Evaluating Privacy on Synthetic Images Generated using GANs: Contributions of the VCMI Team to ImageCLEFmedical GANs 2023

Notebook for the ImageCLEFmedical GANs Lab at CLEF 2023

Helena Montenegro[1,2,*], Pedro Neto[1,2], Cristiano Patrício[2,3], Isabel Rio-Torto[2,4], Tiago Gonçalves[1,2] and Luís F. Teixeira[1,2]

[1]*Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal*

[2]*INESC TEC, Campus da FEUP Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal*

[3]*Departamento de Informática, Universidade da Beira Interior, Rua Marquês de Ávila e Bolama, 6201-001 Covilhã, Portugal*

[4]*Departamento de Ciência de Computadores, Faculdade de Ciências, Universidade do Porto, Rua do Campo Alegre s/n, 4169–007 Porto, Portugal*

Abstract

This paper presents the main contributions of the VCMI Team to the ImageCLEFmedical GANs 2023 task. This task aims to evaluate whether synthetic medical images generated using Generative Adversarial Networks (GANs) contain identifiable characteristics of the training data. We propose various approaches to classify a set of real images as having been used or not used in the training of the model that generated a set of synthetic images. We use similarity-based approaches to classify the real images based on their similarity to the generated ones. We develop autoencoders to classify the images through outlier detection techniques. Finally, we develop patch-based methods that operate on patches extracted from real and generated images to measure their similarity. On the development dataset, we attained an F1-score of 0.846 and an accuracy of 0.850 using an autoencoder-based method. On the test dataset, a similarity-based approach achieved the best results, with an F1-score of 0.801 and an accuracy of 0.810. The empirical results support the hypothesis that medical data generated using deep generative models trained without privacy constraints threatens the privacy of patients in the training data.

Keywords

Privacy, Deep Generative Models, Generative Adversarial Networks, Medical Image Analysis

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

*Corresponding author.

maria.h.sampaio@inesctec.pt (H. Montenegro); pedro.d.carneiro@inesctec.pt (P. Neto); cristiano.p.patricio@inesctec.pt (C. Patrício); isabel.riotorto@inesctec.pt (I. Rio-Torto); tiago.f.goncalves@inesctec.pt (T. Gonçalves); luisft@fe.up.pt (L. F. Teixeira)

https://github.com/helenaMontenegro (H. Montenegro); https://netopedro.github.io/ (P. Neto); https://cristianopatricio.github.io (C. Patrício); https://github.com/icrto (I. Rio-Torto); https://tiagofilipesousagoncalves.github.io (T. Gonçalves); https://www.inesctec.pt/en/people/luis-filipe-teixeira (L. F. Teixeira)

0000-0001-6237-3011 (H. Montenegro); 0000-0003-1333-4889 (P. Neto); 0000-0003-2215-3334 (C. Patrício); 0000-0002-2302-8597 (I. Rio-Torto); 0000-0003-4744-9174 (T. Gonçalves); 0000-0002-4050-7880 (L. F. Teixeira)

1. Introduction

Deep learning models have great potential to provide valuable insights that support medical diagnosis and treatment, having achieved promising results in various medical image analysis tasks. However, the training of these models requires large amounts of data, which is often difficult to obtain. Deep generative models can generate highly-realistic medical images [1, 2] and have been used to obtain large synthetic datasets to facilitate the training of models [3, 4]. Nonetheless, since generative models model the probability distribution of the data, there are concerns that the synthetic images obtained using these models may threaten the privacy of the patients whose images were used in their training. These concerns are aggravated by recent claims suggesting that it is possible to re-identify patients based on medical images such as chest radiographs [5] and magnetic resonance images [6]. In order to identify the potential privacy threats of using and sharing synthetic medical data in various real-world scenarios, a new challenge (ImageCLEFmedical GANs [7]) arose as part of the medical track of the ImageCLEF Challenge 2023 [8].

