Comparison of Anomaly Detection Methods in Chest X-rays

Daniel Alejandro Galindo Lazo[1,2,3] , **Isabel Sarzo Wabi**[1,2,4] , **Léo Valette**[2,5]
[1]Université de Montréal
[2]École Polytechnique Montréal
daniel.alejandro.galindo.lazo@umontreal.ca, isabel.sarzo.wabi@umontreal.ca, leo.valette@polymtl.ca
Student number : [3]2360272, [4]2359192, [5]2307835

Abstract

Anomaly detection in medical imaging poses a significant challenge due to the complexity and subtlety of abnormalities. Inspired by the paper *Anomaly Detection in Medical Imaging with Deep Perceptual Autoencoders*, this project compares the performance of three methods: Deep Isolation Forest (Deep IF), Perceptual Image Anomaly Detection (PIAD), and Deep Perceptual Autoencoder (DPA), on chest X-ray images. This project further proposes evaluating the models on additional metrics including ROC-AUC, accuracy, and F1 score, as well as analyzing the confusion matrix of each. Furthermore, this study compares the obtained results with those reported in the original paper, confirming the superiority of the DPA model, followed by the PIAD and Deep IF, as well as discussing experiences encountered when evaluating the models and the obtained results. The project's source code, data subset, and notebooks are accessible via its dedicated GitHub repository[1].

1 Introduction

Anomaly detection is a technique employed to identify rare occurrences in data based on the knowledge of normal data. Essentially, anomaly detection models excel at identifying data that diverges from their training set [Chandola *et al.*, 2009]. Nonetheless, despite significant advances in deep learning and anomaly detection, these models do not exhibit the same level of efficacy when applied to medical images as they do with natural images. This occurs because images from the medical domain are generally characterized by being very complex and closely resembling normal images. For example, abnormalities in chest X-ray scans are barely visible [Wang *et al.*, 2017], leading these models to struggle in detecting them and potentially confusing them with normal scans. To address this challenge, Shvetsova *et al.* introduced the deep perceptual autoencoder (DPA), a reconstruction-based method that relies on the traditional compression principle inherent to autoencoders, along with the *perceptual loss* metric's capability to capture relevant content information [2021]. The purpose of this project is to evaluate the performance of the DPA in detecting anomalies in chest X-rays by comparing it to other state of the art methods that utilize both distribution-based and reconstruction-based strategies.

1.1 Distribution-based methods

Distribution-based methods identify abnormal instances by locating them in low probability density regions within the "normal" data distribution, where lower probabilities suggest higher likelihood of being considered anomalies. Density-based methods, such as Isolation Forest, function by delineating boundaries around normal instances [Liu *et al.*, 2008]. Models like Deep Isolation Forest (Deep IF) have integrated deep learning to enhance representation, leveraging Isolation Forest on features extracted from pre-trained deep networks. [Ouardini *et al.*, 2019].

In their study, Ouardini *et al.* used a deep convolutional neural network pre-trained on ImageNet to extract features, which are then fed into Isolation Forests. They evaluated various unsupervised anomaly detection methods on both natural image datasets and medical image datasets for screening diabetic retinopathy. They demonstrated that Deep IF was versatile and well-capable of enhancing performance with limited labeled anomalies.

1.2 Reconstruction-based methods

Reconstruction-based methods rely on the principle that models trained exclusively on normal data struggle to accurately reconstruct images. Thus, the reconstruction error between the output (reconstructed image) and the input (original image) is higher when the input is an anomaly. Approaches that utilize Generative Adversarial Networks (GANs), such as AnoGAN [Schlegl *et al.*, 2017], have also been implemented in the anomaly detection field, based on the premise that if never trained with abnormal data, the generator network is only able to generate normal samples.

