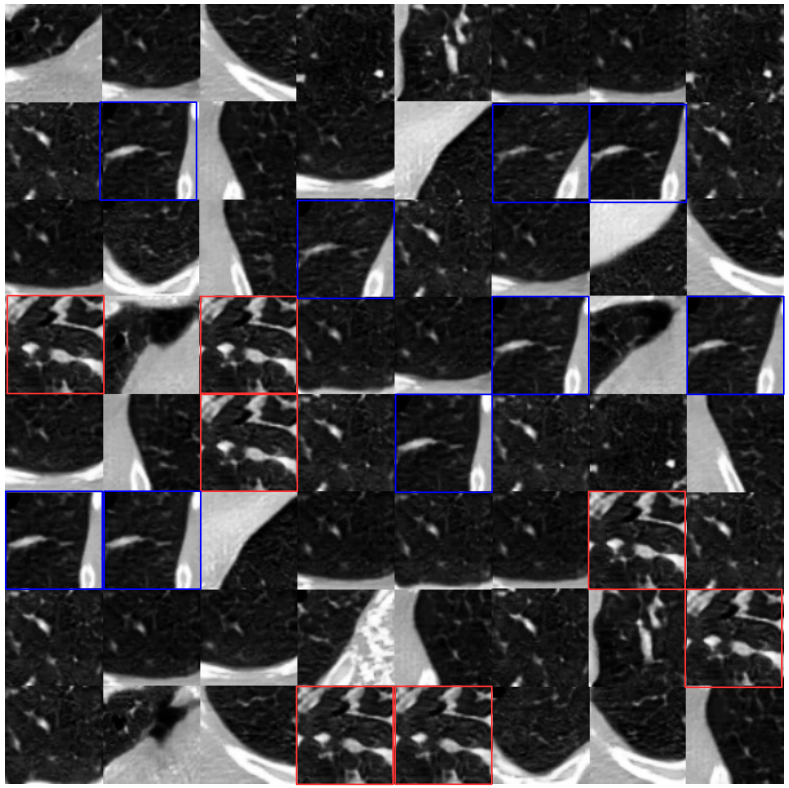
GAN samples

Slide courtesy of Sam Elllis, King's College London, UK

Data Augmentation – StyleGAN: Lung CT*



Real samples

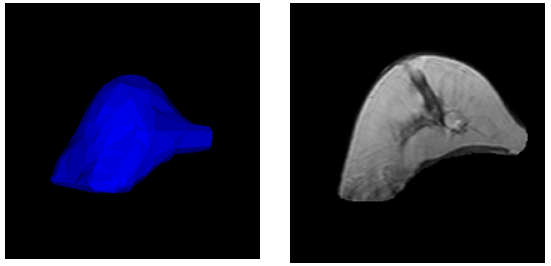


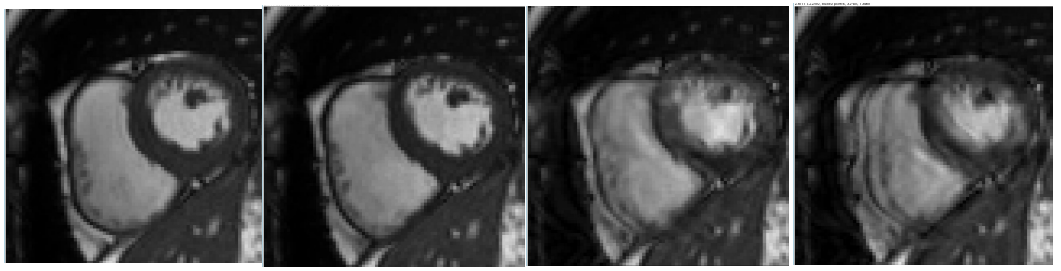
GAN samples

Slide courtesy of Sam Elllis, King's College London, UK

Data augmentation – more realistic techniques

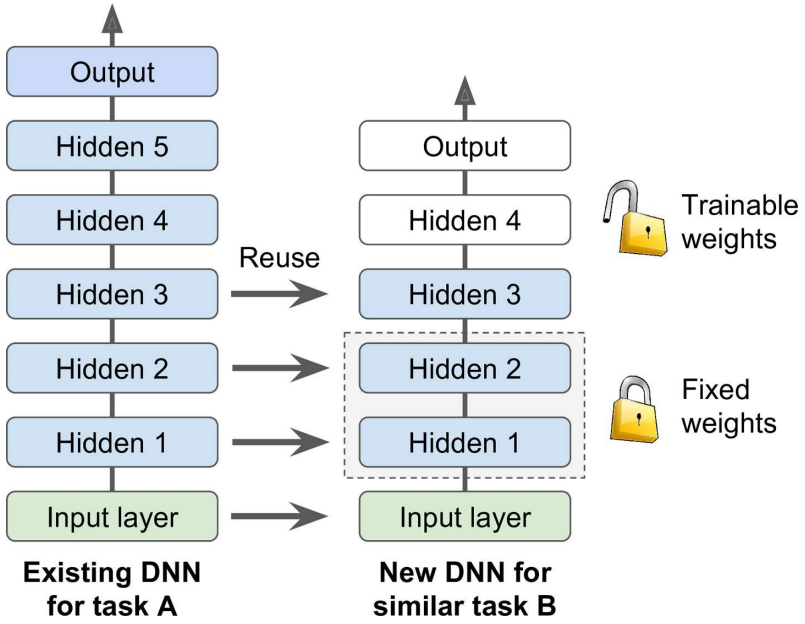
- Medical imaging offers additional data augmentation opportunities by including prior knowledge:
  - Motion models
    - E.g. use respiratory motion, deformation or shape models
  - Biophysical disease models
    - E.g. use tumour growth models
  - Imaging physics
    - E.g. use image simulators (MR, ultrasound, …) or image artefact replication
  - …