ImageCLEF is a multi-modal challenge organized as part of the CLEF Initiative Labs[1] (Conference and Labs of the Evaluation Forum) that proposes various tasks across different domains, aiming to promote the evaluation of technologies for annotation, indexing, classification and retrieval of multi-modal data. ImageCLEFmedical GANs [7] is a task of the medical track of the ImageCLEF Challenge 2023 aiming to verify whether the images generated by generative adversarial networks (GANs) [9] are sufficiently similar to the training data as to compromise its privacy. More specifically, given a set of synthetic images and a set of real images, the goal of the task is to identify which real images were used in the training of the model that generated the synthetic data. It is therefore a binary classification task, where the real images can be classified as "used" or "not used" in the training of the generative models.

Our team (VCMI team), composed of members of the Visual Computing and Machine Intelligence (VCMI) Research Group of the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC) from Porto, Portugal, approached this challenge using various methods:

1. **Similarity-based methods**: identify the real images based on their similarity to the generated images.
2. **Autoencoder-based methods**: rely on autoencoders to identify images whose probability distribution differs from the generated data through outlier detection techniques, and to compare the real and generated images based on their latent representations.
3. **Patch-based methods**: extract patches from images and apply them to identify which real images are the most similar to the generated images.

The best results were obtained with a similarity-based approach that uses Structural Similarity Index Measure (SSIM) [10] to compute the similarity between real and generated images, achieving an accuracy of 0.810 and an F1-Score of 0.802 on the classification task.

The remainder of this paper is organized as follows: section 2 provides an overview of the task and the data provided by the organisation to address the task; section 3 describes the different

[1] http://www.clef-initiative.eu (accessed on: 03-06-2023)

approaches developed to solve the task; section 4 presents the results and their discussion; and section 5 concludes this paper and recommends future work directions. The code related to this paper is publicly available in a GitHub repository[2].

2. Task Description

Given a set of images generated using diffuse neural networks [11], and a set of real images, the goal of the task is to predict which of the provided real images were used in the training of the generative model.

To achieve this task, we had access to two datasets:

1. **Development Dataset:** contains 500 generated images and 160 real images annotated according to their use in the training of the generative network. Out of the real images, 80 were used and the remaining 80 were not used during training.
2. **Test Dataset**: contains 10,000 synthetic images and 200 real images, whose classes we aim to predict. Out of the real images, 100 were used and the remaining 100 were not used during training. The proportions of used and not used images in the real data were not disclosed until the communication of the results of the challenge.

The subsets of real images are composed of axial slices of 3D computed tomography images taken from a dataset of about 8,000 lung tuberculosis patients. The size of the real data in the datasets is considerably small (160 and 200 images for the development and test dataset, respectively), making it difficult to develop deep learning models trained solely on real data.

This section presents an overview of the relationship between the probability distributions of the different subsets of each dataset, and provides an exploratory data analysis based on the similarity between the images of the different sets.

2.1. Overview of the Data Subsets

Figure 1 provides an overview of the existing images in each subset and the relationship between their probability distributions. The goal of the task is to predict $P(u \mid p)$, where u represents used images and p represents real images to which we have access.

Deep generative models model the probability distribution of the data. As such, the probability distribution of the generated data should be similar to that of the images used to train the model. However, only a subset of the used images was provided to us, along with a subset of images that were not used in the model's training. As such, the probability distribution of the provided dataset should differ from that of the generated data, assuming that the generative model used to obtain the synthetic images has a limited generalization capacity, characteristic of deep learning models trained on restricted sets of data.

One of the difficulties of the challenge is that we do not have access to the whole set of images that were used to train the model. The probability distribution of the subset of used images is not necessarily the same as the probability distribution of the whole set of used images. As such, without having access to the whole set of used images, it is difficult to predict the set

[2] https://github.com/helenaMontenegro/imageclef23-medical-gans

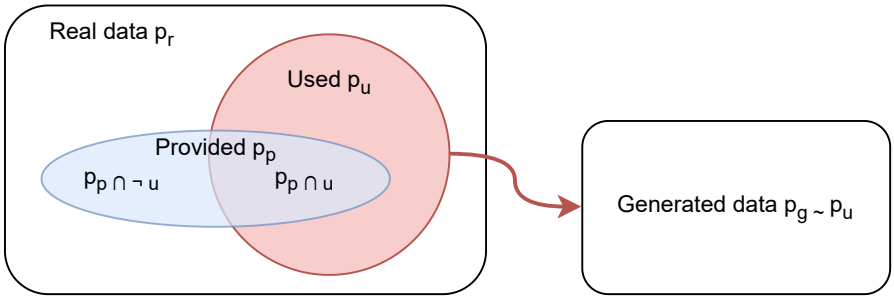


Figure 1: Overview of the relationship between the probability distribution of the task's datasets.