Other approaches combine both the principles of GANs and autoencoders. An example is the *perceptual image anomaly detection* (PIAD) model [Tuluptceva *et al.*, 2019], which leverages the power of GANs by employing them twice: first to map the latent space to the image space and again to create an inverse mapping. Notably, PIAD differs from traditional denoising autoencoder approaches by relying on the strength of adversarial loss for distribution map-

[1]https://github.com/deca1111/Anomaly_Detection_Comparison

ping. Moreover, it introduces a novel relative-perceptual-L1 loss for evaluating reconstruction fidelity.

However, despite all of these advances, there is still much room for improvement. Searching for the right size of the bottleneck still proves a bit challenging when designing autoencoder-based models. Shvetsova *et al.* also place a strong emphasis in the importance of choosing an adequate dissimilarity metric. For instance, a low reconstruction error does not necessarily mean higher accuracy. If the size of the bottleneck is not chosen correctly, the model could also learn to reconstruct anomalous data. Instead, the DPA utilizes the perceptual loss to give more weight to the content of the images rather than to the model's capability of reconstructing the inputs from their compressed representation. In this sense, the output may not be a coherent image at all but rather a tensor containing the most pertinent information of the input data. In their study, Shvetsova *et al.* show that this approach significantly increases the model's capacity to detect anomalies, as it makes it more flexible in collecting meaningful information by removing the restriction of producing a realistic image as output.

2 Methodology

Three anomaly detection models, Deep IF, PIAD, and DPA, were evaluated and compared using NIH's chest X-rays database. Each model was tested within a dedicated Google Colab notebook, where the models' configurations, training, and evaluation routines were retrieved from Shvetsova *et al.* GitHub repository. Image resizing was performed on a subset of images using the authors' routines and zipped into a personal GitHub repository. Two additional metrics, accuracy and F1-score, were implemented in the evaluation routines to further compare the models alongside the AUC-ROC. A confusion matrix was also plotted for each of the models. These metrics required the use of a threshold, which was determined from the ROC curve of each model (see 2.4).

2.1 Theoretical Background

Deep Isolation Forest

Isolation Forest detects anomalies by evaluating their isolation difficulty within the data space by employing a linear axis-parallel isolation examining one dimension at a time. Even in hyper-plane-based isolation, linear partitioning is still required and some anomalies are not distant from other data points, especially when they cluster close to neighboring data points [Liu *et al.*, 2008].

To address this, Xu *et al.*. introduced the Deep IF, using neural networks to map the data into multiple new data spaces, achieving non-linear isolation through simple axis-parallel partitions within transformed data spaces. Deep IF first creates a random representation (function G) and then performs an isolation-based anomaly scoring (function F) [Xu *et al.*, 2023].

The random representations provide freedom in partitioning the original data space, enabling various views of the data and aiding the partition-based isolation in identifying distinct patterns. Deep IF produces the random representation ensemble via optimisation-free neural networks:

$$G(D) = \{\mathbf{X}_u \in \mathbb{R}^d \mid \mathbf{X}_u = \phi_u(D;\theta)\}_{u=1}^{r} \tag{1}$$

where r stands for the number of groups, $u : D \rightarrow \mathbb{R}^d$ is a function that takes the original data and transforms it into new spaces with d dimensions. The weights of the network are randomly set. Each group has t trees part of a forest of $T = r \times t$ trees.

Once all trees are set, the score function is used to find out how unusual a given data point is with Isolation Forest:

$$F(o|\mathcal{T}) = \Omega_{\tau_i \sim \mathcal{T}} I(o|\tau_i) \tag{2}$$

where $I(o|\tau_i)$ denotes a function to measure the isolation difficulty in τ_i.

Ouardini *et al.* obtained expressive representations from CNN classifiers trained on ImageNet. Features were extracted from pre-trained models. To manage computational complexity, outputs from early layers were spatially averaged and combined with outputs from fully connected layers to form a comprehensive network representation. Additionally, the paper selects the optimal layer of the feature extractor network through validation [Ouardini *et al.*, 2019].

