LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Fakultät für Mathematik, Informatik und Statistik
Institut für Informatik

**Lehrstuhl für Datenbanksysteme
und Data Mining**

Anomaly Detection in X-Ray Images

Practical Course "Big Data Science"
DeepC Final Presentation
29.07.2019

Davletshina, Diana
Melnychuk, Valik
Singla, Hitansh
Tran, Viet

Presentation for:

deepc

About Us

Diana Davletshina

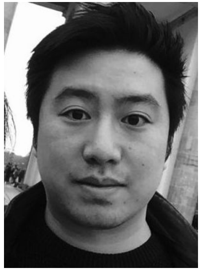
MSc Data Science (LMU)
Bachelors in Informatics
(Innopolis, Russia)

Valentyn Melnychuk

MSc Data Science (LMU)
Bachelors in System Analysis
(Kyiv, Ukraine)

Hitansh Singla

MSc Data Science (LMU)
Bachelors in Comp. Science
(New Delhi, India)



Viet Tran

MSc Biostatistics (LMU)
Bachelors in Statistics,
Philosophy
(LMU)

Mentors

Max Berrendorf

Research Assistant
Department of Informatik
LMU

Evgeniy Faerman

Research Assistant
Department of Informatik
LMU

Agenda

- Task & Data
- Preprocessing workflow
- Method & Models training
- Visualizations
- Future work

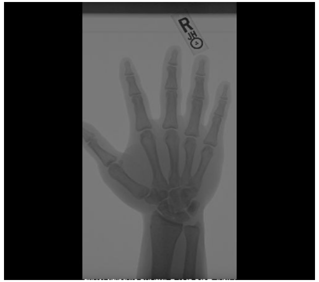
Task Allocated to Us

- Classify hand X-ray images into normal and not normal
- Because of high labelling cost → unsupervised
- Visualize problematic areas of not normal hand

Data Provided

- Subset of MURA dataset
- Contains hand X-rays grayscale images
- Very noisy

- **Size**: 5543 images
- **Percentage of positive images**: 26%
- **Number of patients**: 1964
- Approx. 10% of images are **mislabeled**.




Agenda

- Task & Data
- Preprocessing workflow
- Method & Models training
- Visualizations
- Future work

Legends

✅ **Implemented**: The algorithm has been implemented and included in the repository
✅ **Tested**: It has been tested
✅ **Integrated**: It has been integrated into the pipeline and evaluated

✅ **Green**: Working Fine
⚠️ **Yellow**: No major success
❌ **Red**: Skipped this step

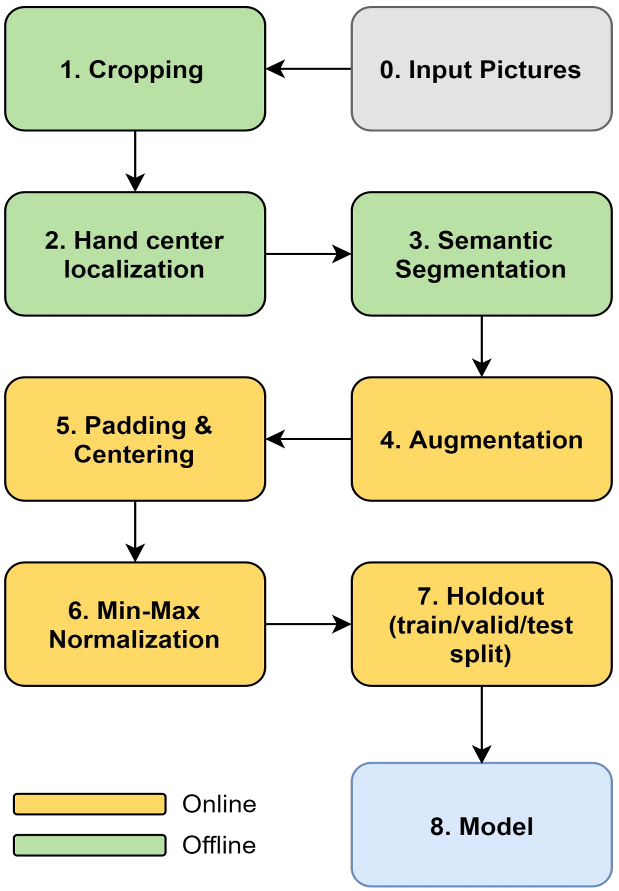
Workflow Followed

Preprocessing:

- **Cropping** using OpenCV
- **Object detection** using Tensorflow
- Semantic segmentation (Photoshop)

Learning and Prediction:

- 3 types of **Autoencoders**
- 3 types of **GANs**



Preprocessing: Square Detection

- **Idea**: Crop out X-ray-scans, synthetically created for MURA
- Based on OpenCV Contours Detection

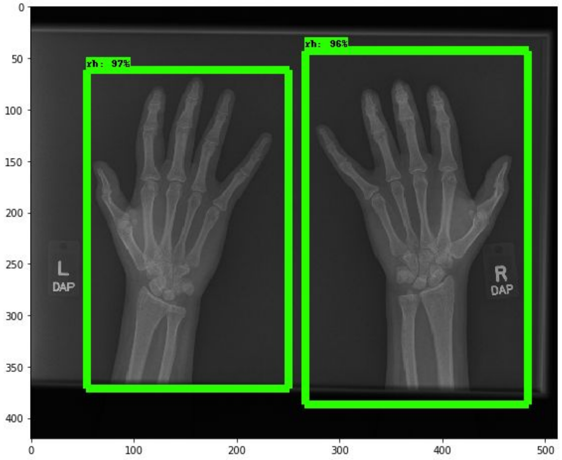
Frameworks Used:

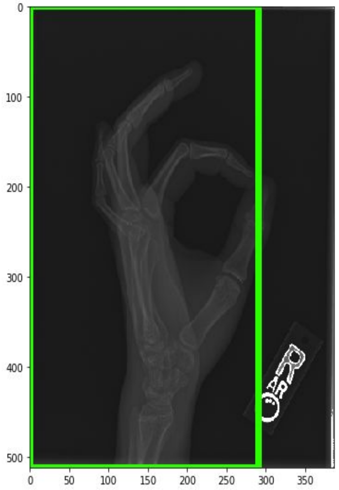
✅ OpenCV

✅ Implemented
✅ Tested
✅ Integrated

Preprocessing: Hand Detection & Cropping

- **Idea**: Detect hand and crop, output image has centered hand
- Based on **Single shot multibox detector** (SSD) with MobileNet
- Manually labeled bounding boxes for over 150 hands
- Fails for tilted images, not whole palms, two hands on one image
- Manually cropped all undetected images





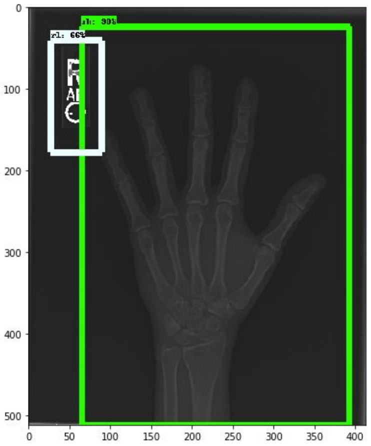
Frameworks Used:

✅ OpenCV
✅ Tensorflow object detection

✅ Implemented
✅ Tested
✅ Integrated

Preprocessing: Writings Detection & Removal

- **Idea**: Detect writings and remove by inpainting
- Analogous SSD method
- Manually labeled bounding boxes for over 100 labels
- Fails for writings, which is too close to hand, tilted writings



Frameworks Used:

✅ OpenCV
✅ Tensorflow object detection

✅ Implemented
⚠️ Tested
❌ Integrated

Preprocessing: Semantic Segmentation

- **Idea**: Segment hand and background, remove background
- DL Semantic segmentation requires labeled masks
- Tried with **GrabCut**, but it is hard to adapt it for all images
- We used Photoshop **‘Select Subject’** option
- Still not perfect solution