of real images whose probability distribution is the most similar to that of the generated data. Furthermore, some of the provided synthetic images may be similar to used images that were not provided to us, threatening their privacy. In case these used images are somewhat similar to some of the not used images that were provided to us, there is a risk that some not used images may be misclassified.

2.2. Exploratory Data Analysis

The datasets present low variability, as all the images are very similar and centered, enabling the application of similarity metrics like SSIM to compare them. As such, we computed the SSIM between the generated and real images in both the provided datasets, presenting the results in Table 1.

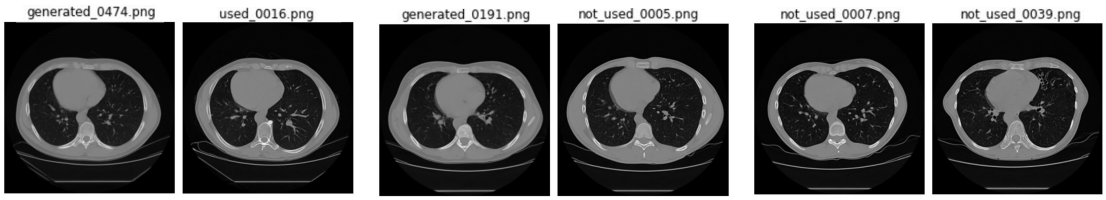
Table 1
Exploratory data analysis presenting the average, minimum and maximum SSIM between images of the subsets of the development and test datasets.

Dataset	Subsets	Average SSIM	Minimum SSIM	Maximum SSIM
Development	Real-Real	41.00%	16.59%	59.49%
	Generated-Real	42.46%	21.08%	73.94%
	Generated-Used	42.58%	27.11%	73.94%
	Generated-Not Used	42.34%	21.08%	64.39%
Test	Real-Real	40.89%	15.82%	60.27%
	Generated-Real	42.34%	17.07%	83.67%

The values of structural similarity calculated between generated images and real images are generally higher than the values calculated between real images. Furthermore, the generated images seem to be more similar to images that were used to train the generative network, than to not used images. These results suggest that some of the generated images may contain some identifiable factors of the images used during training.

Figure 2 contains the pairs of images with highest structural similarity of the development dataset. Visually, the images of the generated-used pair are very similar. The images of the

generated-not used pair and of the real-real pair present more pronounced differences.



(a) Generated-Used Pair (b) Generated-Not Used Pair (c) Real-Real Pair

Figure 2: Pairs of the most similar images between the sets of the development dataset.

3. Methodology

This section describes the methods developed to classify the real images as "used" or "not used". In specific, we describe the proposed methods organized in three groups: similarity-based methods, autoencoder-based methods and patch-based methods.

