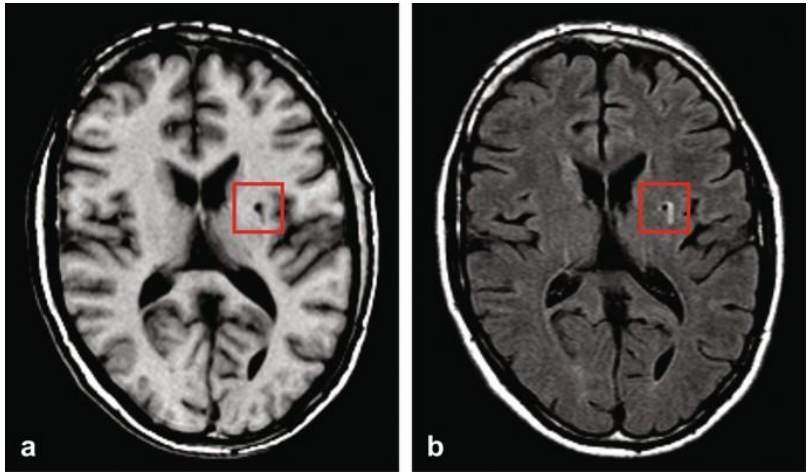
Unsupervised Anomaly Detection for Medical Images

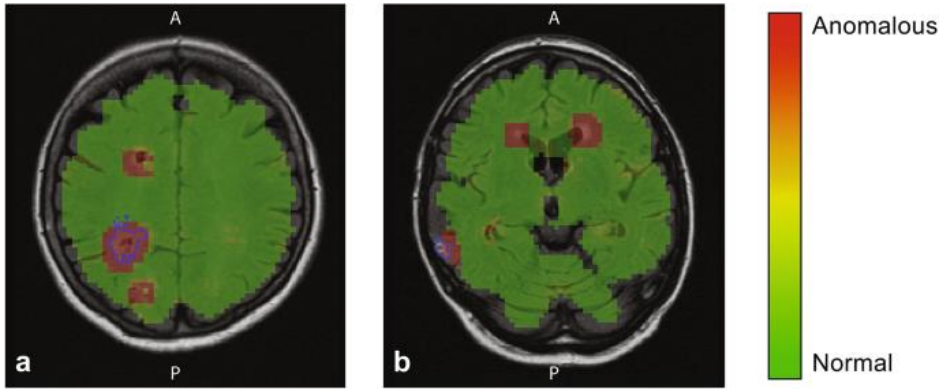
CT Images

1. Introduction

1.1 Background

- **Anomaly Detection**: A process of identifying data points or patterns that deviate significantly from the norm or expected behavior, often indicating potential errors, outliers, or rare events.
- **Importance of detecting anomalies in data**:
  - Early identification of potential problems or issues
  - Enhancing data quality and accuracy
  - Improving decision-making based on data insights
  - Detection of fraud, system failures, or security breaches
- **Types of Anomaly Detection**:
  - **Supervised Anomaly Detection**: Uses labeled data (normal and anomalous instances) to train a model that can classify future instances as normal or anomalous.
  - **Unsupervised Anomaly Detection**: Does not require labeled data, instead relying on the intrinsic structure of the data to identify anomalies. It is particularly useful when labeled data is scarce or costly to obtain.
  - **Semi-supervised Anomaly Detection**: Combines elements of both supervised and unsupervised methods, using a small amount of labeled data to guide the identification of anomalies in the larger, unlabeled dataset.

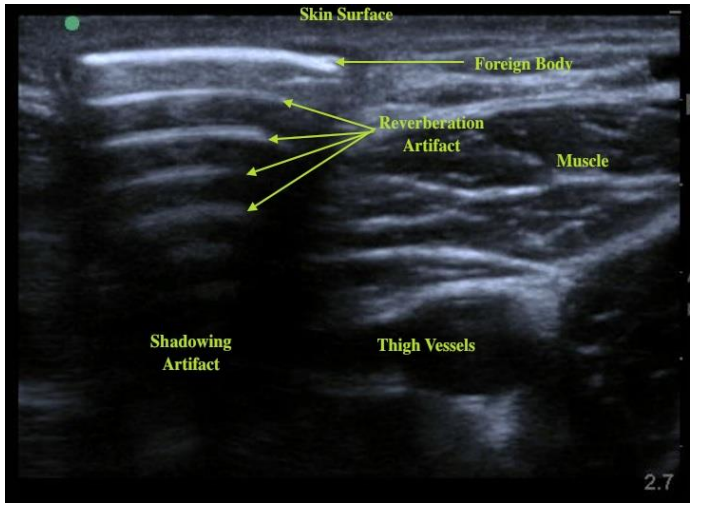




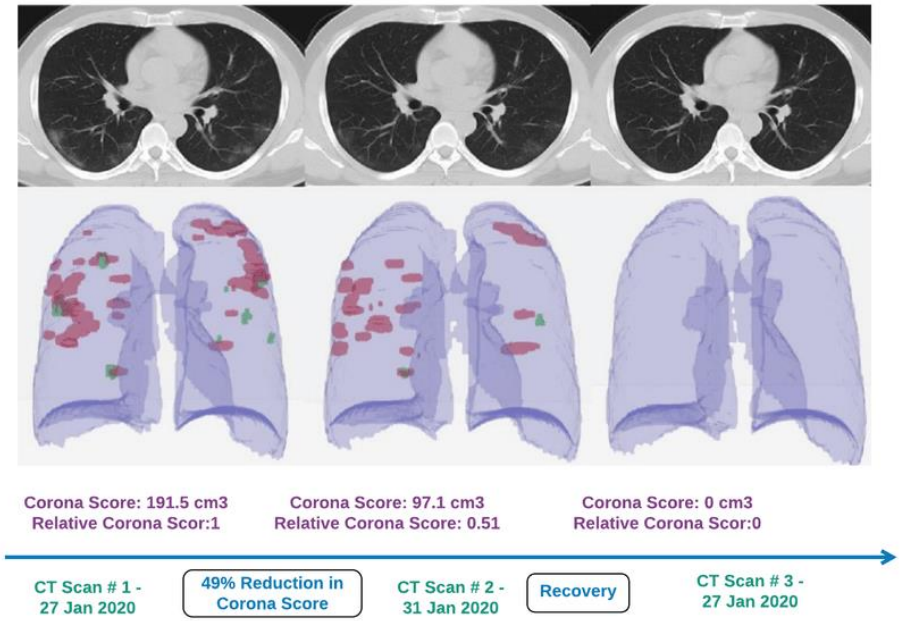
van Hespen, K.M., Zwanenburg, J.J.M., Dankbaar, J.W. *et al.* An anomaly detection approach to identify chronic brain infarcts on MRI. *Sci Rep* **11**, 7714 (2021). https://doi.org/10.1038/s41598-021-87013-4

1.2 Applications

- Early disease detection
- Quality Assurance and artifacts identification
- Assisting in diagnosis and treatment planning
- Monitoring disease progression
- Identifying rare diseases or conditions