Transfer learning

- When moving from one (similar) imaging domain or clinical application to another, you may not wish to train your deep neural network architecture from scratch, but just recycle it:



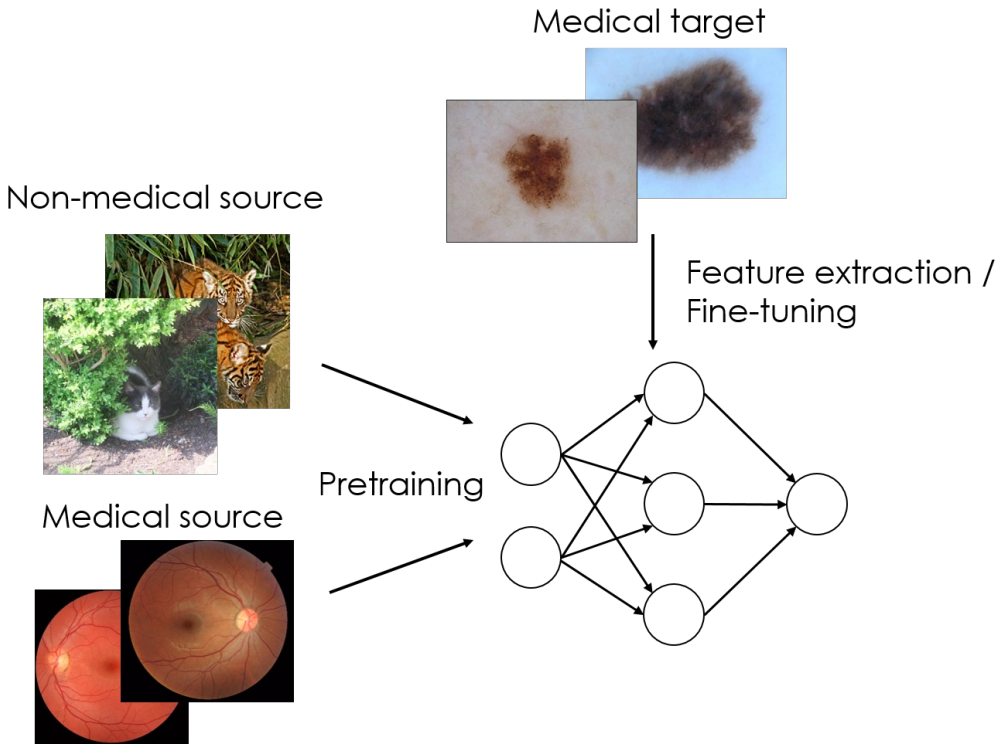
Src: Hands-on Machine Learning with Scikit-Learn & Tensorflow

You can re-use the pre-trained lower layers and add new layers for training on the new task

This will speed up training and will require fewer new training data.

Transfer learning

- A classic example is to train on ImageNet to obtain a nice set of features, then freeze the first few layers, and train new added layers on e.g. medical images
  - E.g. you can use pretrained **VGGNet** on a medical imaging task!
  - Not dissimilar to pretext tasks in self-supervised learning



Cats or CAT scans: transfer learning from natural or medical image source datasets?
Veronika Cheplygina arXiv:1810.05444

Summary

- In medical imaging we often lack enough data for training our machine learning models
  - Problem of **class imbalance**:
    - Leads to bias to majority class, ie overprediction of majority class
    - Can be alleviated with:
      - Data-level methods, ie alterting the training dustribution (e.g. by resampling)
      - Algorithmic level methods, ie altering the learning process (e.g. loss function)
  - Problem of too few data:
    - Leads to overfitting
    - Can be alleviated with **data augmentation**:
      - Simple methods: geometric or colour transformations, adding noise etc
      - Advanced methods: VAEs, GANs, StyleGAN, realistic methods
  - Problem of changing training domains:
    - Leads to lack of generalisation
    - Can be alleviated with **transfer learning**

(At least for now!)