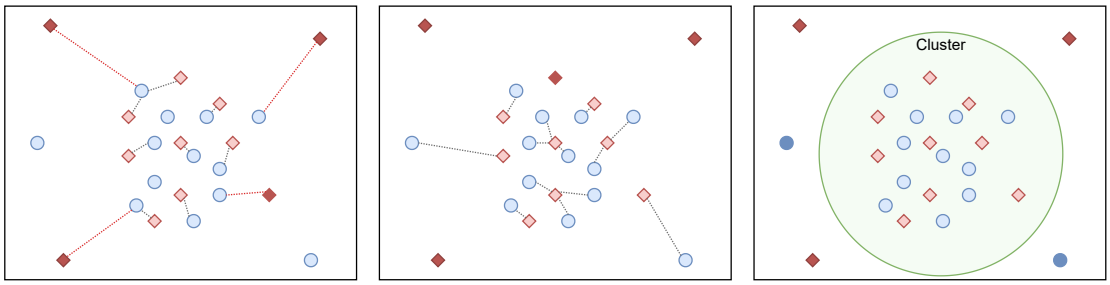
3.1. Similarity-based Methods

During the exploratory data analysis, we verified that some of the generated images are notably similar to real images, as the average SSIM and maximum SSIM between real and generated images surpass the values calculated between real images. As such, we devised various methods to classify the real images based on their similarity to the generated images. To compute the similarity between two images, we use SSIM and the Euclidean distance between the latent representations of images obtained with autoencoders (described in the following section) and with a ResNet-50 [12] model trained on ImageNet [13]. Using these metrics, we compute matrices of similarity between real and generated images, and between real images, and apply the methods described in the following subsections: threshold, retrieval, ranking, clustering and ensemble. Figure 3 presents a representative example of how the threshold, retrieval and clustering approaches work.

3.1.1. Threshold

The threshold approach finds real images whose similarity to their most similar generated image is higher than a threshold, classifying them as "used". The threshold is calculated based on the similarity between real images. We consider two thresholds: the maximum similarity between two images from the real data (MAX), and the sum between the average and standard deviation of the similarity between all real images (AVG).

Figure 3a depicts the process of searching for the synthetic images that are the most similar to each real image and verifying whether their similarity is higher (gray lines) or lower (red lines) than the threshold.



(a) Threshold (b) Retrieval (c) Clustering

Figure 3: Representative example of the similarity-based approaches. Diamonds represent real images while circles represent generated images. The lighter colors represent samples that are classified as used by the methods, while darker colors represent outliers that are classified as not used. The threshold approach checks whether the distance between each real image and its closest generated image is higher than a threshold. The retrieval approach verifies whether a real image is the closest image to any of the generated images. The clustering approach forms a cluster based on the distance between all data samples from real and generated images.

3.1.2. Retrieval

The retrieval approach finds a set of real images that are the most similar to at least one generated image, classifying them as "used". For each generated image, it retrieves the most similar real image, as depicted in Figure 3b. All retrieved images that are, therefore, the most similar to at least one of the generated images, are classified as "used". Real images that are not retrieved are classified as "not used".

3.1.3. Ranking

The ranking approach classifies real images based on a ranking that defines how similar they are to the generated images. The method starts by calculating a threshold that represents the average rank of similarity of a real image when compared with other real images. As such, for each real image, the method ranks the remaining real images based on their similarity. Then, it calculates the average rank of each real image, which is used as a threshold.

Once the threshold is set, the method ranks the real images according to their similarity to each generated image and calculates their average rank. Finally, if this average rank is higher than the threshold, then the image is classified as "used", as it shows high similarity with respect to the generated images. Otherwise, the image is classified as "not used".

3.1.4. Clustering

The clustering approach finds outliers in the data, classifying them as "not used". First, the method maps both generated and real images into a common space. Then, it uses the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [14] algorithm to form clusters, as depicted in Figure 3c, and to find outliers. In this approach, we define outliers as images whose similarity to any other data point is lower than the sum of the average similarity with

three times its standard deviation. Outliers identified in the subset of real images are classified as “not used”, while the remaining images are classified as “used”.

3.1.5. Ensemble

Since some of the proposed methods may be particularly good at identifying a specific subset of either used or not used images, we also implemented an ensemble model that merges the results of the different methods. For example, the retrieval method may be good at identifying a subset of the not used images, while the threshold method using the maximum similarity between two real images may be good at identifying a subset of the used images.

The ensemble model uses the results of ranking as a base. Then, it changes the class of the images that were classified as “used” by the threshold method to “used”, independently from the class assigned by the ranking approach. Finally, it alters the class of the images that were classified as “not used” by the retrieval approach to “not used”. Images that are simultaneously classified as “used” by the threshold method and “not used” by the retrieval method, are assigned the class “not used”.

3.2. Autoencoder-based Methods

In the autoencoder-based methods, we devised different strategies to train autoencoders. We used two main methods to classify the images using autoencoders:

- Computing the similarity between the images based on their latent representations, enabling the direct application of the techniques defined in the previous section.
- Applying outlier detection techniques to identify data points from the real data that do not follow the probability distribution of the generated data.