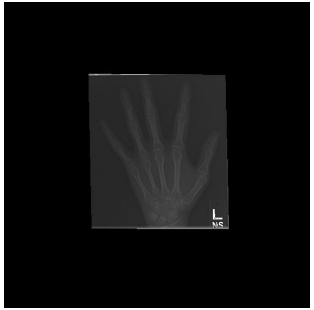
Frameworks Used:

✅ Photoshop batch processing




⚠️ Implemented
✅ Tested
✅ Integrated

Preprocessing: Augmentation



Frameworks Used:

✅ OpenCV
✅ imgaug

Augmentation techniques:
- Flipping
- Rotating
- Brightness adjustment
- Zoom in/out

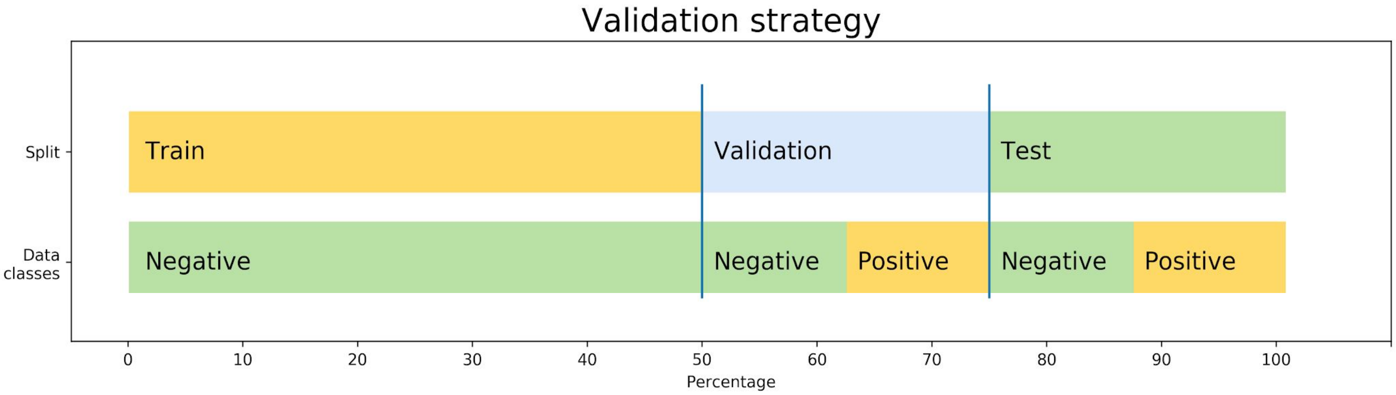
✅ Implemented
✅ Tested
✅ Integrated

Processing: Data-split

- Unsupervised fashion means that we use only normal images in a training phase
- Still need to know, that images are normal
- Train is only 50% - to make test and validation balanced

Frameworks Used:

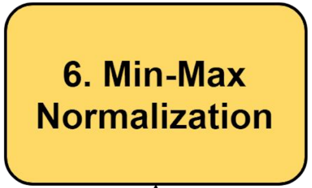
✅ Scikit-learn

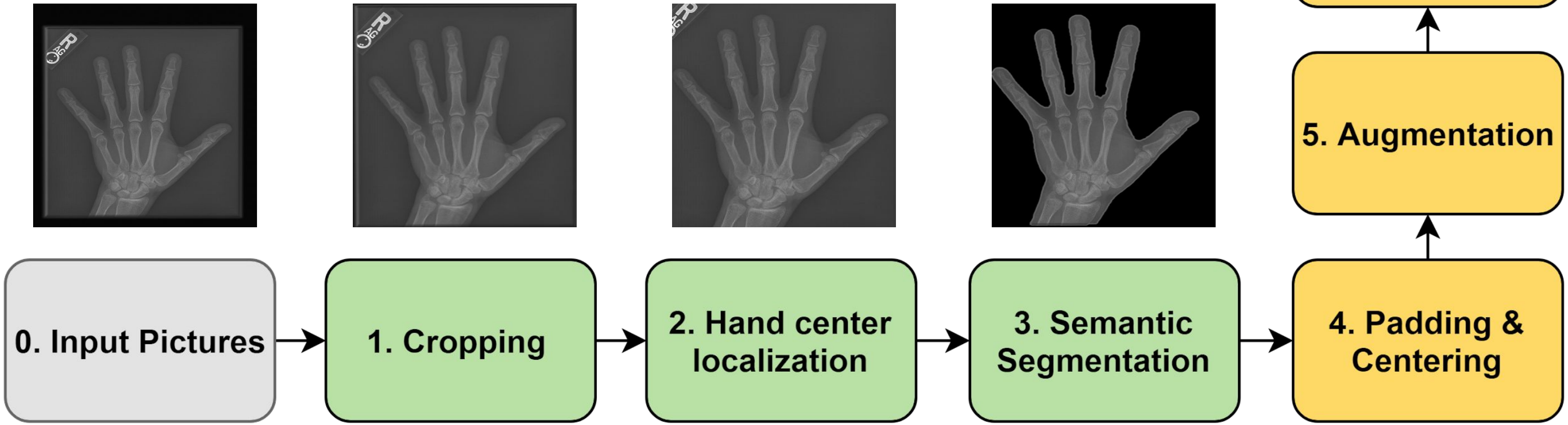


✅ Implemented
✅ Tested
✅ Integrated

Preprocessing Pipeline: Altogether

- **Centered Padding** converts all images to 512x512 shape (also tried uniform padding)
- **Min-Max Normalisation** scales all pixels to [0, 1]





Agenda

- Task & Data
- Preprocessing workflow
- Method & Models training
- Visualizations
- Future work

Method: Outlier Detection

We discriminate between **normal** and **abnormal** cases using the following statistics (outlier scores):

- Reconstruction loss (CAE, BiGAN, Alpha-GAN)
- Reconstruction loss + Kullback-Leibler divergence (VAE)
- Discriminator output probability (DCGAN, BiGAN, Alpha-GAN)
- Latent features outlier detection:
    - One-class SVM (CAE)
    - DBSCAN (CAE)

Training: Overview

- **Unsupervised Learning:**
  - Non-negative matrix factorisation + DBSCAN / One-Class SVM
- **Unsupervised Deep Learning:**
  - Deep One-Class Classification
  - Convolutional Autoencoder (CAE)
  - Variational Convolutional Autoencoder (VAE)
  - Deep Convolutional GAN (DCGAN)
  - Bidirectional GAN (BiGAN)
  - Alpha-GAN (GAN + VAE)

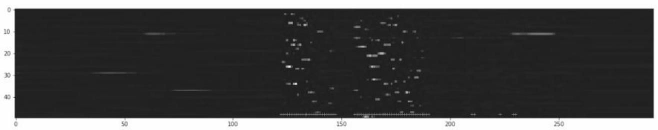
Training: Static Methods

Non Negative Matrix Factorisation:

An image V is factorized in W and H matrices

$$V(m \times n) = W(m \times p) \bullet H(p \times n)$$

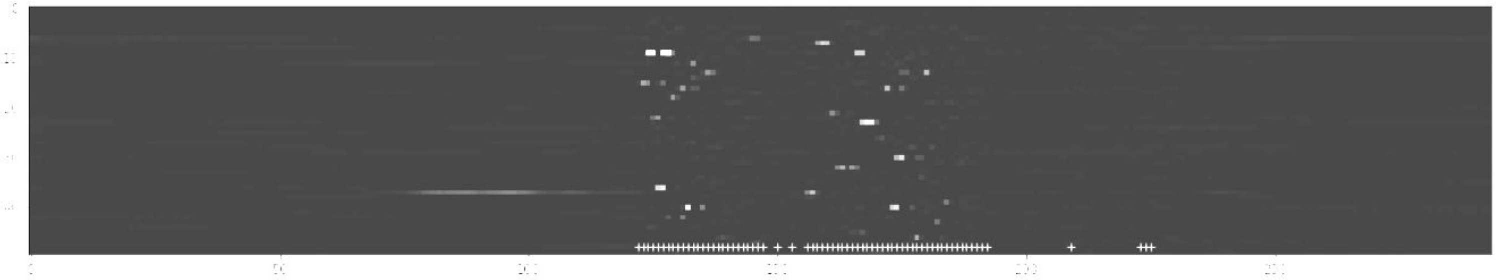
V



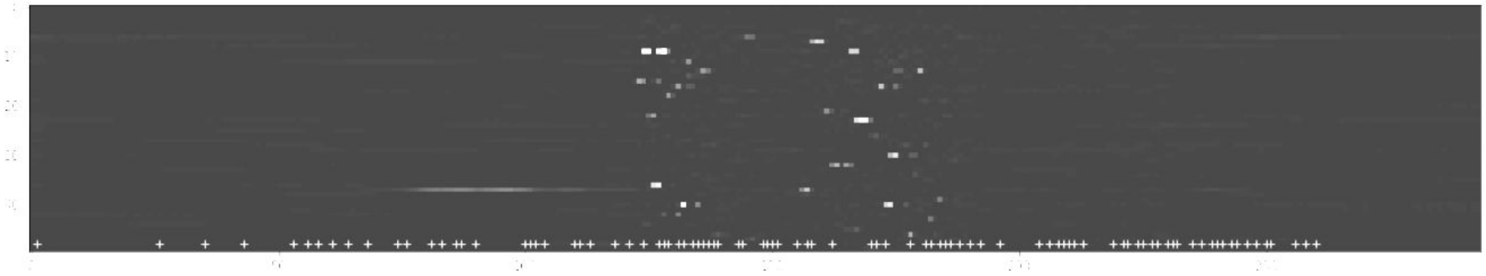
W and H looks like this

V = W • H

Training: Static Methods



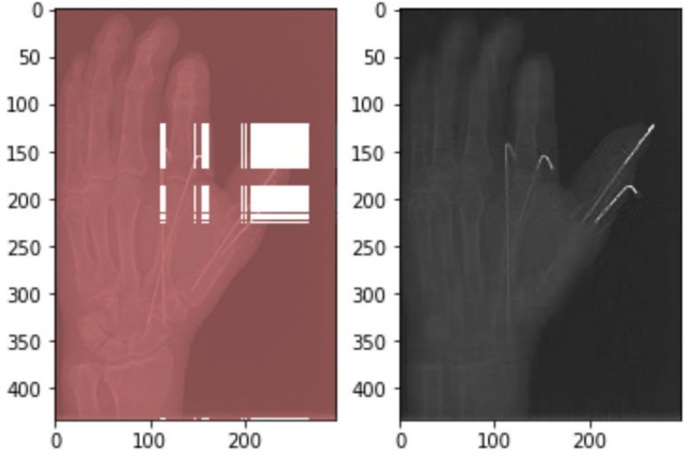
DBScan

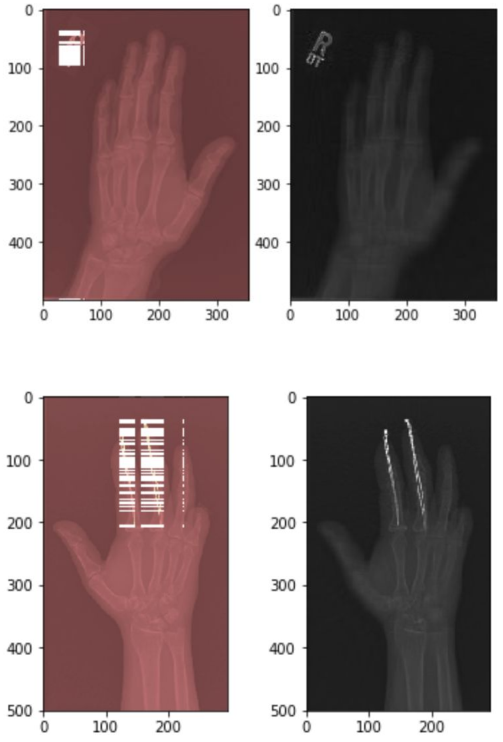


One Class SVM

Training: Static Methods

Reconstruction





Frameworks Used:

✅ OpenCV
✅ Scikit-learn

✅ Implemented
⚠️ Tested
❌ Integrated

Training: Deep One-Class Classification

- Predicts **score** for an image
- **Higher the score, higher the abnormality**
- Completely unsupervised
- Not a single label needed for training
- It however **requires an outlier classes**
- For example, to train the model for hand images, we need images of non-hands, like legs, chest, etc, which is fairy easy and cheap to obtain

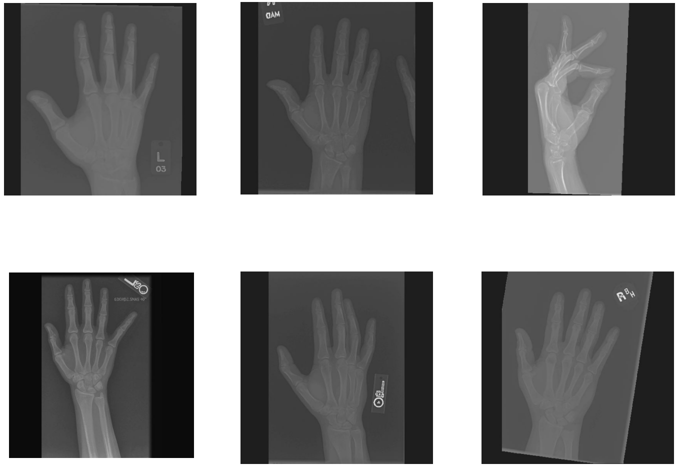
Frameworks Used:

✅ OpenCV
✅ Tensorflow

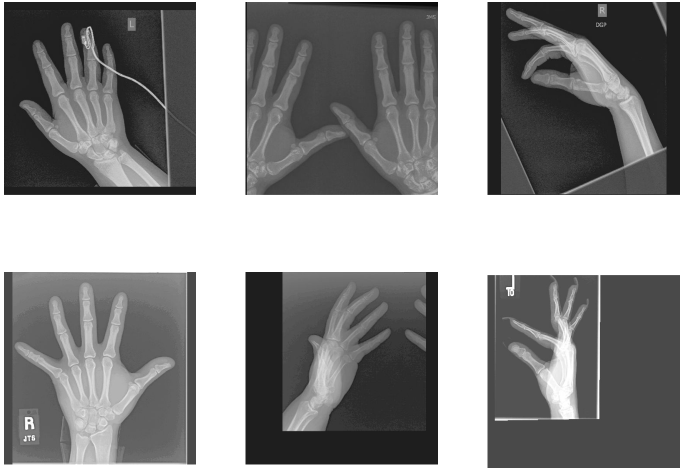
✅ Implemented
⚠️ Tested
⚠️ Integrated

Training: Deep One-Class Classification

Low Score Samples:



High Score Samples:



Training: Convolutional Autoencoder

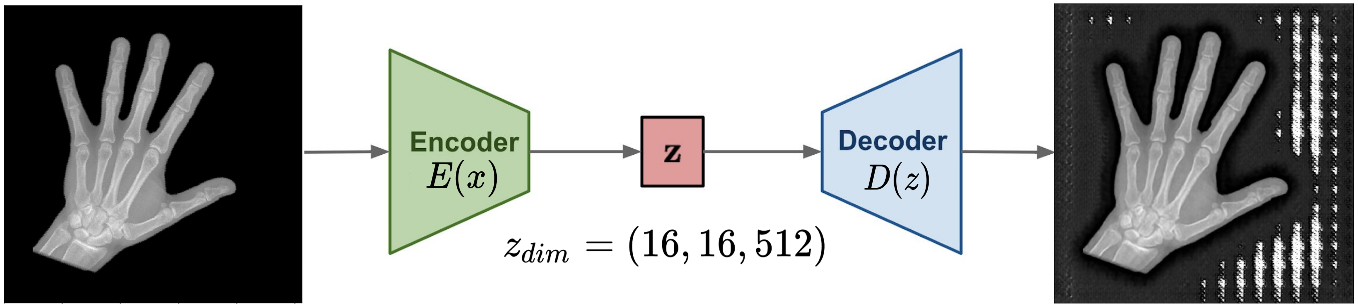
Loss: Masked Reconstruction MSE (calculated only on non-zero parts)