Artifacts Identification



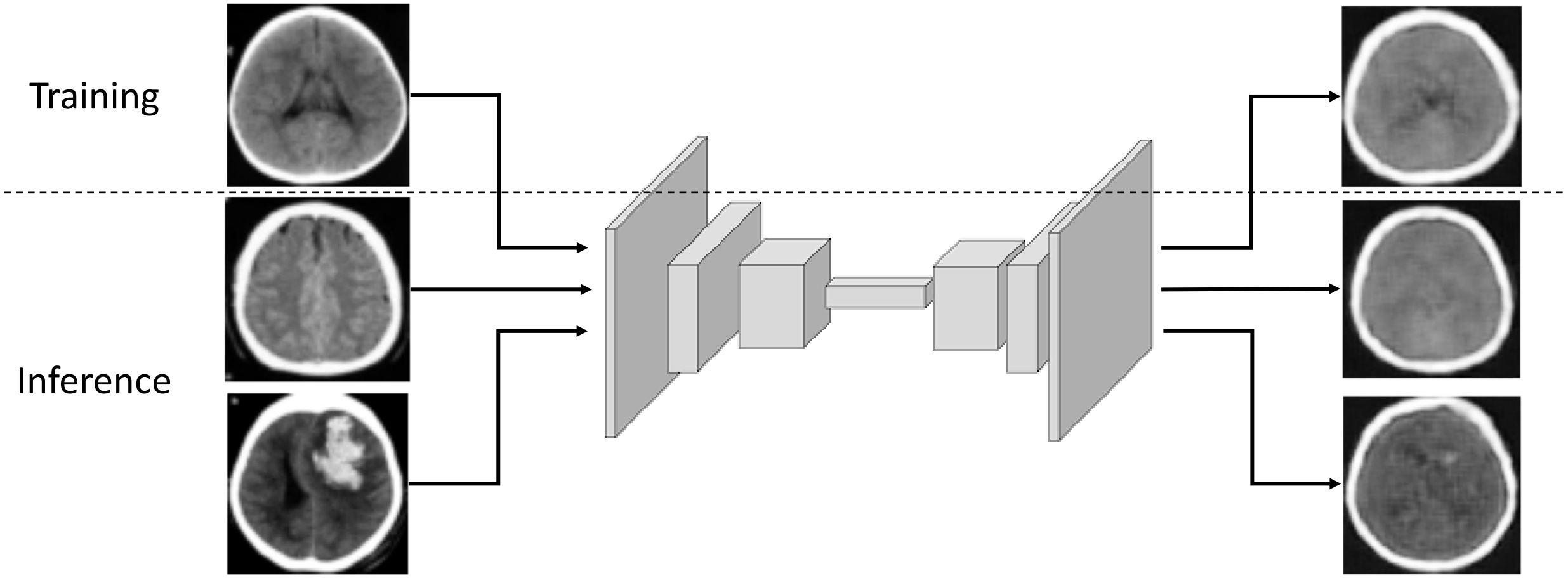
Disease Progression

1.3 Significance

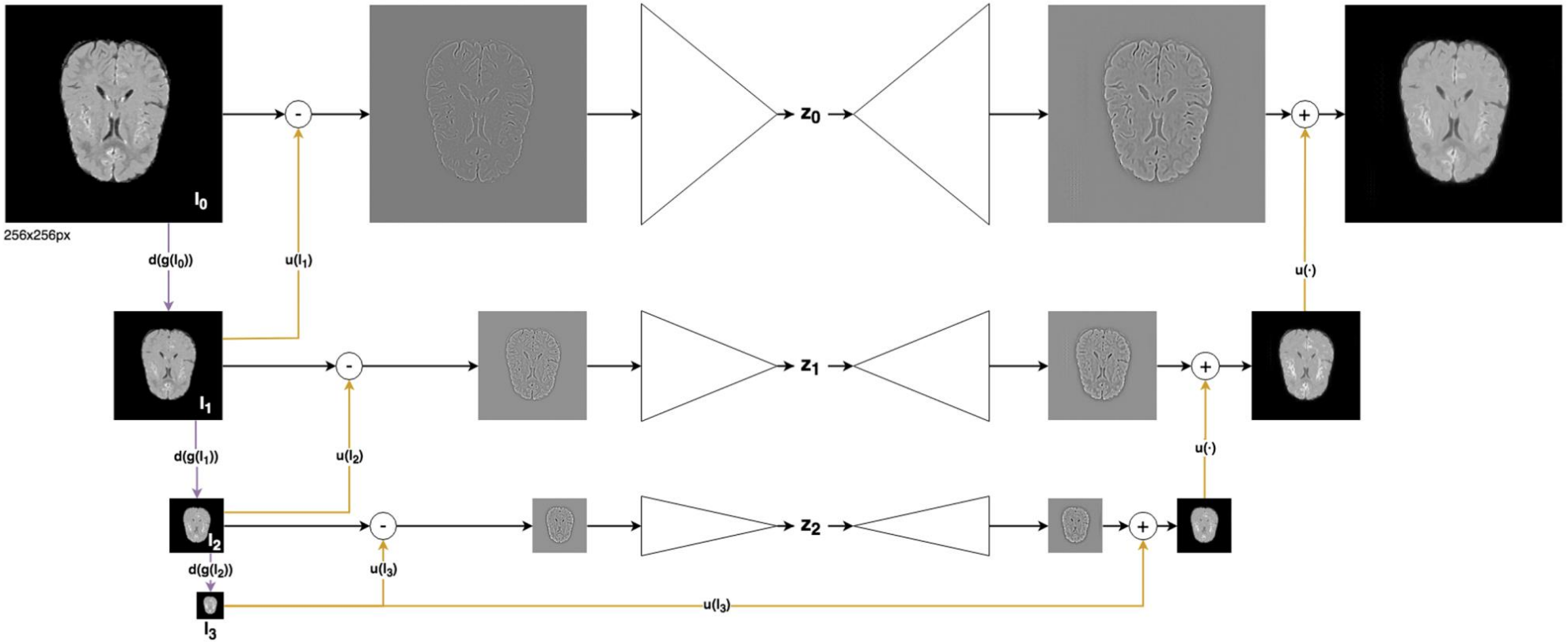
- Improving patient outcomes
- Reducing the burden on healthcare professionals
- Enhancing the efficiency of healthcare systems
- Facilitating early intervention and personalized treatment

2. Methodology

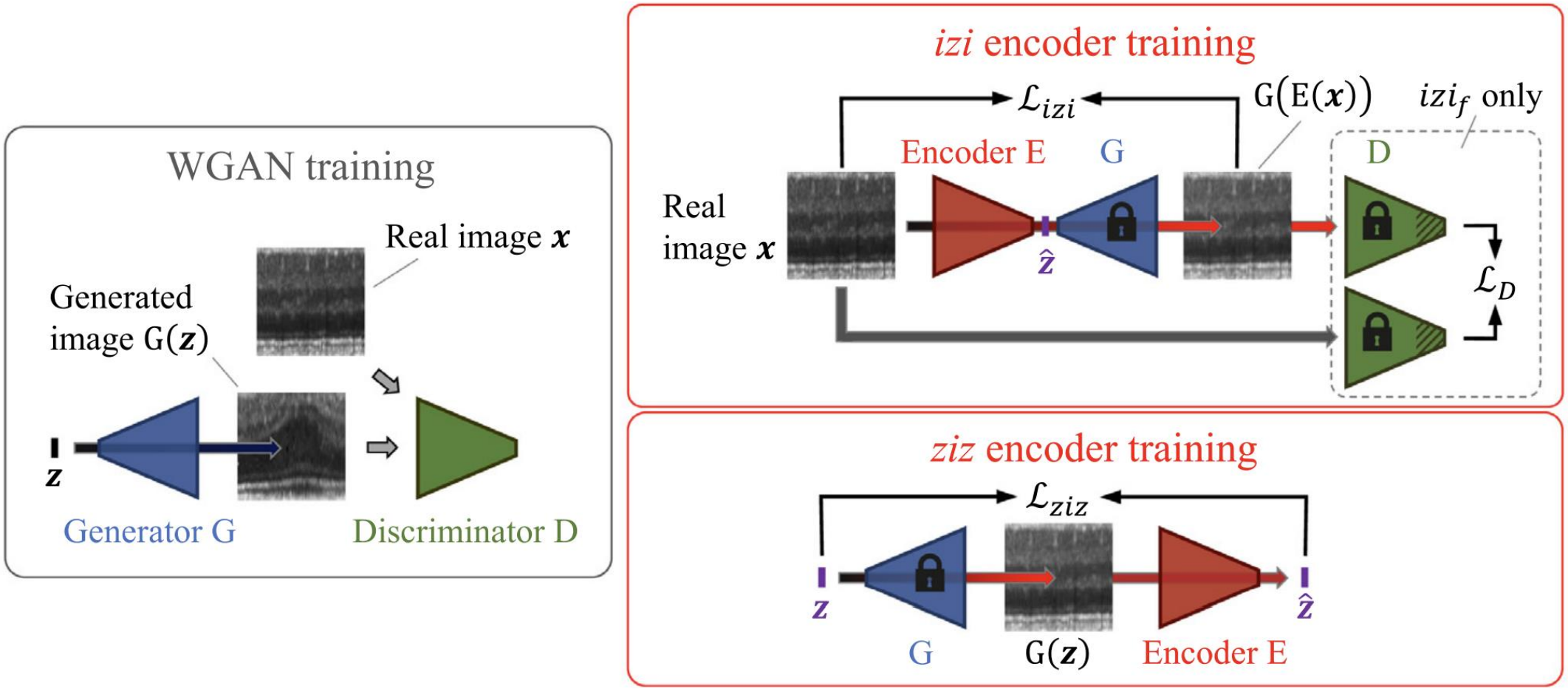
2.1 Convolutional Autoencoder (CAE)



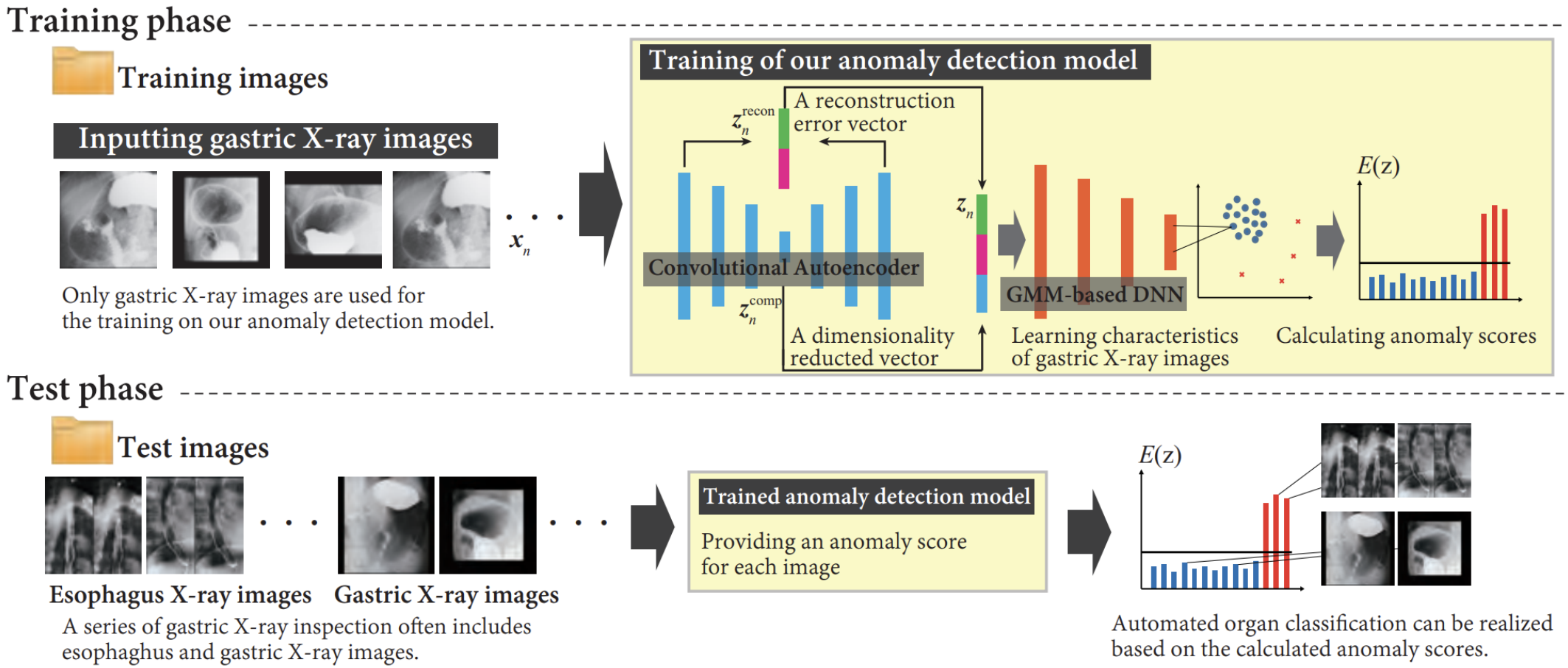
2.2 Scale Space Autoencoder (SSAE)



2.3 Fast Anoamly Detection GAN (f-AnoGAN)



2.4 Deep Autoencoding Gaussian Mixture Model (DAGMM)

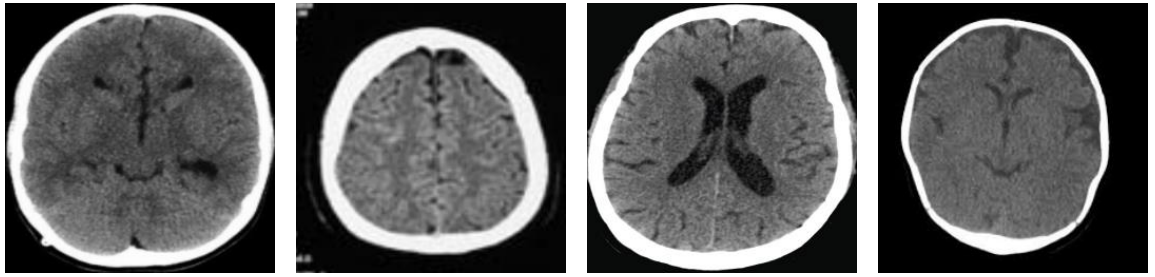


3. Experiments

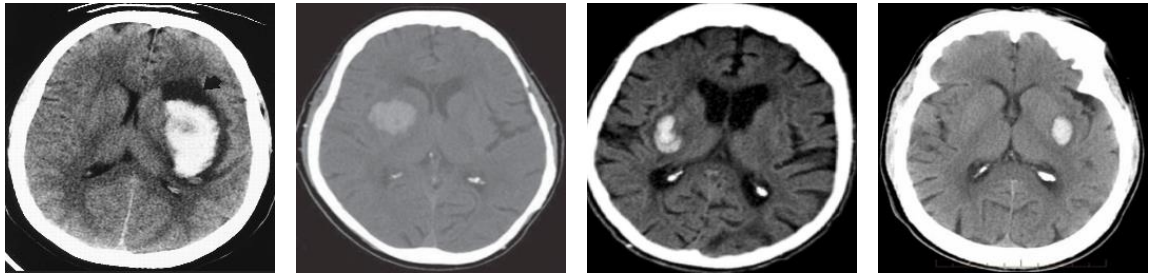
3.1 Dataset

- Images: 2D head CT scans
- Two class:
  - Normal: 100 images
  - Abnormal: 100 images
- Image size: 1x256x256