We experimented with two types of architectures of autoencoders:

- **Basic Autoencoder**: The encoder contains 5 stridden convolution layers with batch normalization and Leaky ReLU as the activation function. The decoder contains 5 convolution layers with upsampling, batch normalization and Leaky ReLU as the activation function.
- **ResNet Autoencoder**: The encoder and decoder are composed of 5 blocks of convolution layers with batch normalization, Leaky ReLU as the activation function, and residual layers, similar to those of a ResNet [12].

The following subsections explain in detail the approaches that we developed based on autoencoders.

3.2.1. Outlier Detection with Autoencoder trained on Generated Data

The first autoencoder-based approach consists of using an autoencoder trained on the generated data to detect outliers among the real images. We start by splitting the generated data into training (95% of the data) and validation (5% of the data). Then, we train the autoencoder on the training data for 200 epochs, using a reconstruction loss to minimize the mean squared error

between its input and output, as depicted in Figure 4. Afterwards, we apply the autoencoder to the validation data, measuring the corresponding reconstruction error, which is used to compute a threshold. In particular, we considered two thresholds: the maximum of the reconstruction error on the validation data (MAX), and the sum of the average of the reconstruction error with two times its standard deviation (AVG).

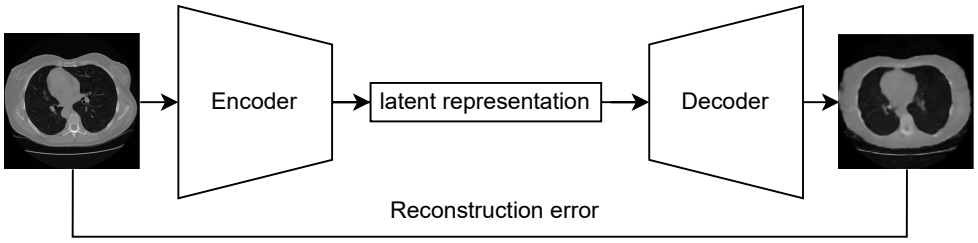


Figure 4: Overview of autoencoder for outlier detection.

Finally, we apply the autoencoder to the real data and measure the reconstruction error. Images whose reconstruction error is higher than the threshold are images whose probability distribution deviates from the probability distribution of the generated data, and are, therefore, classified as "not used". All other real images are classified as "used".

Since this network was only trained on the generated data, we do not use its latent representations to compute the similarity between real and generated images.

3.2.2. Autoencoder trained on Generated and Real Data

In a second approach, we train the autoencoder depicted in Figure 4 simultaneously with real and generated data, minimizing the reconstruction error between its input and output. On inference, we compute the latent representations of the images and calculate a similarity metric based on the mean squared error between the latent vectors, which is used to apply the previously defined similarity-based techniques.

On the test dataset, we devised two experiments. In the first experiment, we use all the 10,000 generated images in addition to the 200 real images to train the autoencoder. In a second experiment, we use only 600 images of the generated data and all the real images, to emulate the dimensions of the development dataset.

3.2.3. Autoencoder with one Encoder and two Decoders

As the final autoencoder-based approach, we train an autoencoder with one encoder and two decoders, as depicted in Figure 5. The encoder receives both generated and real data. One of the decoders receives latent features of the real data, while the other receives latent features of the generated data. In each epoch, we provide a real image and a generated image to the network, and apply each decoder to the corresponding image, minimizing the reconstruction error between the inputs of the encoder and the output of each of the decoders. Thus, the decoders are trained simultaneously.