Perceptual Image Anomaly Detection (PIAD)

The core idea behind PIAD, proposed by Tuluptceva *et al.* (2019), is to harness the power of Generative Adversarial Networks (GANs) by employing them twice in the anomaly detection process. Initially, GANs are utilized to map the latent space to the image space, and subsequently to create an inverse mapping. This dual application of GANs enables the model to learn representations that effectively capture the underlying data distribution while simultaneously ensuring fidelity in image reconstruction.

The structure of PIAD involves the joint training of a generator (G) and an encoder (E) to fulfill three key conditions:

1. Firstly, G is tasked with mapping from the latent distribution to the data distribution: $G : p_Z \rightarrow p_X$.
2. Secondly, the encoder (E) is responsible for mapping from the data distribution to the latent distribution: $E : p_X \rightarrow p_Z$.
3. Finally, the reconstructed image generated by G from the latent vector predicted by E should closely resemble the original input image (reconstruction term): $G(E(x)) \approx x$.

To meet conditions 1 and 2, PIAD exploits the adverse losses facilitated by two discriminators (DX and DZ). To evaluate the reconstruction term, they propose to use the relative perception loss-L1, denoted by $L_{rel-perc-L1}$, defined as follows:

$$L_{rel-perc-L1}(x, \check{x}) = \frac{||\hat{f}(x) - \hat{f}(\check{x})||_1}{||\hat{f}(x)||_1}, \tag{3}$$

where $\hat{f}(x)$ represents the feature map obtained from a deep layer of the network on image x. This loss function evaluates the reconstruction fidelity by measuring the relative difference between **feature maps** from a given image x and its reconstructed version $\check{x}$. By normalizing the difference with

respect to the magnitude of the feature map of the original image x, the perceptual loss accounts for variations in image content and contrast, making it robust to changes in image characteristics.

Deep Perceptual Autoencoder

As mentioned in the above sections, the DPA is based on the classic autoencoder approach that is trained to learn patterns from a compressed representation of the input and then use those learned patterns to reconstruct the images correctly. The main difference compared to a traditional autoencoder is that the DPA uses a loss function to measure content dissimilarity. Essentially, the DPA uses the same relative-perceptual-L1 loss introduced in the PIAD model (see eq. 3) to train the autoencoder and compute the reconstruction error.

In addition to the perceptual loss, the DPA also integrates a progressive growing technique, which increases the level of the perceptual content in the loss function. During training, more layers are incrementally added to the autoencoder and the resolution is scaled-up by a factor of 2, resulting in a progressive augmentation of the depth of the features [Shvetsova *et al.*, 2021]:

$$L_{rec} = \alpha * L_{rec}(f_2(x), f_2(\check{x})) + (1-\alpha) * L_{rec}(f_1(down(x)), f_1(down(\check{x}))), \quad (4)$$

where $down()$ represents a downsampling operation by a factor of 2, and α has a linear increase from 0 to 1.

Furthermore, the DPA model challenges the idea of an unsupervised setting during training by introducing a weakly-supervised paradigm. Instead of training the autoencoder with only normal images, a small percentage of labeled anomalous samples are included (less than 0.5 % of the training set) mainly to select the model hyperparameters during setup. Shvetsova *et al.* report that this paradigm increased the DPA's performance by 2 %.

2.2 Data Set

This project utilized the NIH's public dataset, ChestX-ray14, comprising over 100,000 frontal-view X-ray images obtained from more than 30,000 patients. Each image is annotated with up to eight distinct diseases, identified through natural language processing of corresponding radiological reports [Wang *et al.*, 2017].

- Elements: 112,120 frontal-view X-ray images
- Patients: 30,805
- Collected: 1992-2015
- Size: $\sim$ 42 GB zipped

Due to the immense size of the original dataset, it surpassed the storage capacity available within Google Colab Pro's environment. Given the limitations in space and the significant time required for code execution and resizing, it was decided to utilize only a subset of 7,311 images from the original dataset. This subset was suggested by the authors, which contains clearer differences between normal and abnormal chest X-rays.