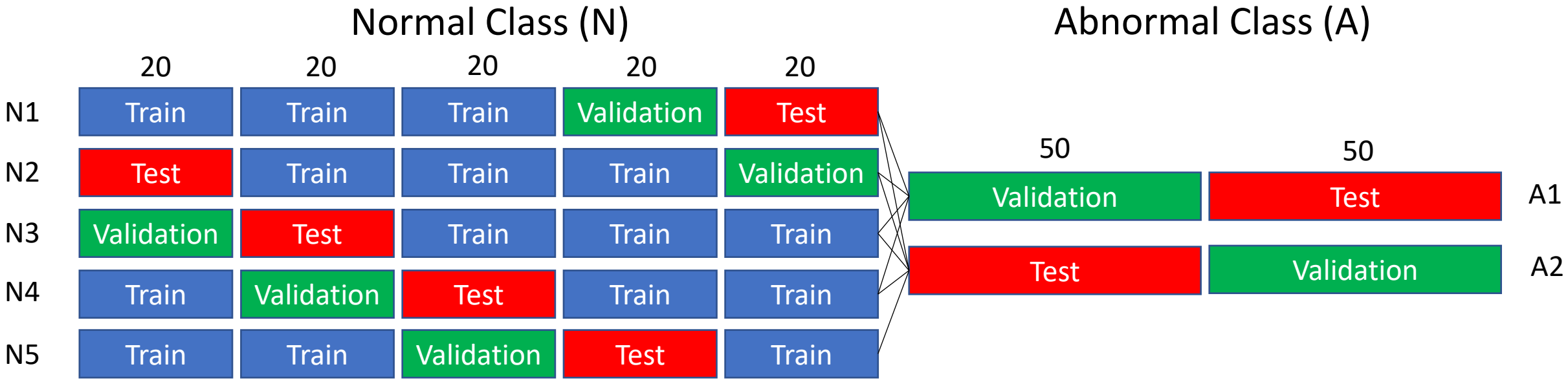
Normal



Abnormal



3.2 Cross Validation Experiment



- 2x5 = 10 iterations
- Model with best validation AUC is evaluated on test set
- Threshold is determined using f1-score on validation set
- Final average performance on test set is reported

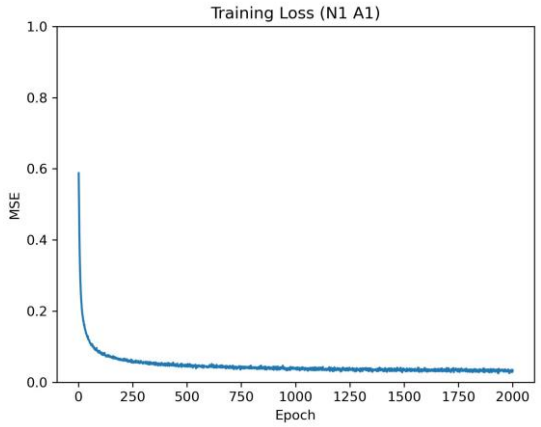
3.3 Implementation Details

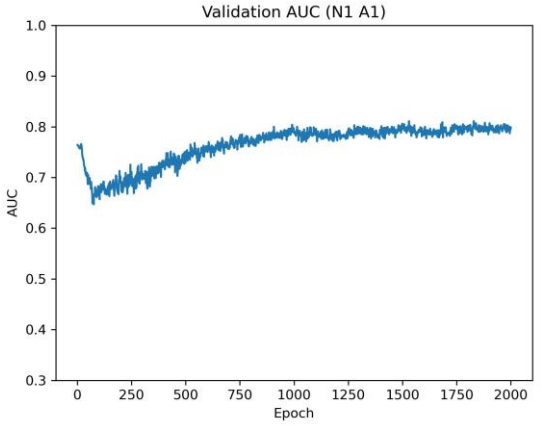
- Input images:
  - Resized to: 64x64
  - Intensity rescaled to: [0, 1]
- Data augmentation:
  - Random horizontal flip
  - Random affine: rotation, translation, scaling, shearing
  - Color jitter: brightness, contrast, satuation, hue
- Convolutional blocks:
  - Encoder block: Convolution + Batch Normalization + Leaky ReLU + Dropout
  - Decoder block: Transpose Conv + Batch Normalization + Leaky ReLU + Dropout
- Model training:
  - Batch size: 30
  - Optimizer: Adam
  - Automatic mixed precision to speed up training