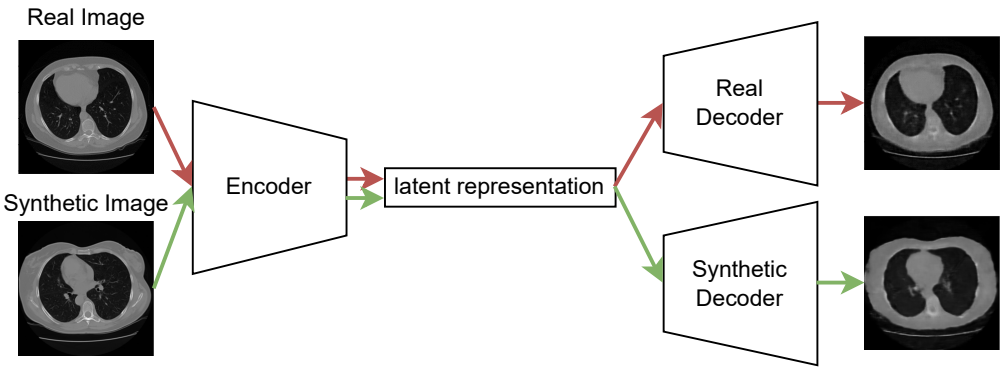


Figure 5: Overview of autoencoder with two decoders.

On inference, we apply the decoder of the generated images to the real images to detect outliers, in a similar manner to the method described in section 3.2.1. Moreover, we use the encoder to obtain latent features, which are used to calculate the similarity between images and apply the previously defined similarity-based techniques to classify the real samples. We define similarity as the opposite of the mean squared error between two feature vectors.

3.3. Patch-based Methods

The patch-based methods extract patches from images and perform the operations described in the following subsections to classify the real images.

3.3.1. Matching Patches using Triplet Loss

The first patch-based approach consists of a model that compares image patches and predicts whether they belong to the same image. It aims to verify whether a generated image is sufficiently similar to a real image so that the model predicts that their patches belong to the same image.

The model is a Convolutional Neural Network (CNN) that extracts features from image patches and calculates the distance between them. We train the network using a triplet loss that maximizes the Euclidean distance between patches from different images and minimizes the distance between patches of the same image, as depicted in Figure 6. The model is only trained on patches from real images. Since generated images are slightly blurrier than the real images, we add Gaussian noise to the latent representations of the real images during training, to emulate the lack of quality of the generated images. The network is trained for 800 epochs and then applied to compare patches from real and generated images, to verify whether patches from generated images can be identified as belonging to a real image.

On inference, we calculate the Euclidean distance between patches of generated and real images and verify whether this distance is lower than the maximum distance between patches of the same image on the real data. For each generated image, if there is at least one real image whose distance is lower than the threshold, then that real image is classified as “used”. Real images for which there is no generated image that is similar to it are classified as “not used”.

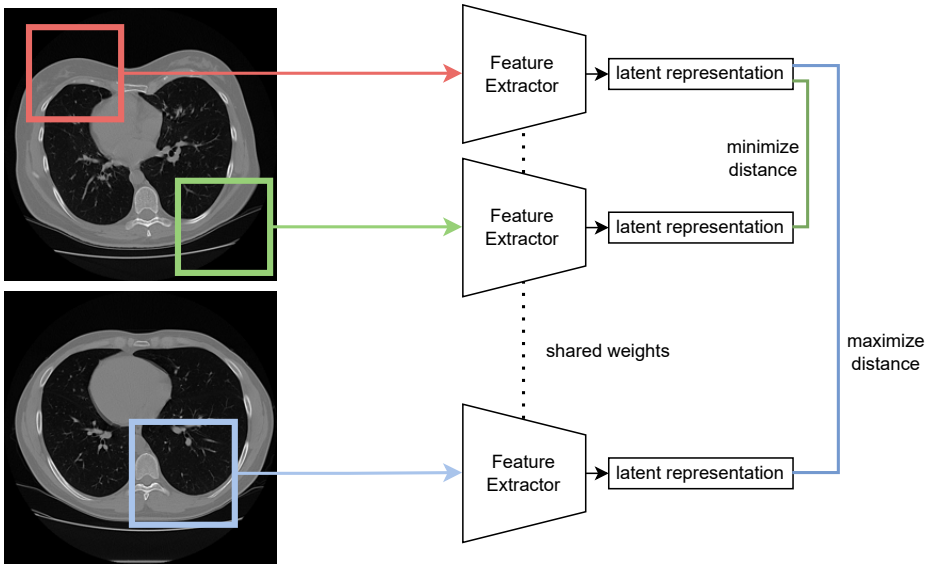


Figure 6: Overview of method that matches patches using triplet loss.

In our implementation, the feature extractor contains 5 stridden convolution layers with batch normalization and Leaky ReLU as the activation function, followed by global average pooling and one fully-connected layer with linear activation.