2.3 Materials

A consistency between experiments was maintained by conducting experiments in a uniform environment, leveraging the Google Colab Pro plataform. Additionally, efficient collaboration within the team was ensured through version control and code sharing on GitHub. The following resources were used:

- Backend Google Compute Engine Python 3 (GPU)
- Tensorboard
- Github

It should be emphasized that the methodologies for subset splitting, model configuration, training, validation, and preparation of runs were extracted not only from the discussed paper by Shvetsova, *et al.* but also from the GitHub[2] repository shared by the authors.

2.4 Evaluation metrics

Metrics for evaluating anomaly detection algorithms measure the model's ability to distinguish between normal and anomalous instances. These metrics guide the algorithm refinement to detect anomalies and minimize false alarms.

Receiver Operating Characteristic (ROC) Curve

A ROC curve is a graphical representation of the performance of a binary classification model across various thresholds. ROC curves provide visualization of a model's ability to distinguish between normal and anomalous instances across various threshold settings.

The area under the ROC curve (AUC-ROC) is used as a summary statistic to quantify the overall performance of the model. A higher AUC-ROC value indicates better discrimination between the positive and negative classes.

Threshold selection

Detecting anomalies needs establishing a threshold beforehand, a task fraught with complexity. This difficulty stems from the variability in anomaly types, each demanding a tailored approach for effective detection. In the realm of medical imaging, our paramount concern lies in mitigating false negatives, which pose a tangible risk to patient well-being. Conversely, false positives, while less critical, warrant attention and can be mitigated through secondary expert review.

To evaluate these models and produce the following metrics, we have opted for an automated threshold determination method rooted in ROC curve analysis. By pinpointing the threshold that strikes the optimal balance between false positives and false negatives (see Figure 4), we aim for robust and comparable outcomes across diverse anomaly detection methodologies.

F1 Score

The F1 score is a metric that combines precision and recall into a single value. It provides a balance between these two metrics and is particularly useful when classes are imbalanced. The F1 score is calculated as the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

[2]https://github.com/ninatu/anomaly_detection/

Accuracy

Accuracy, as a basic metric, provides an overall measure of how well a model correctly identifies anomalies and normal instances.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{6}$$

Confusion matrix

Confusion matrix complements these metrics by providing a breakdown of the model's predictions, highlighting true positives, true negatives, false positives, and false negatives. It is typically represented as:

True Negative (TN)	False Positive (FP)
False Negative (FN)	True Positive (TP)

3 Results and experiences

The following subsections provide a description of the results obtained after training and evaluating each of the three models.

3.1 Deep IF

The Deep IF algorithm was trained and evaluated within minutes, with three runs conducted to measure timing during both training and evaluation stages. Training required more time due to the need to download the ImageNet pretrained model for feature extraction, taking approximately 11 minutes. Evaluation, on the other hand, was notably faster, averaging around 1.2 minutes per run.

As an experience, during the process of downloading the pretrained model, it was discovered that Google Colab requires webpages to have an SSL certificate. As the model's webpage lacked this certificate, an error was encountered. However, it was determined that installing a package for SSL certificate management and then disabling SSL certificate verification could bypass this issue.

Upon evaluating the results, Table 1 presents the model's performance metrics.

Metric	Score
AUC-ROC	0.760
Accuracy	0.704
F1 Score	0.730

Table 1: Performance of Deep IF evaluated with AUC-ROC, Accuracy, and F1 Score

Additionally, Figure 1 offers a detailed insight into the model's predictions through its confusion matrix.