4. Results

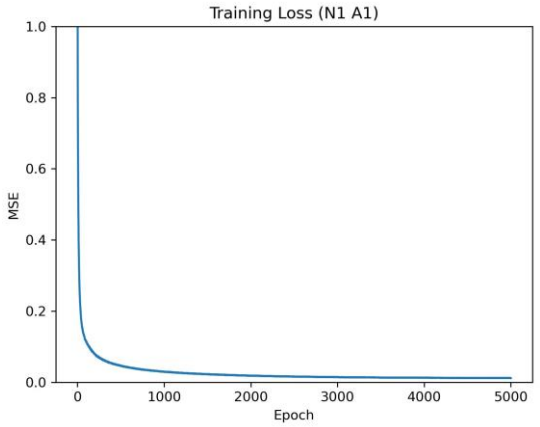
4.1 Model Training

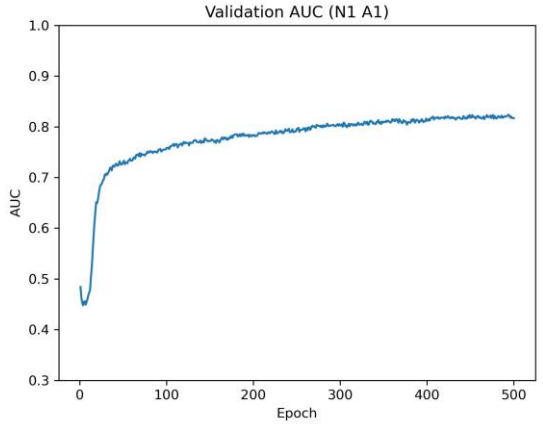
CAE



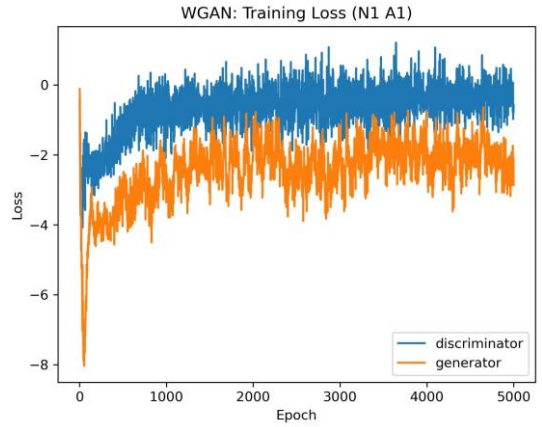


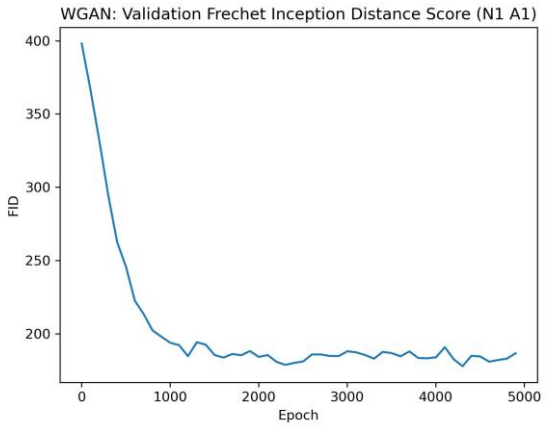
SSAE



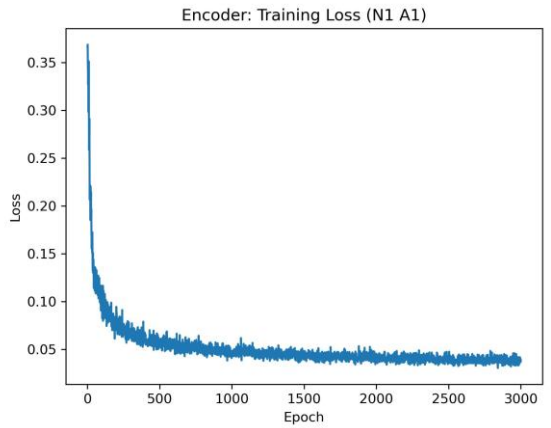


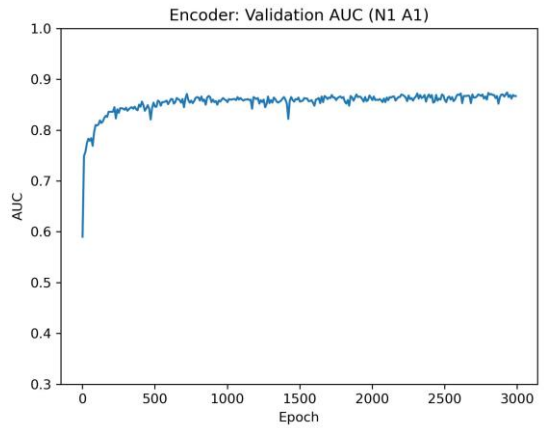
f-AnoGAN: WGAN-GP





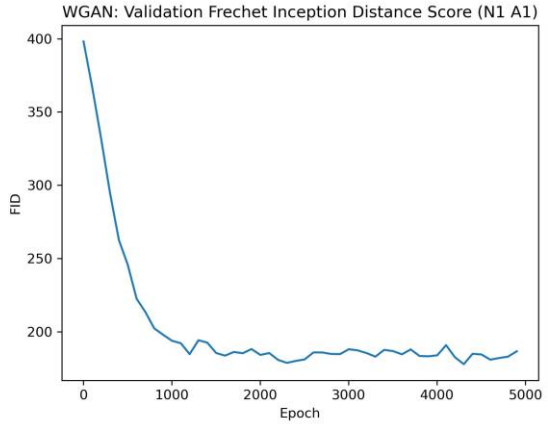
f-AnoGAN: Encoder



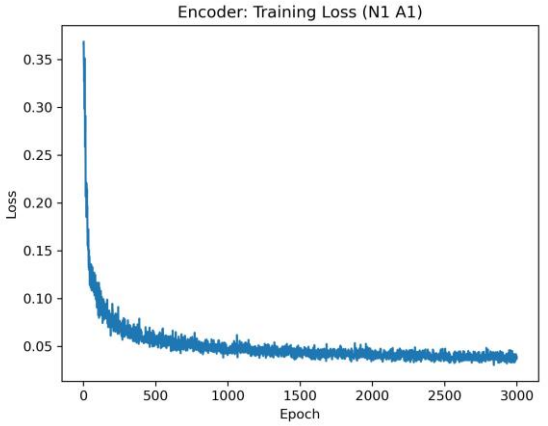


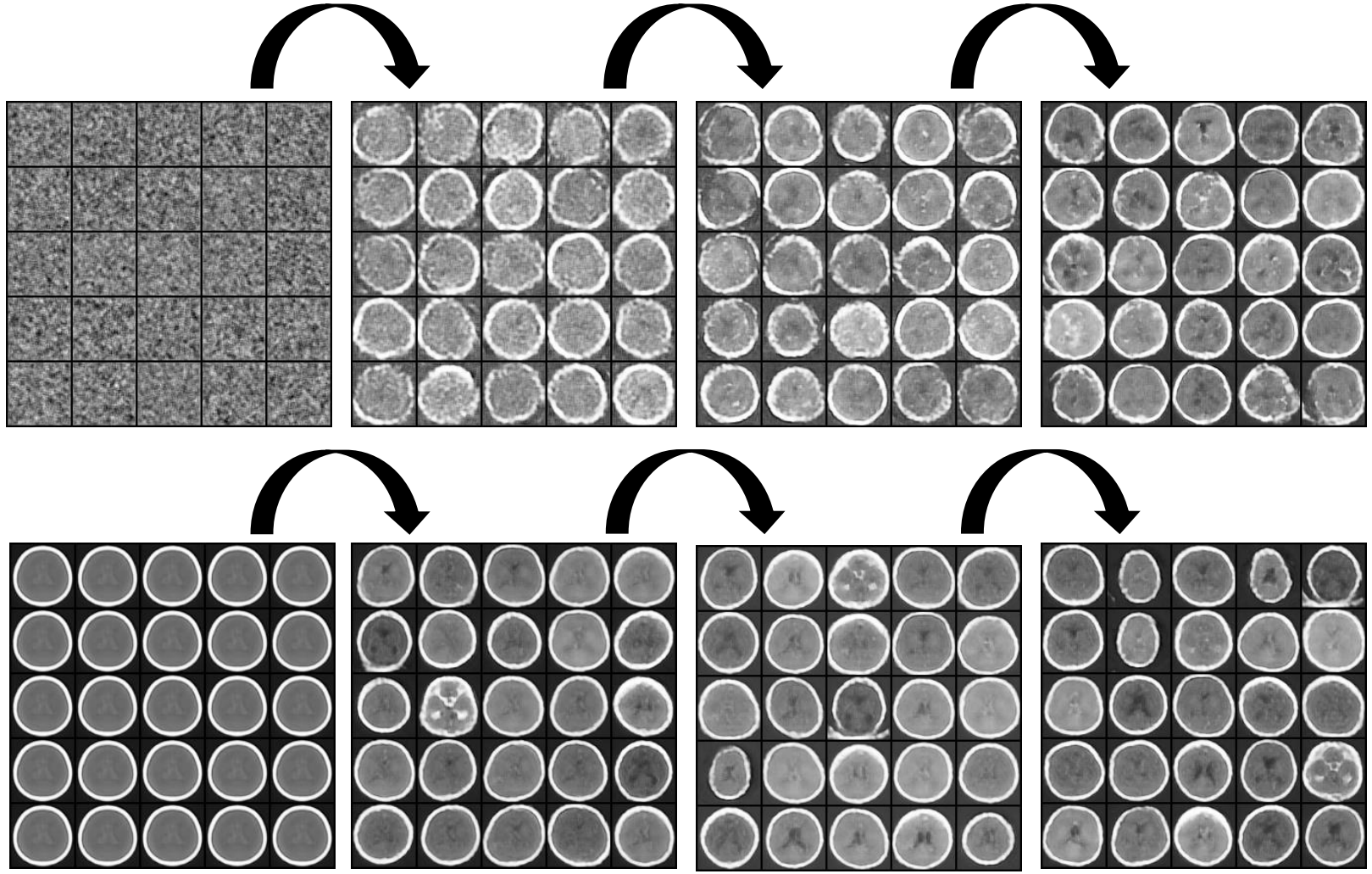
4.1 Model Training: f-AnoGAN Image Generation

f-AnoGAN: WGAN-GP



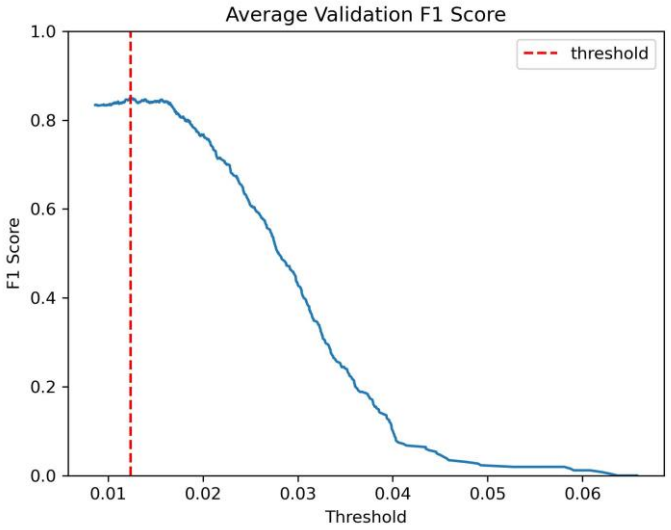
f-AnoGAN: Encoder



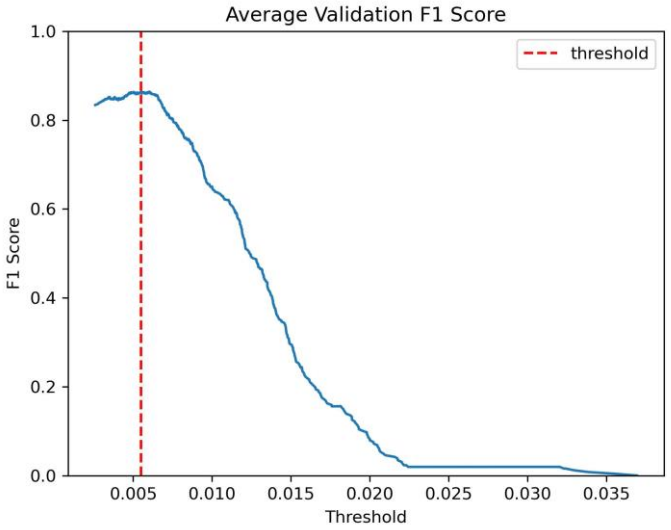


4.2 Anomaly Threshold Selection

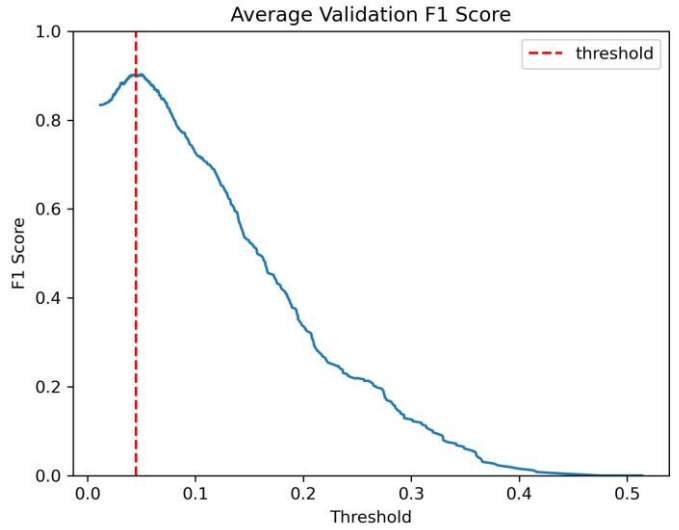
CAE



SSAE



f-AnoGAN

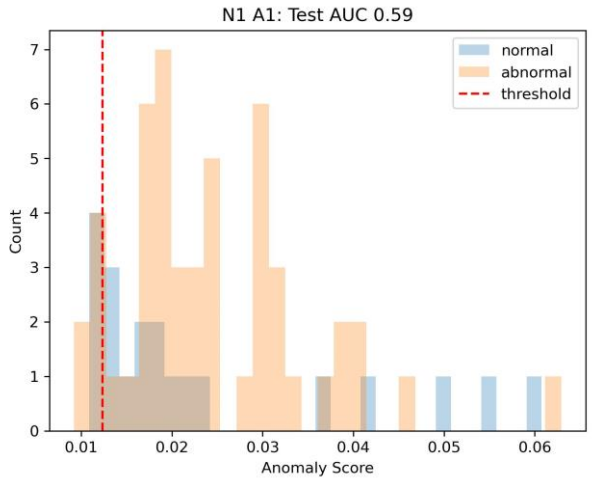


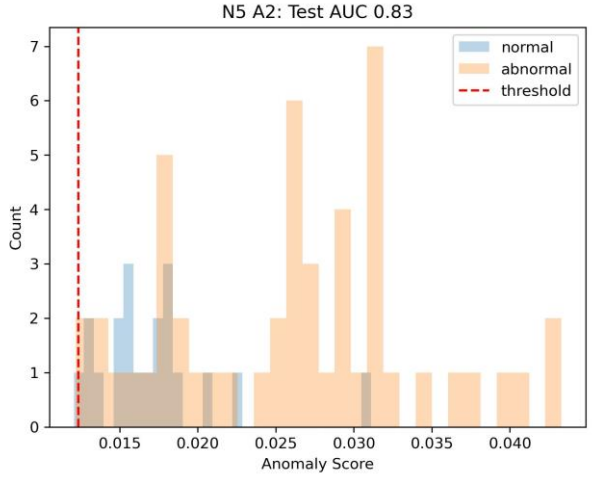
4.3 Test Metrics Summary

Model	AUC	Accuracy	Recall	Precision	F1
CAE	0.78±0.09	0.74±0.04	0.93±0.05	0.77±0.04	0.84±0.02
SSAE	0.81±0.08	0.80±0.02	0.88±0.02	0.84±0.03	0.86±0.01
f-AnoGAN	0.87±0.06	0.86±0.02	0.93±0.02	0.88±0.01	0.91±0.01