Outlier score: Masked MSE / Top-K SE

Best ROC-AUC: 0.58

Frameworks Used:

✅ OpenCV
✅ PyTorch



$x_{dim} = (512, 512, 1)$ $x_{dim} = (512, 512, 1)$

✅ Implemented
✅ Tested
✅ Integrated

Training: Bottleneck Convolutional Autoencoder

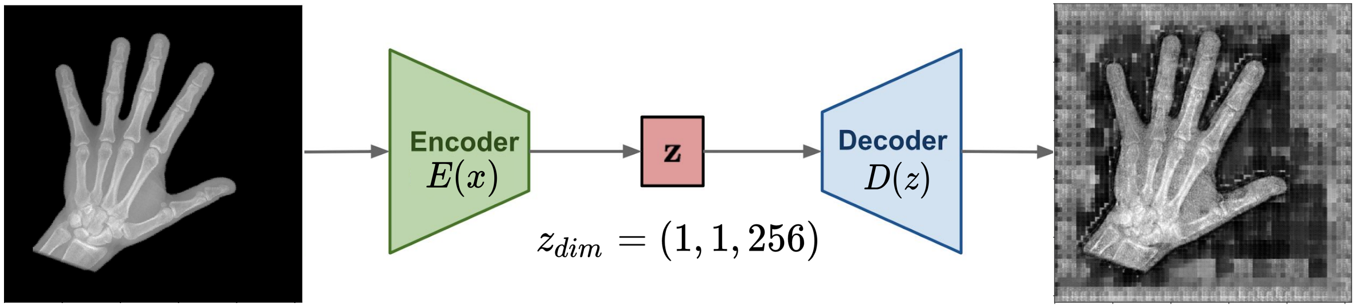
Loss: Masked Reconstruction MSE (calculated only on non-zero parts)

Outlier score: Masked MSE / Top-K SE

Best ROC-AUC: 0.57

Frameworks Used:

✅ OpenCV
✅ PyTorch

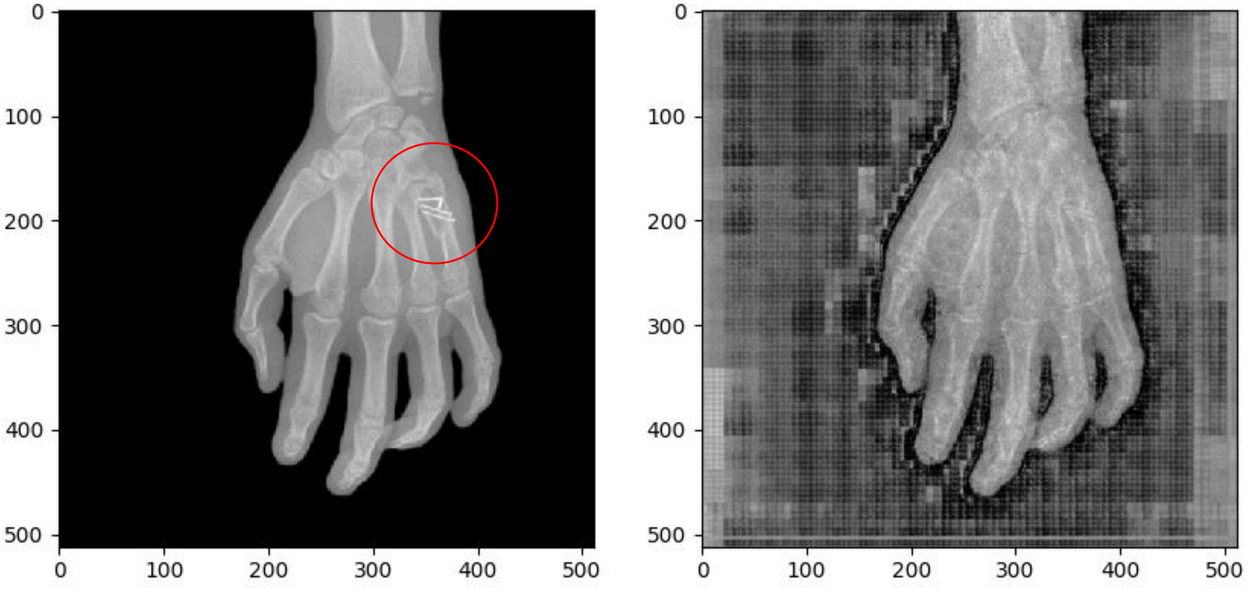


$x_{dim} = (512, 512, 1)$ $x_{dim} = (512, 512, 1)$

✅ Implemented
✅ Tested
✅ Integrated

Training: Bottleneck Convolutional Autoencoder

Epoch 73: Not reconstructing extrinsic objects → A way to detect them



Training: Variational Autoencoder

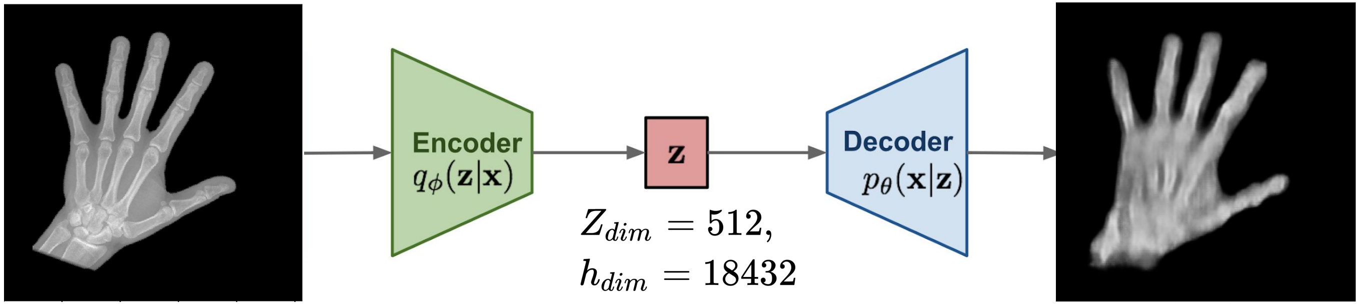
Loss: Binary Cross Entropy (Pixelwise) + Kullback Leibler Divergence (Latent features)

Outlier score: Loss, used for training

Best ROC-AUC: 0.53

Frameworks Used:

✅ OpenCV
✅ PyTorch



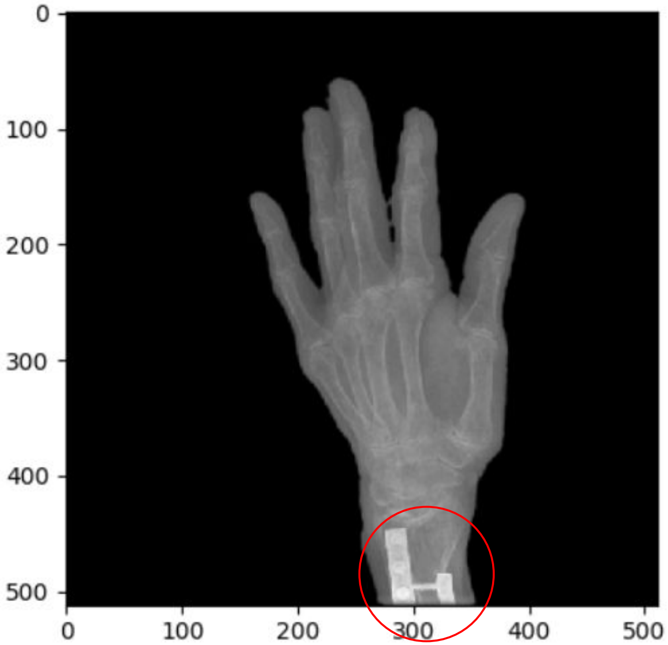
$x_{dim} = (512, 512, 1)$ $x_{dim} = (512, 512, 1)$