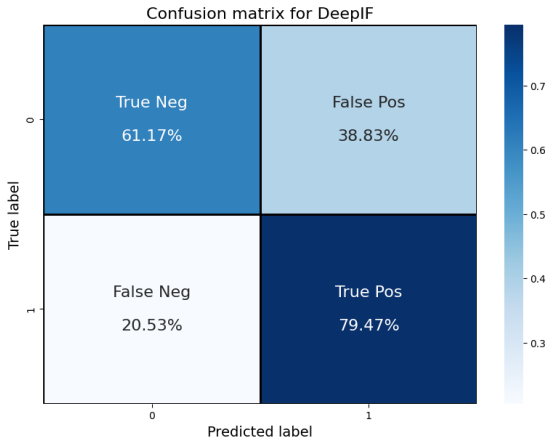

Figure 1: Deep IF Confusion Matrix ($threshold = -0.435$).

3.2 PIAD

Training the PIAD model took over 2 hours (125 minutes) despite the use of high-performance GPUs in the cloud. For this reason, we were only able to carry out one complete and conclusive training run, the numerous unsuccessful attempts having consumed more time and computing capacity than expected.

The model evaluation, on the other hand, took less than a minute. Table 2 shows the results obtained at the end of model training.

Metric	Score
AUC-ROC	0.885
Accuracy	0.815
F1 Score	0.804

Table 2: Performance of PIAD evaluated with AUC-ROC, Accuracy, and F1 Score

In addition, the confusion matrix in Figure 2 shows the strengths and weaknesses of the model in terms of anomaly detection.

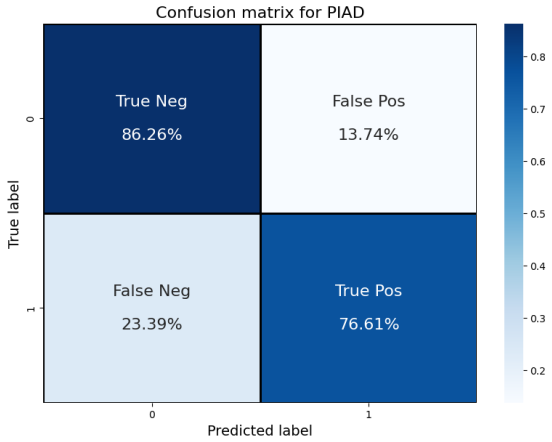

Figure 2: PIAD Confusion Matrix ($threshold = 0.548$).

3.3 DPA

The DPA model was trained under the weakly-unsupervised paradigm, which took a total of 71.26 minutes to complete. On the other hand, its evaluation was significantly faster, finishing in under a minute.

It should be noted that integrating the progressive growing technique imposed a significant resource demand on the algorithm. In the study by Shvetsova *et al.*, training with progressive growing tripled the total runtime when compared to training the model without it [2021]. Unfortunately for this project, due to resource constraints, the model training couldn't be completed using this technique. Despite multiple attempts to execute the progressive growing, each endeavor resulted in Colab shutting down and becoming unresponsive for several hours.

Table 3 shows the performance results of the DPA model (without progressive growing) in terms of AUC-ROC, accuracy, and F1 score.

Metric	Score
AUC-ROC	0.925
Accuracy	0.865
F1 Score	0.864

Table 3: Performance of DPA evaluated with AUC-ROC, Accuracy, and F1 Score

In addition, the confusion matrix presented in Figure 3 displays the distribution of true and false predictions made by the model.

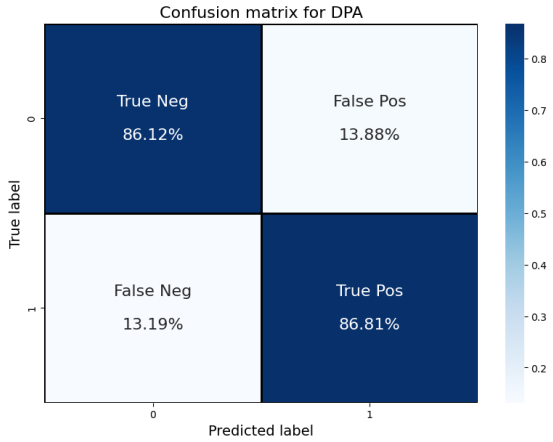

Figure 3: DPA Confusion Matrix ($threshold = 0.402$).