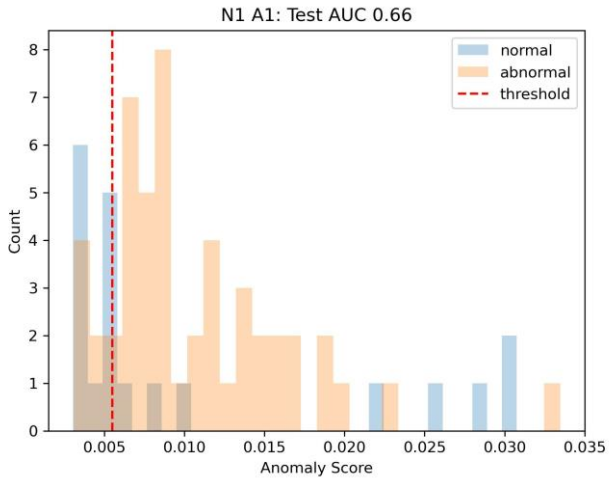
4.4 Test Anomaly Histograms

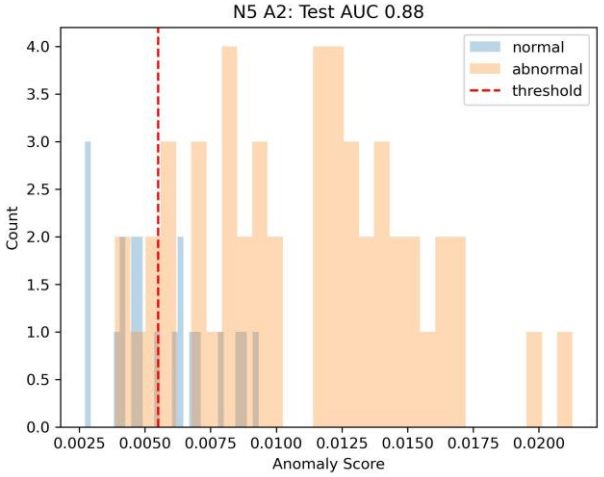
CAE



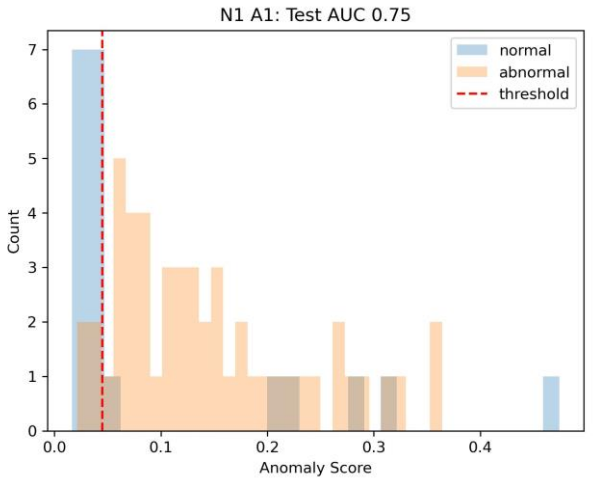


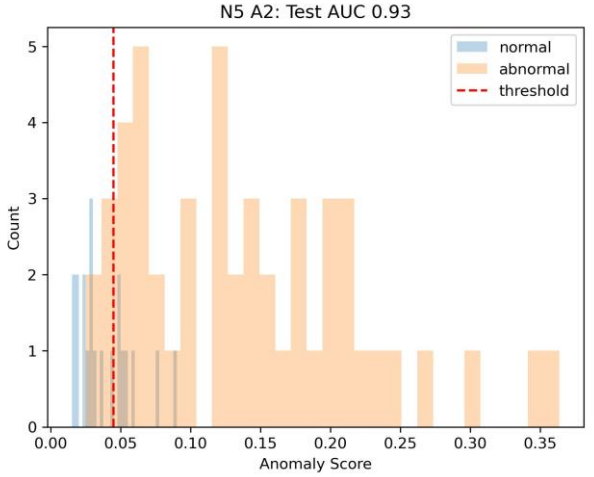
SSAE



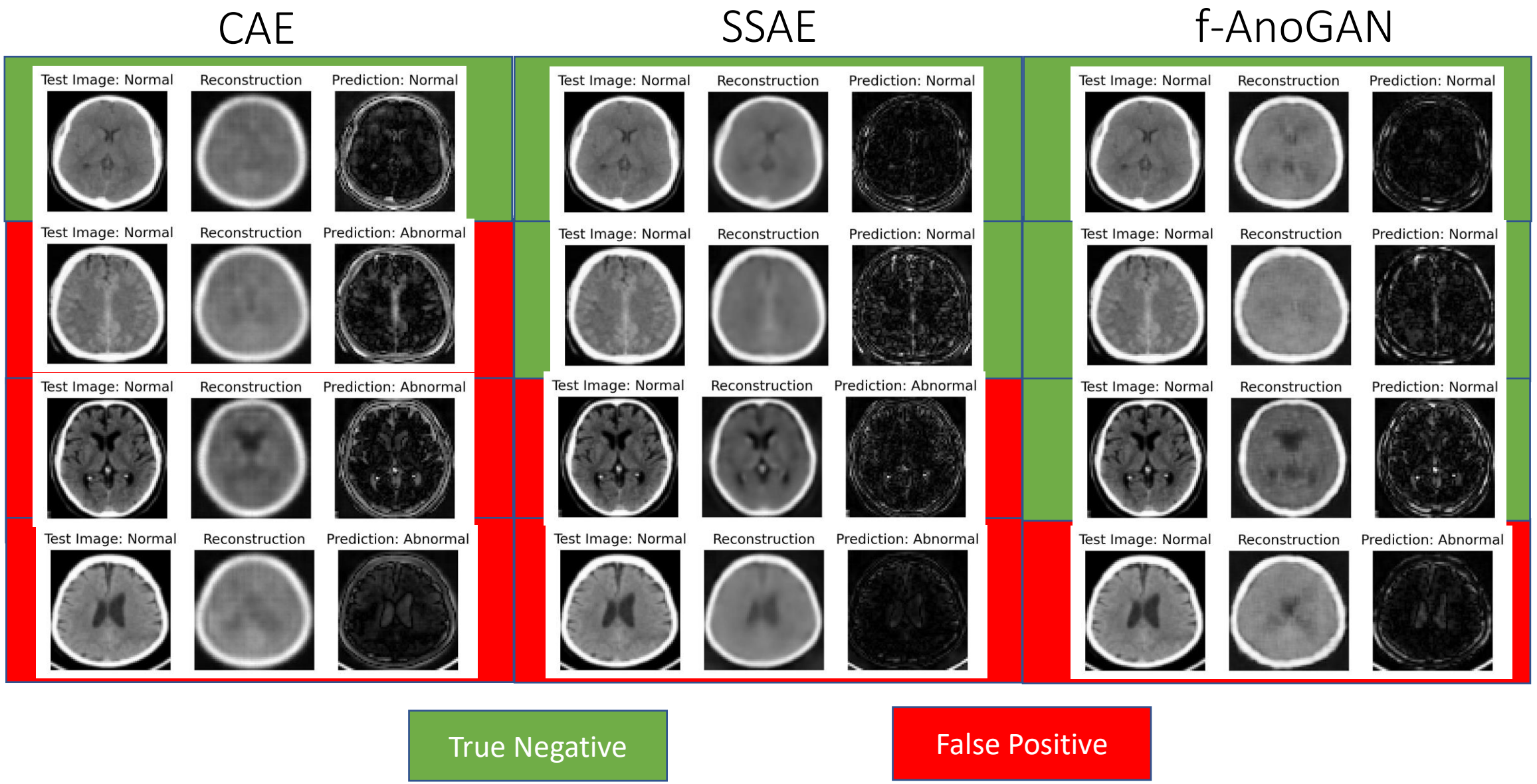


f-AnoGAN

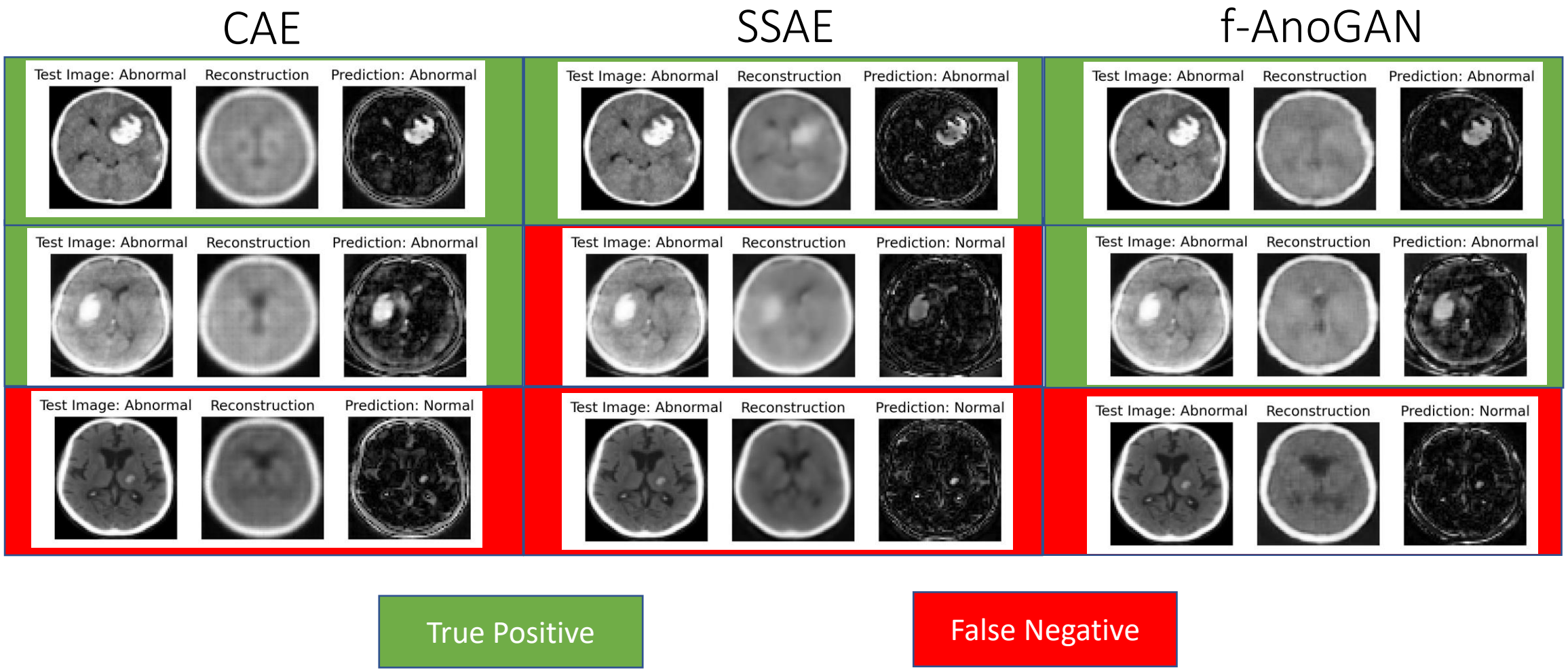




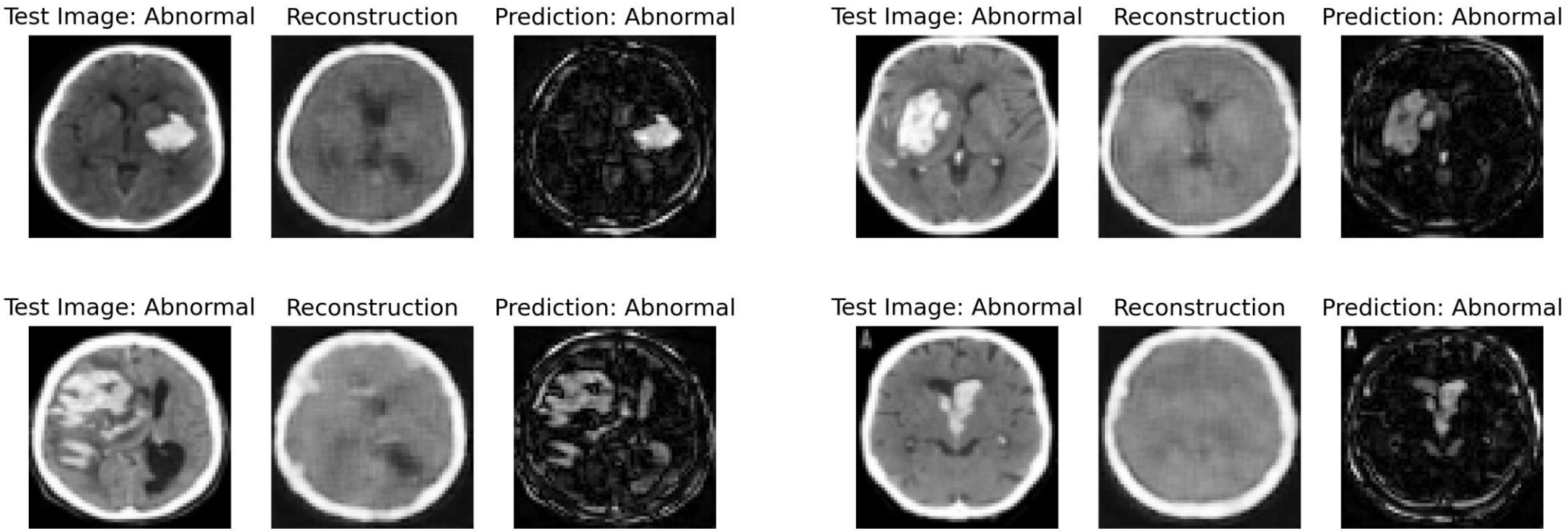
4.5 Test Image Reconstruction and Anomaly Prediction



4.5 Test Image Reconstruction and Anomaly Prediction



4.6 Anomaly Segmentation



4.7 Other Attempts

- Patch-wise training: no significant improvement
- Higher image resolution: performance degradation
- DAGMM: did not work on this dataset

5. Discussion

5.1 Dataset Limitation

- Insufficient images for training
  - Image resolution has to be reduced
  - Difficult to learn data distribution
  - GAN does not generate a diverse set of images
  - Solution: data augmentation
    - Affine transformation is very helpful
    - Excessive color jitter harms model performance
- Absence of Patient Information
  - Possibility of information leakage (images from the same person present in both training and inference sets)
    - Training with N, inference with N: biased towards better performance
    - Training with N, inference with A: biased towards worse performance
- Image noise and artifacts
  - Data preprocessing might help: noise reduction, brain segmentation, etc.