3.3.2. Replacing Patches

In a final approach, we extract patches from all the generated images and put these patches in the same position on the real images, modifying them. Then, we pass the modified images through the autoencoder trained on all the data (from section 3.2.2) and verify its reconstruction error, as depicted in Figure 7. Real images that contain modified images with a low reconstruction error are similar to the generated images used to build the modified images and are, therefore, classified as "used". The remaining images are classified as "not used".

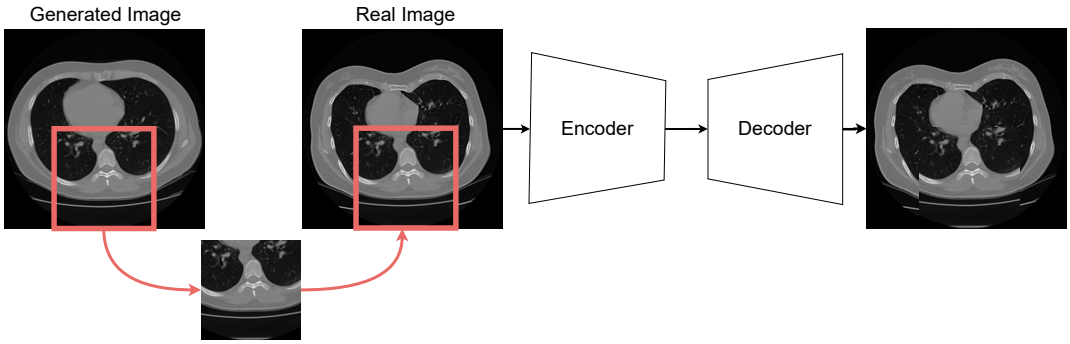


Figure 7: Overview of method that replaces patches.

4. Results and Discussion

Table 2 exposes the results in terms of accuracy, precision, specificity, recall and F1-score, obtained with each approach on the development dataset. The methods that were applied to the test data are highlighted in bold. The official metric of the competition is the F1-score.

Table 2
Results on the development dataset. Metric refers to the similarity metric that was used to compute the similarity between images in the similarity-based approaches. AE stands for autoencoder.

Method	Metric	Accuracy	Precision	Specificity	Recall	F1-score
Threshold (AVG)	SSIM	0.675	0.663	0.638	0.713	0.687
Threshold (MAX)		0.675	0.868	0.938	0.413	0.559
Retrieval		0.613	0.576	0.375	0.850	0.687
Ranking		0.650	0.650	0.650	0.650	0.650
Clustering		0.650	0.962	0.9875	0.313	0.472
Ensemble		0.731	0.740	0.750	0.713	0.726
Threshold (AVG)	MSE between Autoencoder embeddings (Simple AE)	0.644	0.626	0.575	0.713	0.667
Threshold (MAX)		0.644	0.646	0.650	0.638	0.642
Retrieval		0.600	0.569	0.375	0.825	0.674
Ranking		0.850	0.868	0.875	0.825	0.846
Clustering		0.606	1.000	1.000	0.213	0.351
Ensemble		0.713	0.677	0.613	0.813	0.739
Threshold (AVG)	MSE between Autoencoder embeddings (ResNet AE)	0.713	0.693	0.663	0.763	0.726
Threshold (MAX)		0.688	0.857	0.925	0.450	0.590
Retrieval		0.569	0.541	0.238	0.900	0.676
Ranking		0.744	0.767	0.788	0.700	0.732
Clustering		0.650	0.900	0.963	0.338	0.491
Ensemble		0.781	0.792	0.800	0.763	0.777
Threshold (AVG)	MSE between Autoencoder embeddings (2 Decoder AE)	0.613	0.610	0.600	0.625	0.617
Threshold (MAX)		0.606	0.623	0.675	0.538	0.577
Retrieval		0.550	0.534	0.325	0.775	0.633
Ranking		0.575	0.575	0.575	0.575	0.575
Clustering		0.531	1.00	1.00	0.063	0.118
Ensemble		0.613	0.602	0.563	0.663	0.631
Retrieval	MSE w/ ResNet embeddings	0.569	0.560	0.500	0.638	0.596
Ranking		0.731	0.768	0.800	0.663	0.711
Simple AE (AVG)	-	0.650	0.722	0.813	0.488	0.582
Simple AE (MAX)		0.594	0.590	0.575	0.613	0.601
ResNet AE (AVG)		0.606	0.681	0.813	0.400	0.504
ResNet AE (MAX)		0.575	0.555	0.513	0.638	0.600
2 Decoder AE (AVG)		0.613	0.641	0.713	0.513	0.570
2 Decoder AE (MAX)		0.525	0.514	0.125	0.925	0.661
Matching Patches	-	0.525	0.517	0.300	0.750	0.612
Replacing Patches	-	0.644	0.689	0.763	0.525	0.596