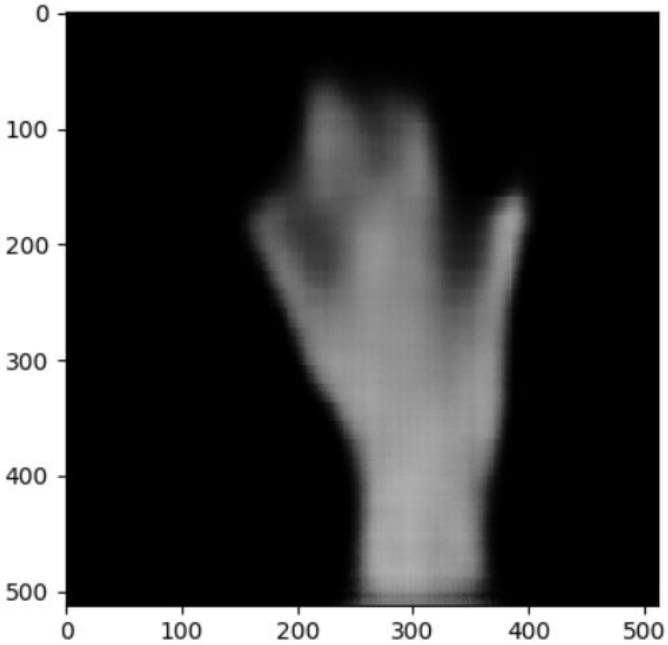
Epoch 600

✅ Implemented
✅ Tested
✅ Integrated

Training: Variational Autoencoder

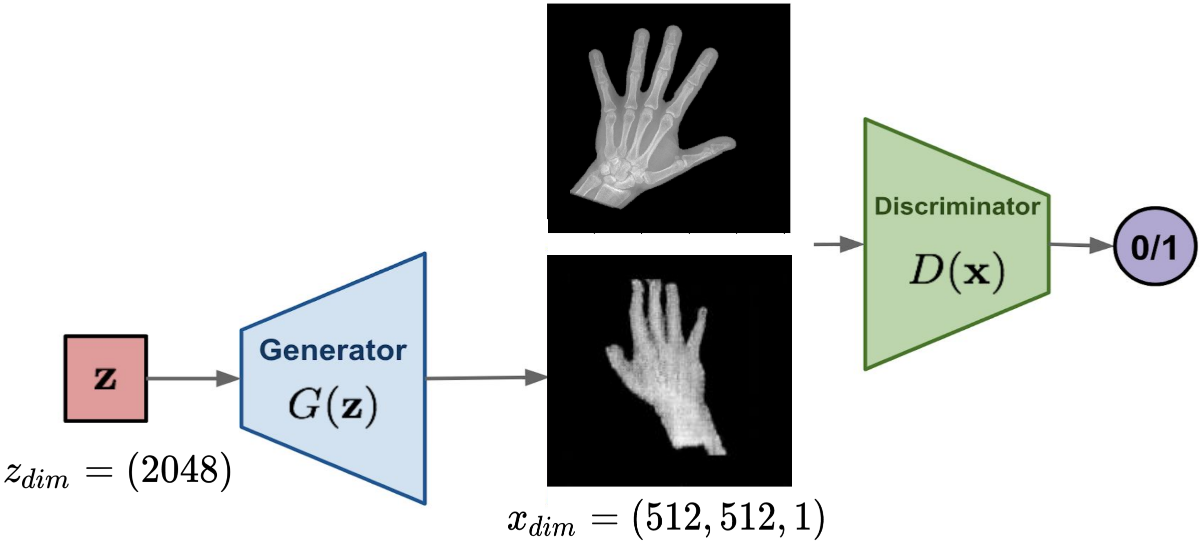
Epoch 17: Not reconstructing obvious anomalies → A way to detect them





Training: Deep Convolutional GAN

Loss: Binary Cross Entropy between fake and real images
Outlier score: Discriminator output probability
Best ROC-AUC: 0.57



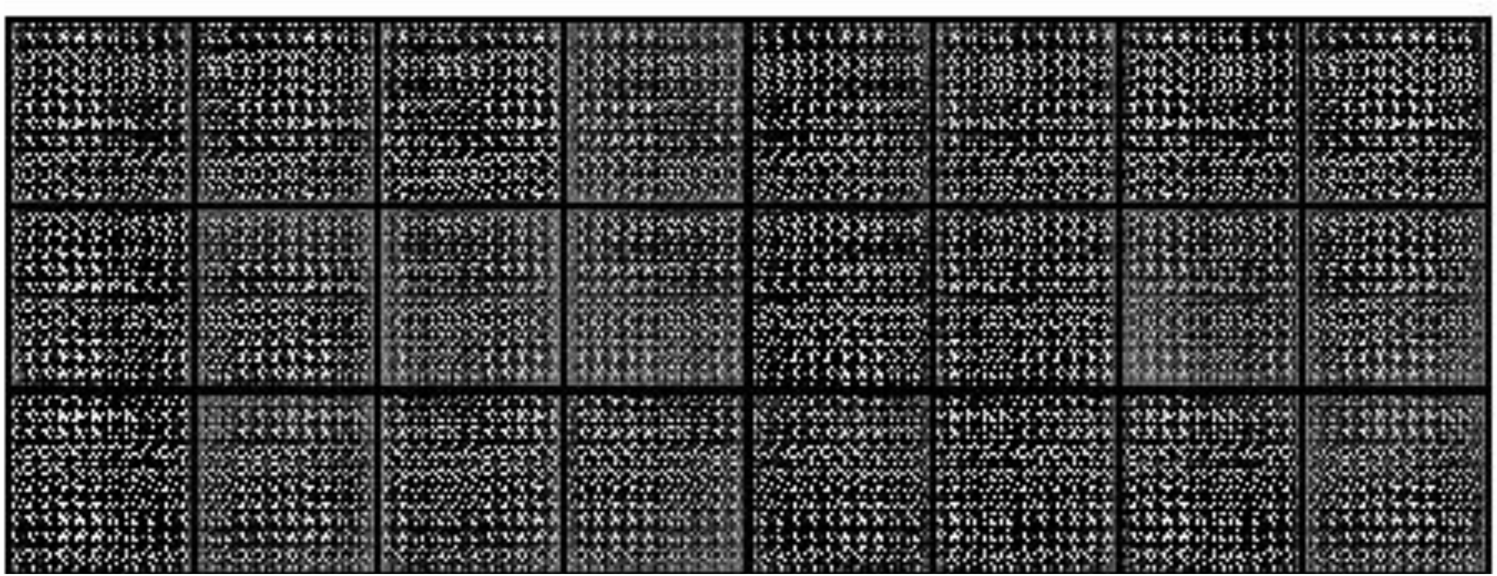
Frameworks Used:

✅ OpenCV
✅ PyTorch

✅ Implemented
✅ Tested
✅ Integrated

Training: Deep Convolutional GAN

- Hard to choose learning rates
- Often - mode collapse (sacrifice diversity on accuracy)