5.2 Model Analysis

- Convolutional autoencoder
  - Simple and fast to train
  - Cannot recover fine details
- Scale space autoencoder
  - Good reconstruction quality
  - Some abnormalities are recovered
- Fast anomaly detection GAN
  - Best reconstruction quality
  - Seldom recover anomalies
  - Slow to train
  - Requires more data for better generalizability

5.3 Image Reconstruction

- Reconstruction quality is sometimes sacrificed
  - Fine details cannot be recovered

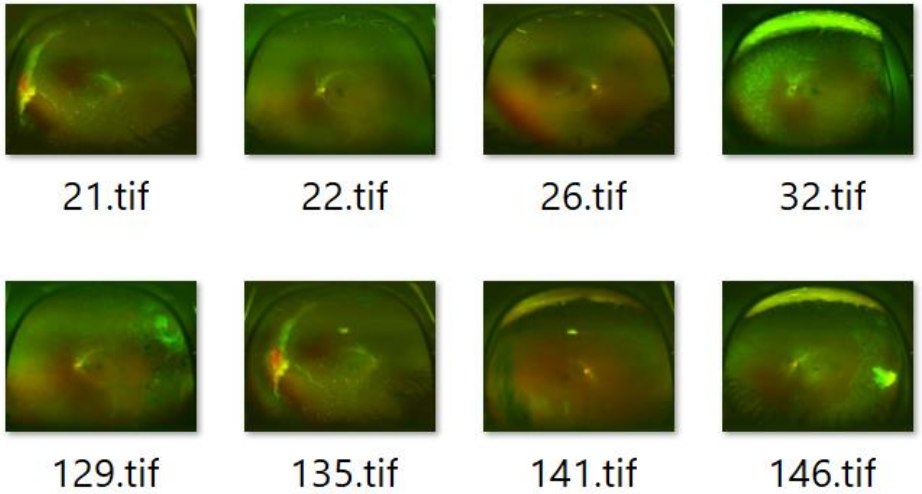
- Find a balance between:
  - Better reconstruction quality => More likely to reconstruct anomalies
  - Worse reconstruction quality => Tend to lose fine details and small anomalies

- Reconstruction loss is averaged over the image:
  - Small abnormal regions are likely to be ignored (low image anomaly score)

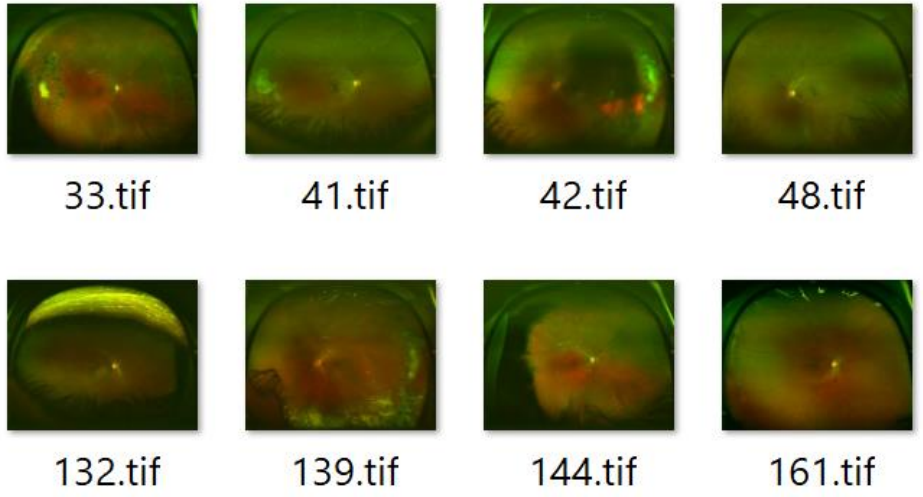
Ultra-Widefield Images

Dataset: UWF Fundus Images

Good: 395



Bad: 202



- Original resolution: 3072x3900x3 and 2048x2600x3
- Low resolution version: 307x390x3 and 205x260x3
- Empty blue channel

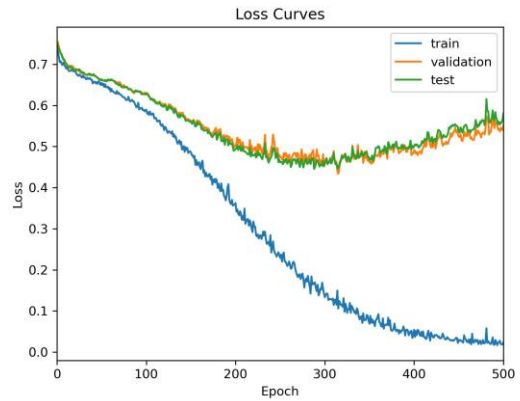
1. Supervised Classification

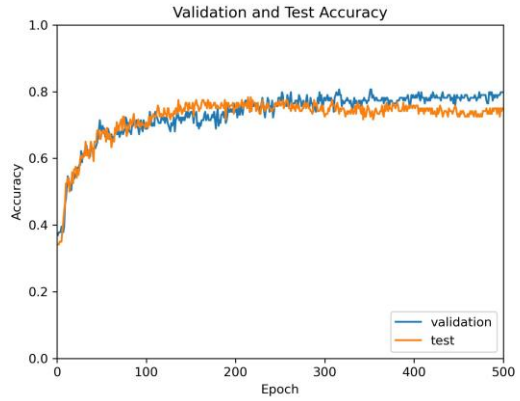
Experiment Setup

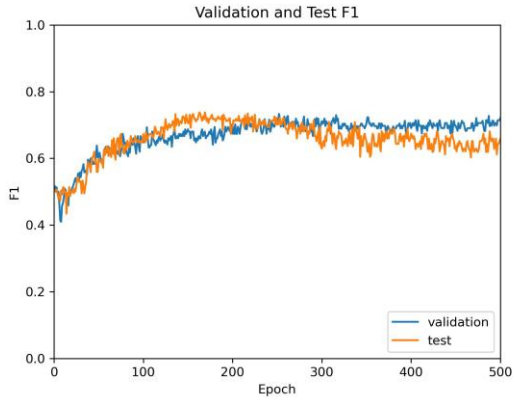
- Data split:
  - Training: 60% good + 60% bad
  - Validation: 20% good + 20% bad
  - Test: 20% good + 20% bad
- Data transformations:
  - resize = T.Resize((224, 224))
  - random_horizontal_flip = T.RandomHorizontalFlip(p=0.5)
  - random_affine = T.RandomAffine(degrees=3, translate=(0.05, 0.05), scale=(0.95, 1.05))
  - normalize = T.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])

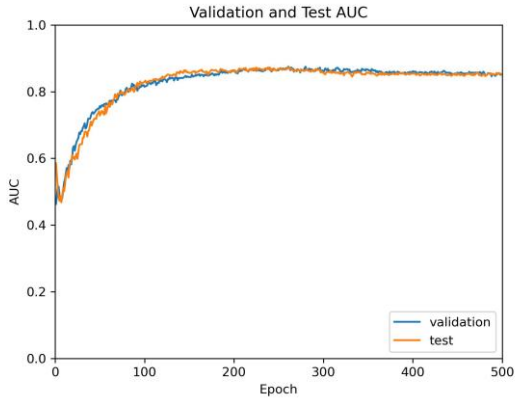
Experiment Setup

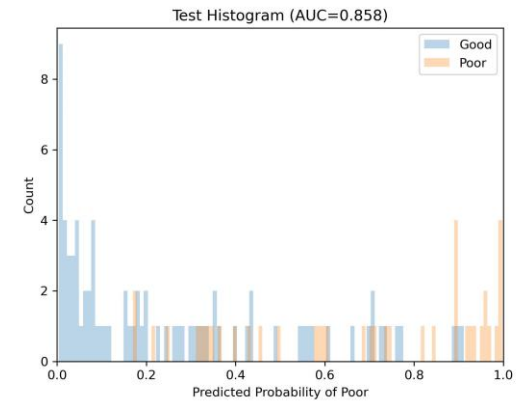
- Model: ImageNet pre-trained DenseNet121 + FC layers
- Training details:
  - Epochs: 500
  - Batch size: 64
  - Optimizer: SGD
  - Learning rate: 1e-3
  - Loss: weighted cross entropy