4 Discussion

The chosen approach facilitated comprehensive anomaly detection analysis, revealing insights into model performances. By individually visualizing and evaluating models, differences were identified and performance was evaluated with proposed metrics. This allowed to identify superior models and their relative optimality in terms of runtime as seen in Table 4.

Model	Deep If	PIAD	DPA
Runtime	11.5	125	71.26

Table 4: Average runtime (in minutes) recorded for experiments conducted on the Google Compute Engine Python 3 backend with GPU, utilizing the T4 with PyTorch 1.4.0.

Deep IF demonstrated the highest speed among the models, likely attributed to its utilization of a pre-trained robust feature extraction model such as ImageNet. However, reconstruction-based models like PIAD and DPA require more resources, making them significantly slower. DPA proves to be considerably faster than PIAD due to the absence of adversarial training, which entails computational cost savings.

In terms of performance metrics, DPA demonstrated excellence in AUC-ROC, as illustrated in Figure 4. It consistently surpassed the other models across various thresholds, followed by PIAD and Deep IF. This can be attributed to the differing methods employed, with reconstruction methods proving more effective in anomaly detection compared to distribution-based approaches. This underscores the challenge of data space partitioning in high-dimensional spaces, where even features extracted using pre-trained deep neural networks may struggle to perform optimally.

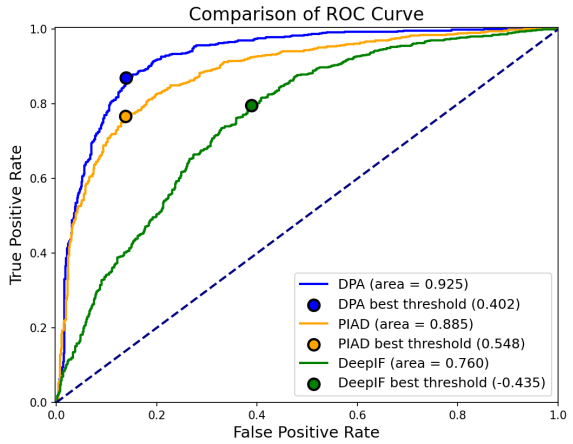

Figure 4: Comparison of ROC curves for DPA (blue), PIAD (orange), and Deep IF (green), along with their respective AUC scores and optimal thresholds.

Even when evaluating models at their optimal thresholds (see Figure 4) for F1 score calculation, the order remained consistent. Despite the smaller magnitude of differences, DPA still outperformed both PIAD and Deep IF in this aspect, as seen in Figure 5.

In terms of accuracy, DPA exhibited superior performance once more. This can be attributed to PIAD's reliance on adversarial training, which requires more resources and training time. Additionally, DPA's emphasis on conserving image content for anomaly detection proved more effective than PIAD's focus on reconstructing realistic images.

As expected, Deep IF exhibited the lowest accuracy and F1 scores among the models. This could be attributed to a domain shift between ImageNet and X-ray images. However, the model's performance was not poor, likely due to the feature extractor network being pre-trained on images closely resembling X-rays.

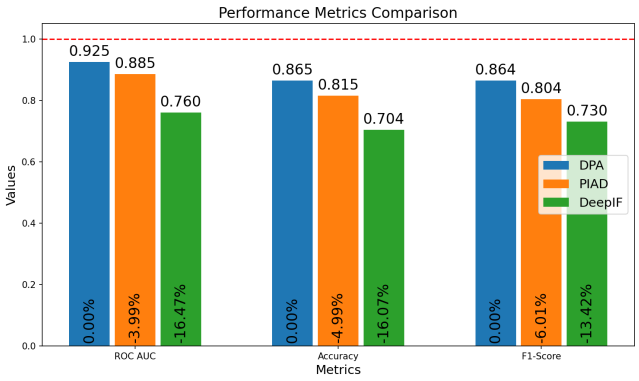

Figure 5: Comparison of DPA (blue), PIAD (orange), and Deep IF (green) performances across ROC-AUC, Accuracy, and F1 score metrics.