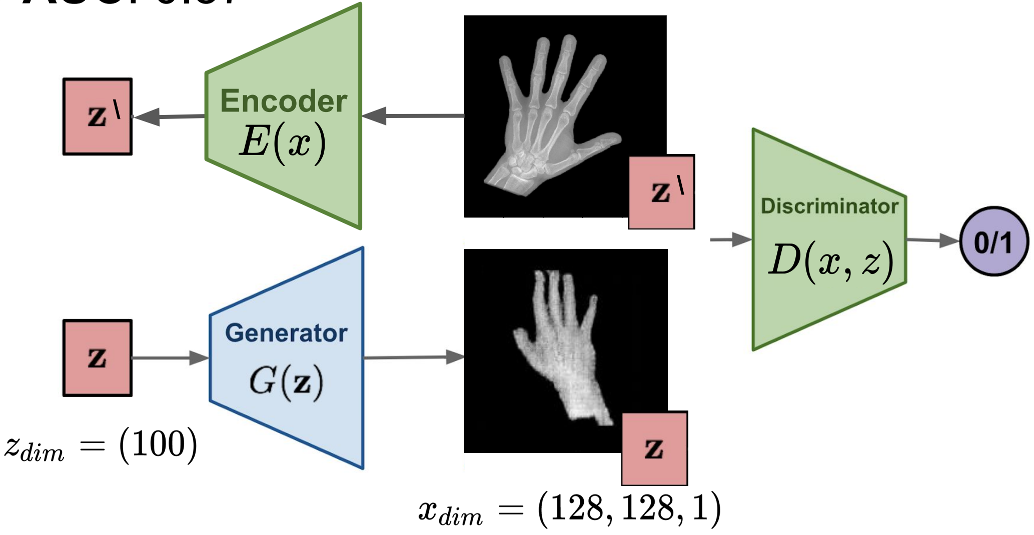
Training: Bidirectional GAN

Loss: Binary Cross Entropy between fake and real images

Added self-attention layer

Outlier score: Discriminator output probability / Masked MSE

Best ROC-AUC: 0.57



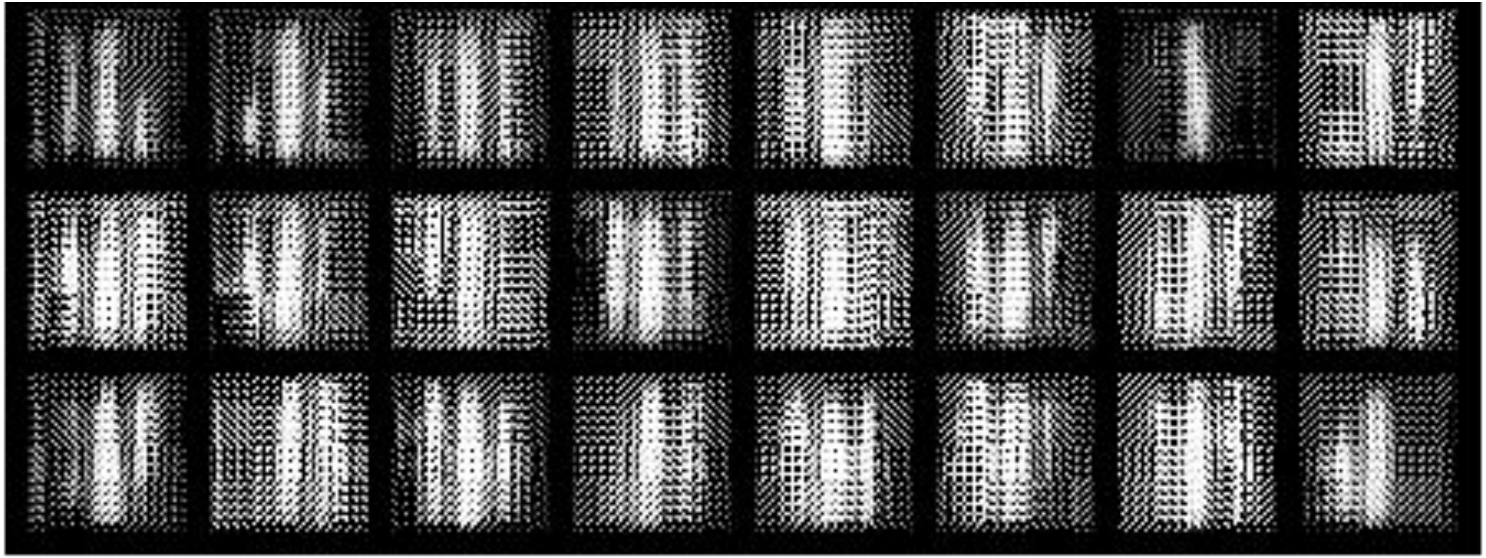
Frameworks Used:

✅ OpenCV
✅ PyTorch

✅ Implemented
✅ Tested
✅ Integrated

Training: Bidirectional GAN

- Better in reconstruction
- Often - mode collapse (sacrifice diversity on accuracy)



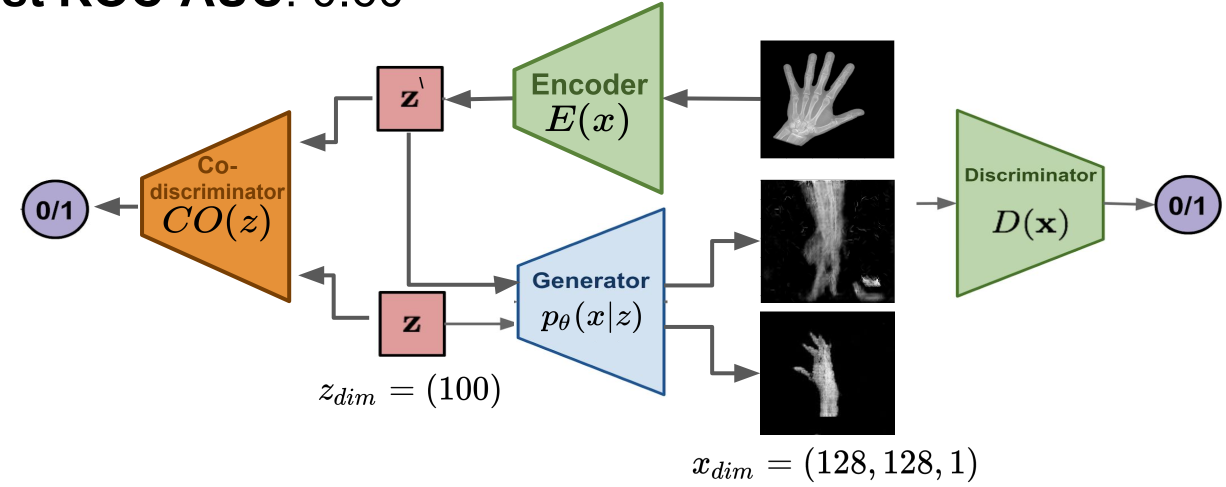
Training: Alpha-GAN

Loss: Binary Cross Entropy between fake/reconstructed and real images + Kullback Leibler Divergence (Latent features)

Added self-attention layers

Outlier score: Discriminator output probability / Masked MSE

Best ROC-AUC: 0.60



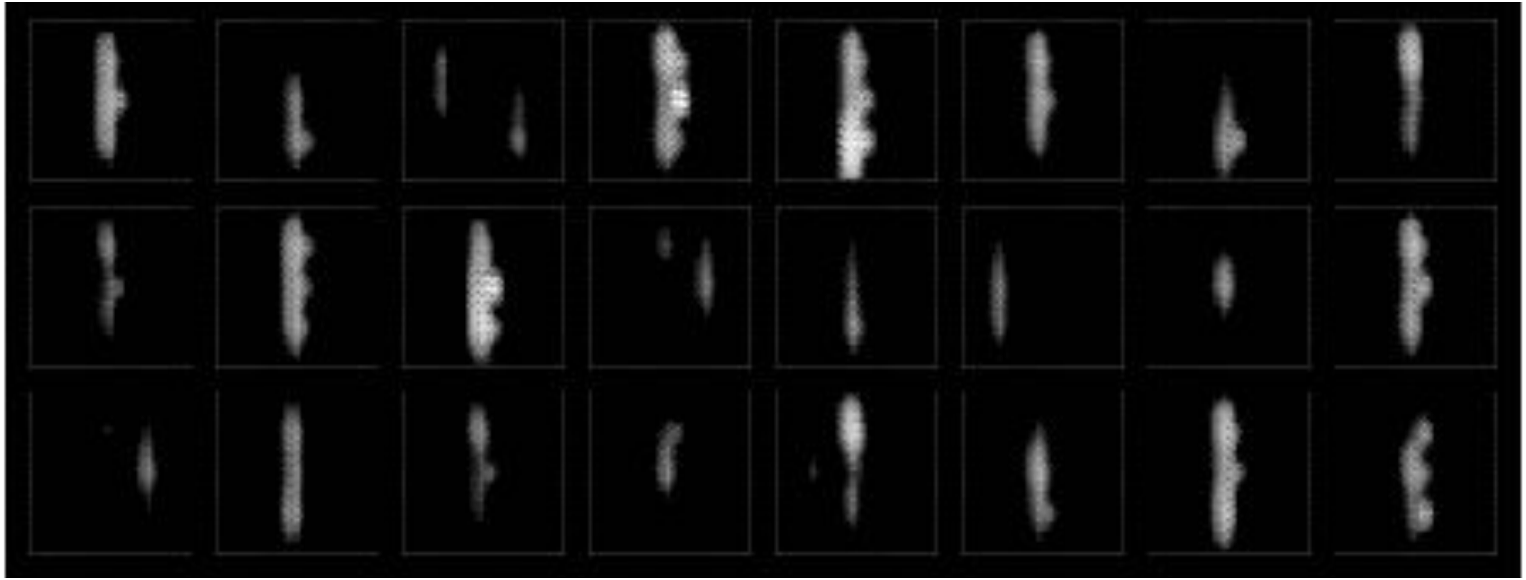
Frameworks Used:

- ✅ OpenCV
- ✅ PyTorch

- ✅ Implemented
- ✅ Tested
- ✅ Integrated

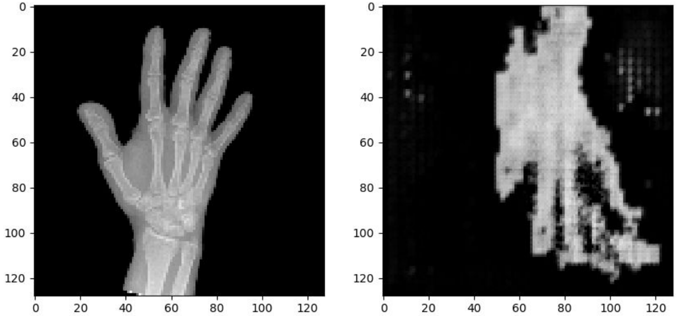
Training: Alpha-GAN

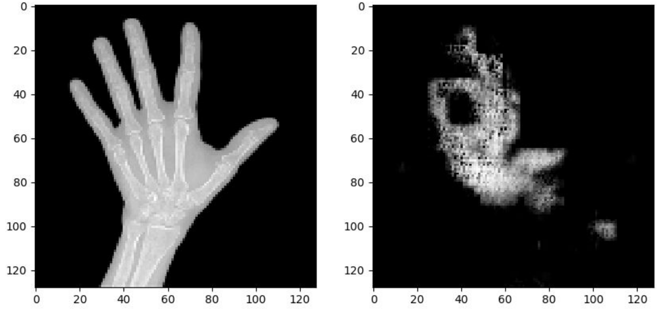
- Worse in reconstruction (hard to choose learning rate)
- Generator outputs more diverse images

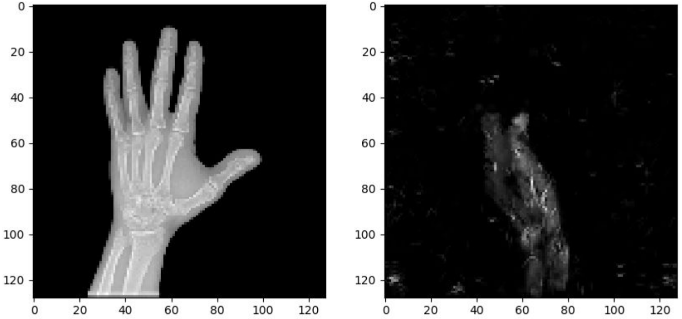


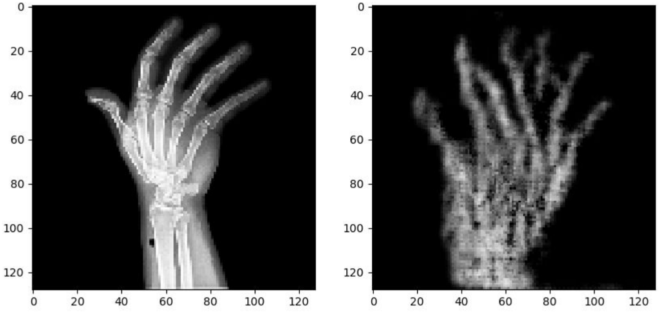
Training: BiGAN & Alpha-GAN

- Main problem: not diverse generator → hard to use RMSE for outlier score









Comparison of Autoencoders and GANS

	Unmasked Images	Masked Images	
Model	ROC-AUC	ROC-AUC	APS
CAE (z_{dim} = (512, 16, 16))	0.45	0.58	0.58
CAE (z_{dim} = (256, 1, 1))	0.44	0.57	0.58
VAE	0.50	0.53	0.57
DCGAN	**0.57**	0.56	0.63
BiGAN	-	0.57	**0.66**
Alpha-GAN	-	**0.60**	**0.66**

Agenda

- Task & Data
- Preprocessing workflow
- Method & Models training
- Visualizations
- Future work

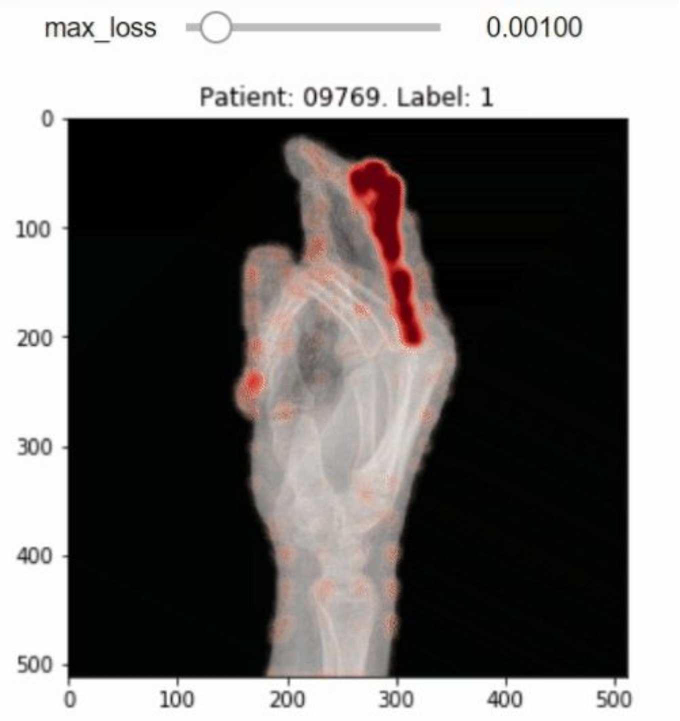
Pixelwise Loss for Visualisation: Abnormal hands

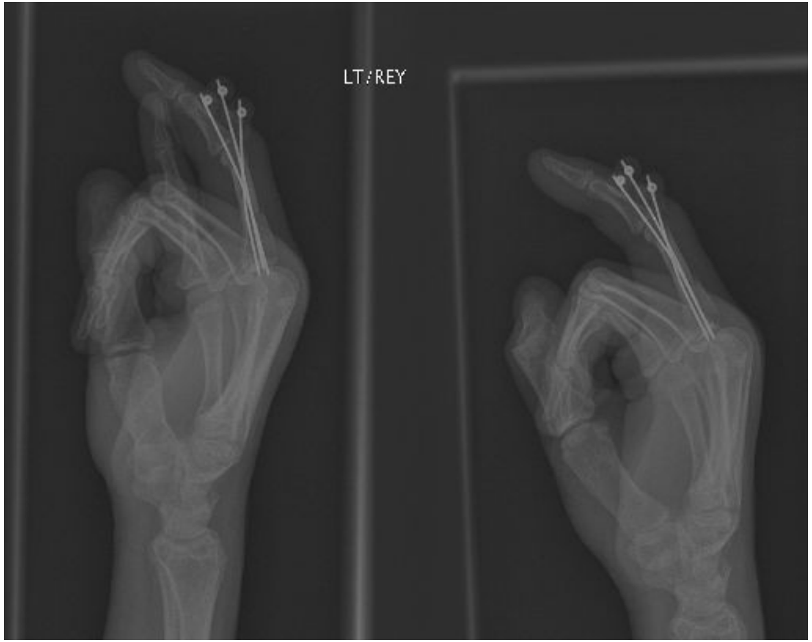
- Possible for Convolutional Autoencoders