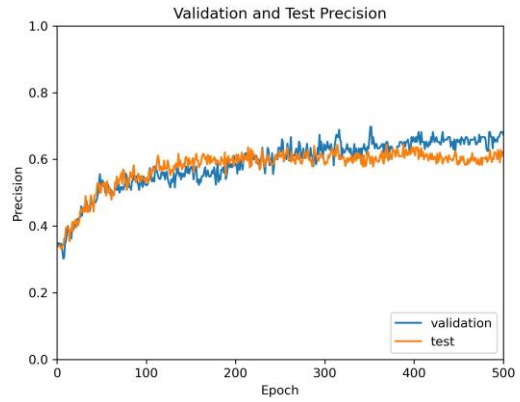


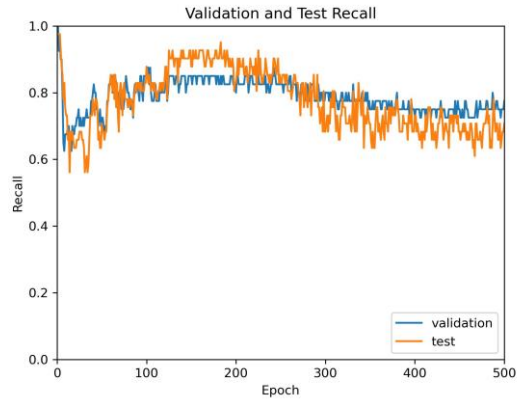


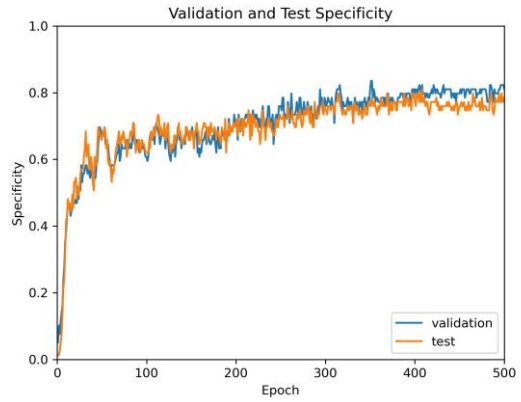






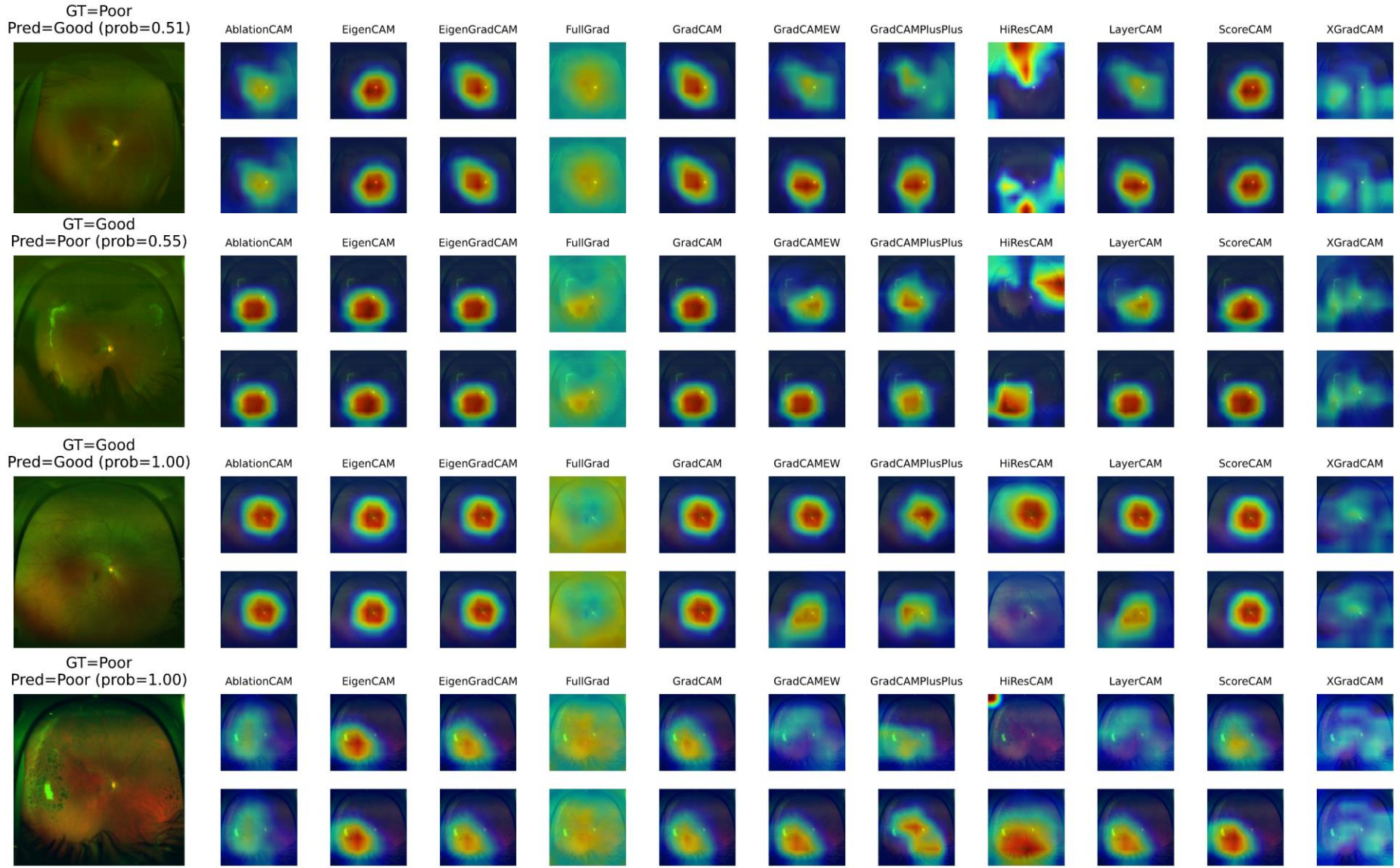






- Minimum validation loss is used to select the best model.
- Final test performance is reported on this best model:
  - Accuracy 0.75
  - Recall 0.66
  - Precision 0.63
  - F1 0.64
  - Specificity 0.80
  - AUC 0.86

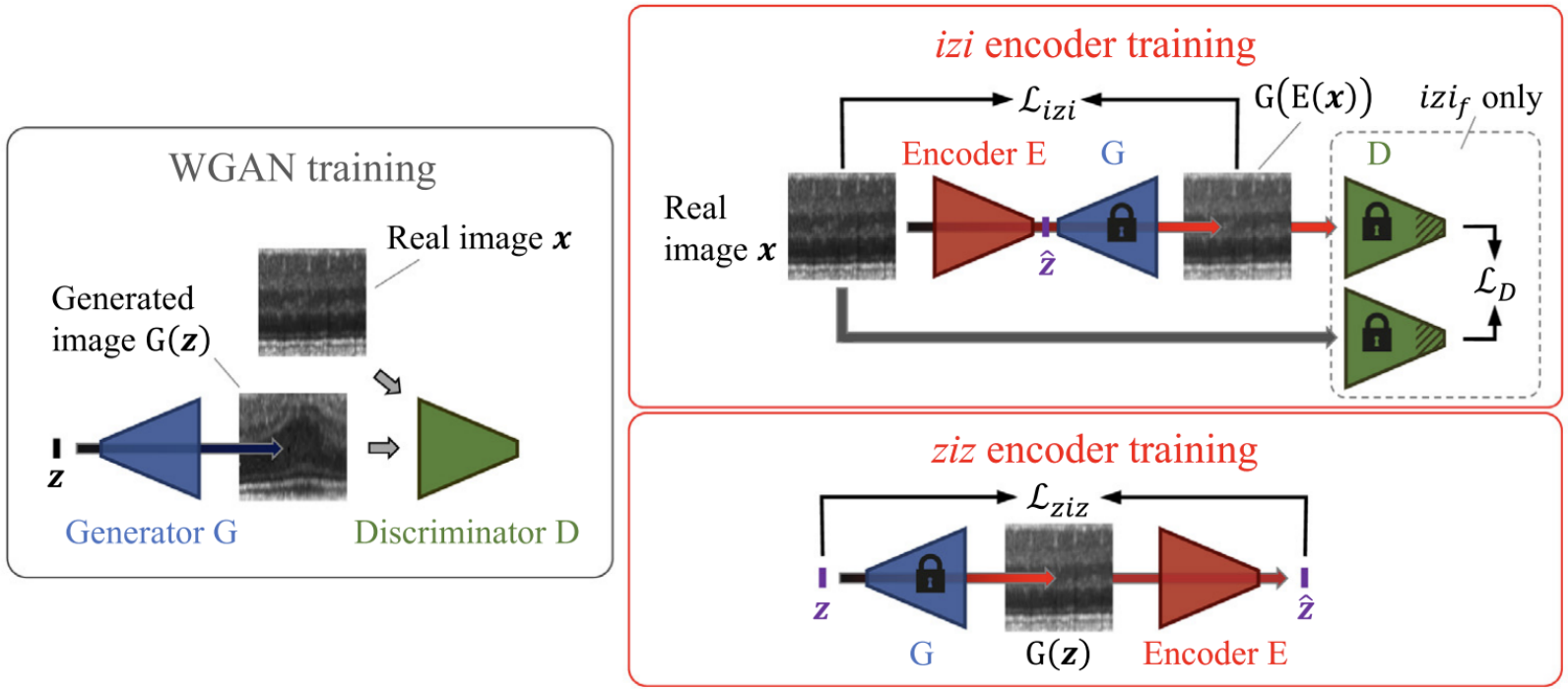
- Note: Test set loss/metrics are ploted here just for model/dataset evaluation purpose. Formally, they should not be.



- Implemented with https://github.com/jacobgil/pytorch-grad-cam
- Target layer is model.features[-1] except FullGrad
- Complete results stored in folder supervised\results\test

2. Unsupervised Learning: f-AnoGAN

Model Architecture: f-AnoGAN



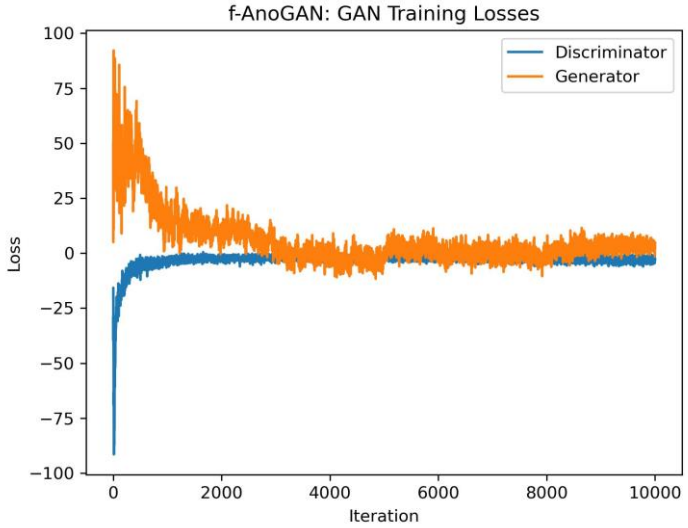
- We use *izi*

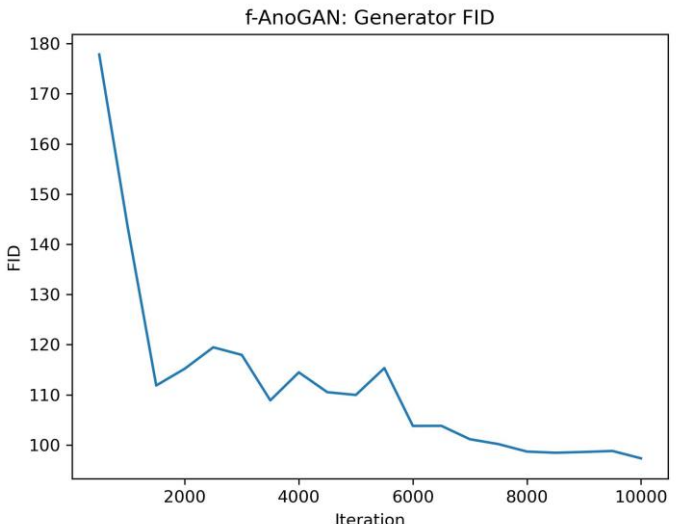
Fig. 2. Components of the proposed anomaly detection framework training. *Wasserstein GAN (WGAN)* training yielding learned parameters for the generator (G) and discriminator (D). Three possible variants of *encoder training* with fixed parameters of G (and D) where only the encoder parameters are adapted. 1) *izi encoder training*: minimizing the loss $\mathcal{L}_{izi}$ based on the residual of real input images and *"reconstructed"* images, 2) *izi*$_f$ *encoder training*: jointly minimizing the loss $\mathcal{L}_{izi}$ based on the residual of real input images and *"reconstructed"* images and the loss $\mathcal{L}_D$ based on the residual on discriminator's features, and 3) *ziz encoder training*: minimizing the loss $\mathcal{L}_{ziz}$ based on the residual of randomly sampled and reconstructed locations in z-space (latent space). (For interpretation of the references to color in the main text, the reader is referred to the web version of this article.)