Comparing the confusion matrices, it's interesting to note that PIAD and Deep IF exhibit contrasting performances. PIAD demonstrates superior performance in predicting negatives, while Deep IF is better in predicting positives. However, Deep IF struggles in predicting negatives, resulting in more false positives and predicting false anomalies more frequently. Conversely, PIAD excels in discerning true negatives with minimal false positives. Selecting a threshold within the confusion matrix enables the calibration of the anomaly detection model to meet specific requirements. Choosing a threshold of must align with the desired level of conservatism, modeling the performance to the particular demands of anomaly detection.

On the other hand, DPA outperforms Deep IF in detecting true positives among the three models. Additionally, DPA performs comparably to PIAD in true negative prediction, effectively identifying patients without anomalies. This analysis of the matrices reaffirms DPA's ability to excel in predicting true labels, while highlighting the distinct performance characteristics of PIAD and Deep IF.

4.1 Comparison with original paper

The results obtained in this project were compared to those presented in the study by Shvetsova *et al.* Table 5 illustrates this comparison. It is not surprising that there is not a substantial difference between the results given that the models were trained with the same hyperparameters reported in the study, as well as the same subset of data.

Regarding the DPA model, it should be mentioned that results from the original study as presented in Table 5 also exclude the progressive growing.

Deep IF		PIAD		DPA	
Original	Ours	Original	Ours	Original	Ours
0.766	0.760	0.88	0.885	0.920	0.925

Table 5: Comparison of ROC AUC for three models between the results of Shvetsova *et al.* and those obtained in this experiment.

5 Conclusions

In this project, three distinct models (Deep IF, PIAD, and DPA) were implemented and evaluated for an anomaly detection task using a popular public dataset containing images of chest X-rays. Comparing the three approaches, it was found that the best detection performance was obtained by the DPA (AUC-ROC = 92.5 %). This outcome supports the hypothesis that employing the perceptual loss as a dissimilarity metric to minimize the reconstruction error enhances model performance as it captures image content effectively without being limited to producing a realistic output. The obtained results were also compared to those reported in the original study [Shvetsova *et al.*, 2021], showing significant similarity, particularly in terms of AUC-ROC. In addition to this metric, the optimal threshold was selected from the ROC curve to measure accuracy and F1-score. Incorporating these additional metrics, alongside constructing corresponding confusion matrices for each model, offers deeper insights into the detection capability of the models.

References

[Chandola *et al.*, 2009] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[Liu *et al.*, 2008] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[Ouardini *et al.*, 2019] Khalil Ouardini, Huijuan Yang, Balagopal Unnikrishnan, Manon Romain, Camille Garcin, Houssam Zenati, J Peter Campbell, Michael F Chiang, Jayashree Kalpathy-Cramer, Vijay Chandrasekhar, et al. Towards practical unsupervised anomaly detection on retinal images. In *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data: First MICCAI Workshop, DART 2019, and First International Workshop, MIL3ID 2019, Shenzhen, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13 and 17, 2019, Proceedings 1*, pages 225–234. Springer, 2019.

[Schlegl *et al.*, 2017] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[Shvetsova *et al.*, 2021] Nina Shvetsova, Bart Bakker, Irina Fedulova, Heinrich Schulz, and Dmitry V Dylov. Anomaly detection in medical imaging with deep perceptual autoencoders. *IEEE Access*, 9:118571–118583, 2021.

[Tuluptceva *et al.*, 2019] Nina Tuluptceva, Bart Bakker, Irina Fedulova, and Anton Konushin. Perceptual image anomaly detection. In *Asian Conference on Pattern Recognition*, pages 164–178. Springer, 2019.

[Wang *et al.*, 2017] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[Xu *et al.*, 2023] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2023.