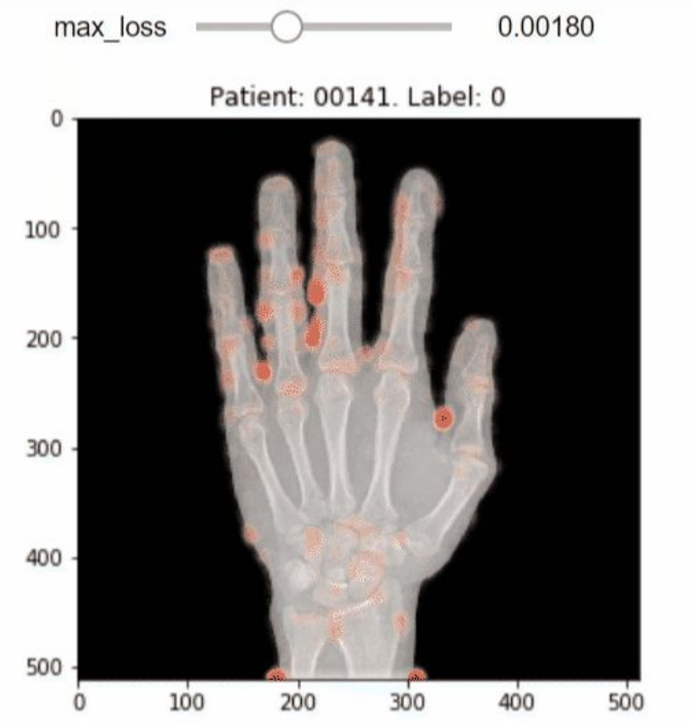
Pixelwise Loss for Visualisation: Normal Hand

Pixelwise Loss for Visualisation: Abnormal hand





Pixelwise Loss for Visualisation: Normal Hand



Agenda

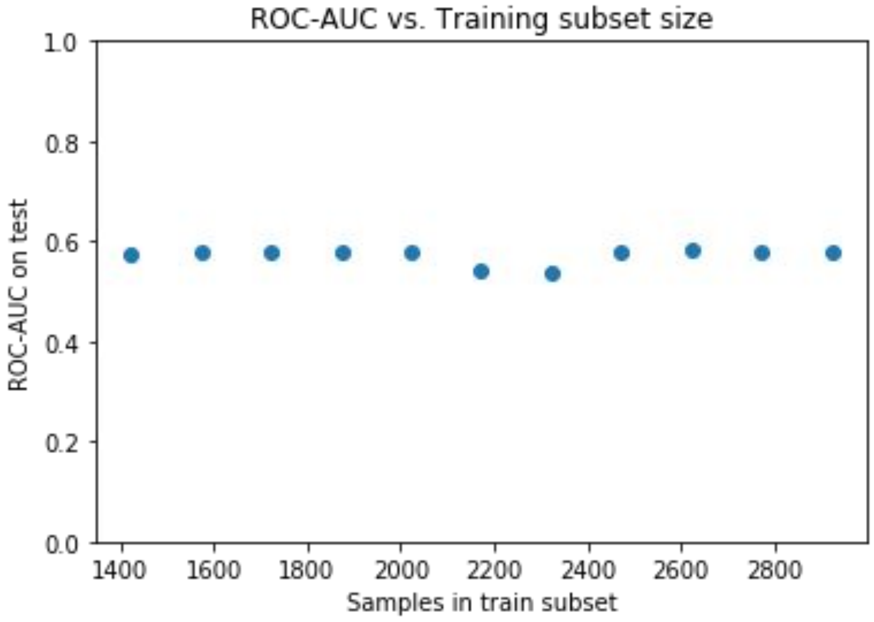
- Task & Data
- Preprocessing workflow
- Method & Models training
- Visualizations
- Future work

Extension Possibilities

- Flexible and generic architecture
- Can be extended for other body parts with slight modifications
- For example:
    - Object detection model can be extended to detect other parts and redirect the flow to corresponding neural network
    - New autoencoders can be trained and saved to find anomalies in other type of data sets

Extension Possibilities: Quantity vs. Quality

- Extension of training subset does not increase performance
- **Quality matters:** need to develop more sophisticated data-cleaning pipeline



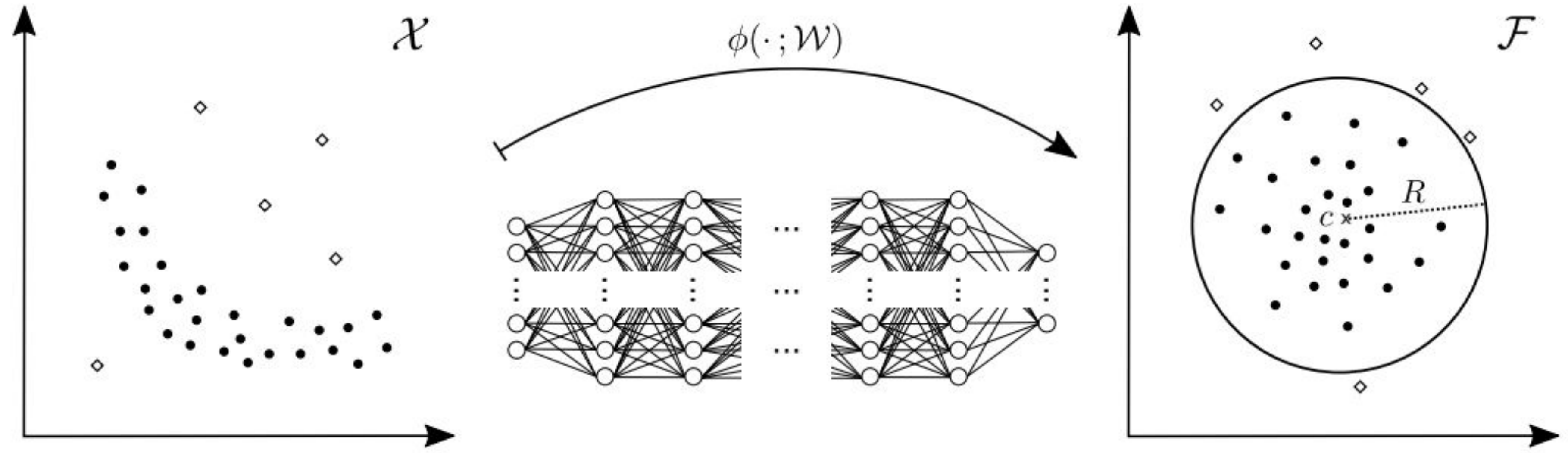
Future Work

- Evaluate the usage of **flow-based models**
- Visualize **attention maps** for Self-Attentive GANs
- **Image pipeline improvements:**
  - Few-shot semantic segmentation for hand masking
  - Tuning of hand detector
  - Median filters evaluation
- **Models hyperparameters tuning:**
  - Learning rates adjustments, learning rate schedulers
  - Model architecture tuning

Thank You for the attention

Feedback and Questions?

Appendix: Deep One Class Classification



Appendix: Finding anomaly in encoded representation

For the initial image one the left
Encoded representation somewhat
looks like the image below.

+ represents the possible outliers as
detected by One-Class SVM algorithm

Frameworks Used:

✅ OpenCV
✅ Tensorflow
✅ Scikit

⚠️ Implemented
❌ Tested
❌ Integrated