Experiment Setup

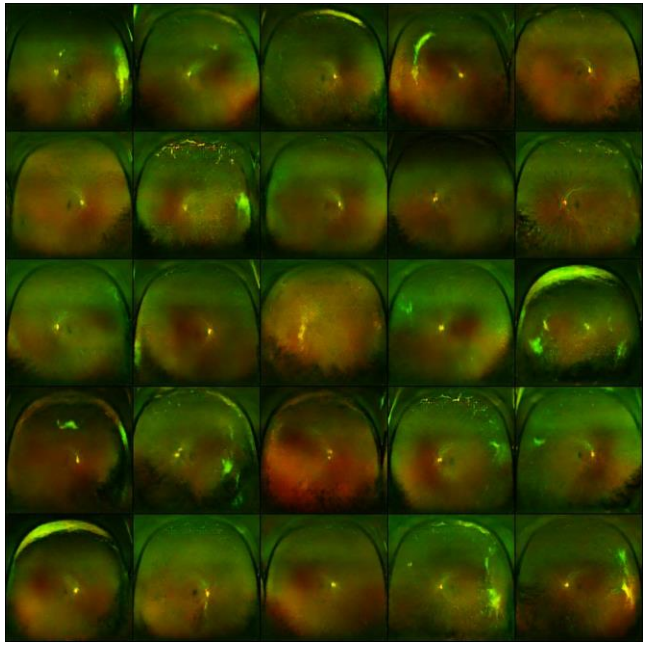
- Data split:
  - Train: 60% good
  - Validation: 20% good + 50% bad
  - Test: 20% good + 50% bad
- Data transformations:
  - Size: 256x256
  - Channels: 2 (blue channel removed)
  - Pixel intensity: percentile 0-99.99 rescaled to [-1, 1]
  - Augmentation
    - horizontal_flip = A.HorizontalFlip(p=0.5)
    - random_affine = A.ShiftScaleRotate(shift_limit=0.05, scale_limit=0.05, rotate_limit=3, p=0.5)

GAN

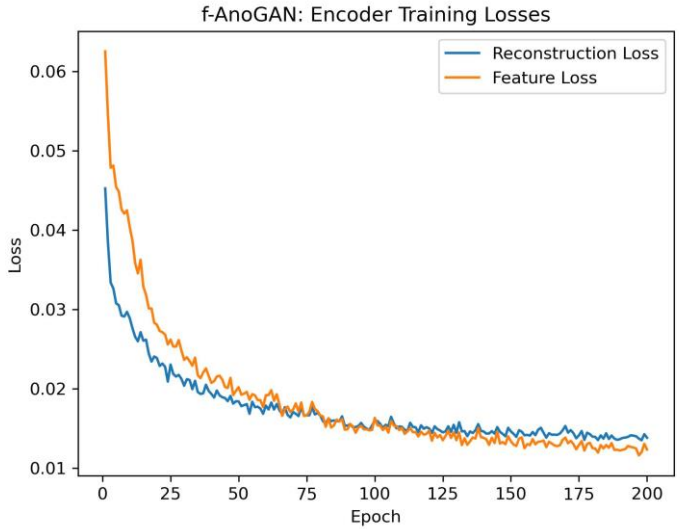


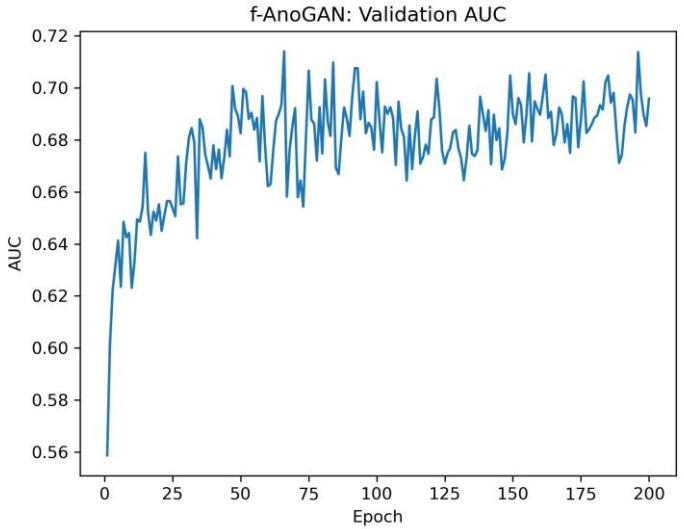


Generated Images

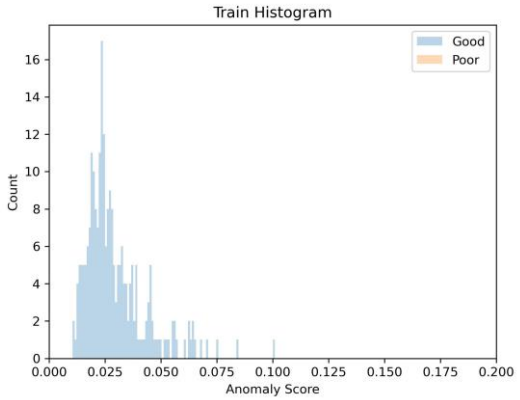


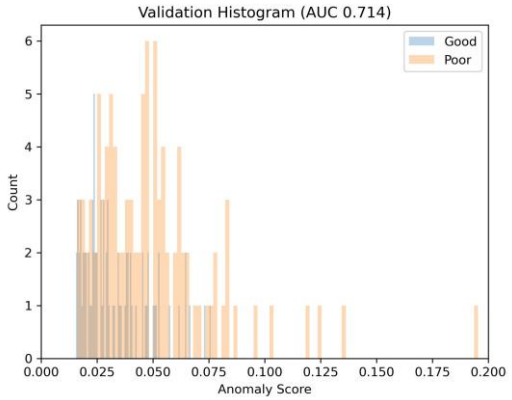
Encoder

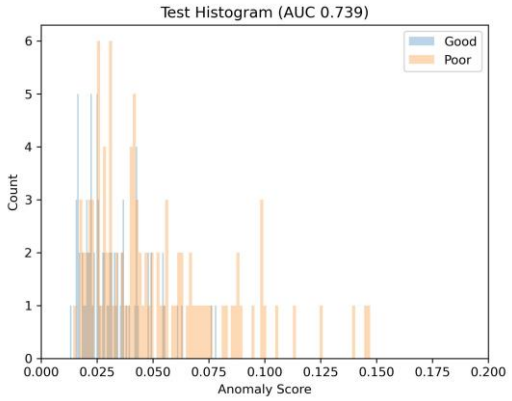




- The final GAN is selected to train the encoder.
- Encoder that results in the maximum validation AUC is chosen to form the best f-AnoGAN, which is then tested on test set.

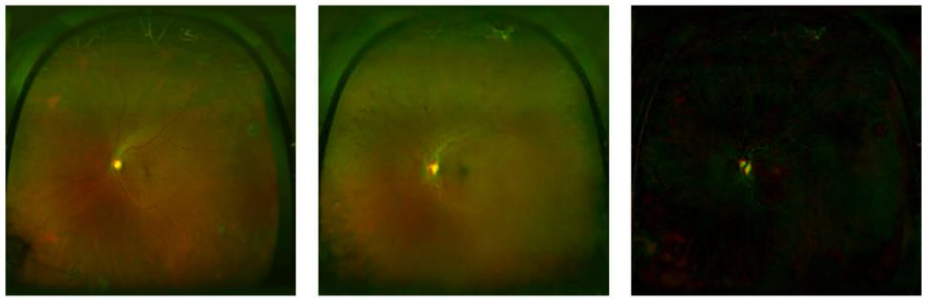




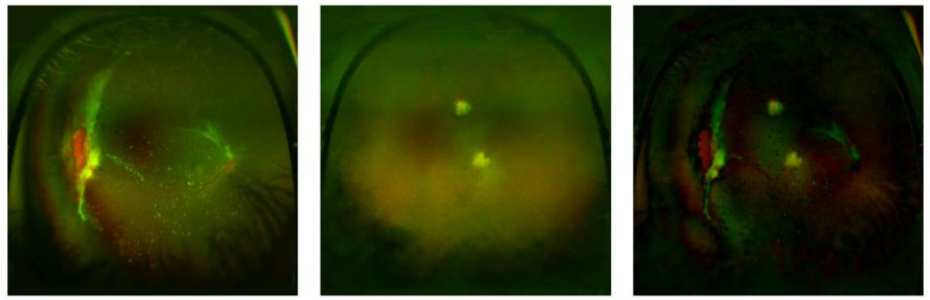


Test set original image, reconstruction, and difference map

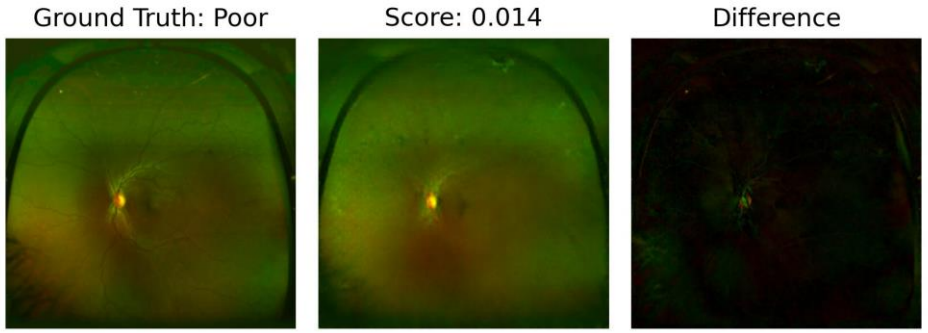
Good Image with Minimum Score



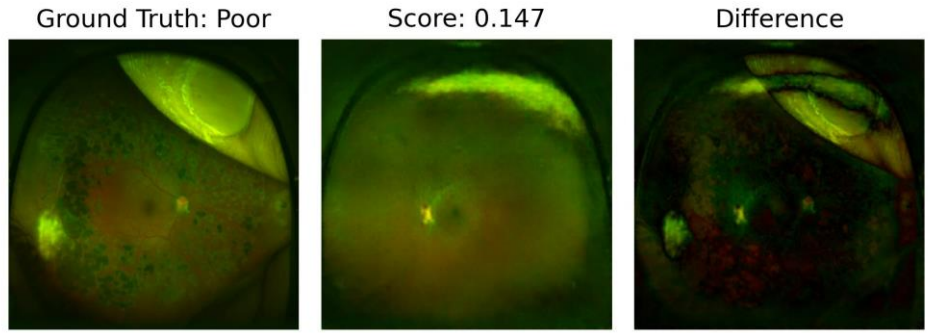
Good Image with Maximum Score



Poor Image with Minimum Score



Poor Image with Maximum Score



- Results reported on the best f-AnoGAN
- Complete results stored in folder fanogan\results\recons_test