Comparing the similarity-based approaches, the ensemble method seems to lead to the best results across the methods using different similarity metrics to compare images. The threshold (MAX) and clustering approaches seem to be particularly good at identifying a subset of the used images, presenting high precision, suggesting that most of the images predicted as used were, indeed, used. Nonetheless, these approaches tend to present low recall, failing to detect a considerable amount of used images. The retrieval approach seems to be particularly good at detecting a subset of the not used images, as proven by its high recall value, indicating that most of the used images are classified as such. Nevertheless, its low specificity suggests that there is a substantial amount of misclassified not used images. Ranking seems to be a more balanced approach, presenting similar values of precision and recall.

The best results on the development dataset were obtained by comparing the latent representations of the simple autoencoder trained on both generated and real images, using the ranking approach. This method achieved an accuracy of 0.850 and an F1-score of 0.846. The method using the autoencoder with two decoders to compare the latent representations of the images achieved the worst results among the different metrics of the similarity-based approaches.

The methods using autoencoders for outlier detection were not capable of achieving higher results than the similarity-based methods. In these methods, using the maximum reconstruction error on the validation data as a threshold (MAX) leads to worse accuracy but better F1-score than using the average reconstruction error as a threshold (AVG).

Regarding the patch-based methods, the method that matches patches to verify whether two patches belong to the same image was incapable of distinguishing between used and not used images, achieving only 0.525 of accuracy on a balanced dataset. Its high recall and low specificity indicate that the method classifies most images as used. Nevertheless, the high recall led this method to achieve a slightly higher F1-score than the method that replaces patches of real images with ones from generated images. Replacing patches leads, however, to a considerably higher accuracy, showing a higher capacity of distinguishing between used and not used images.

Figure 8 compares the visual results of the simple autoencoder with the ResNet autoencoder. The ResNet autoencoder seems to lead to higher-quality images. Nonetheless, the simple autoencoder seems to achieve better results in terms of F1-score and accuracy than the ResNet autoencoder, across the different approaches.

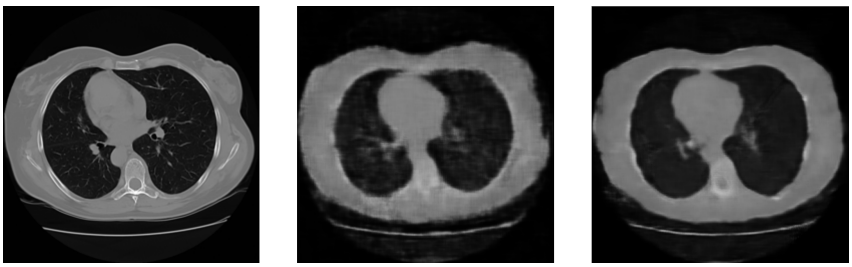


Figure 8: Reconstruction of a synthetic image (left image) using the simple (middle image) and the ResNet autoencoders (right image).

The results of accuracy, precision, specificity, recall and F1-score obtained by some of the proposed approaches on the test data are presented in Table 3. Submissions 5 and 6 represent

the same approach but with models trained with different amounts of data. In submission 5, the autoencoder was trained on all 10,000 generated images and 200 real images. In submission 6, the autoencoder was only trained with 600 generated images and 200 real images. Unlike in the development dataset, the ranking and ensemble methods obtained the same results when applied to the test data.

Table 3
Results on the test dataset. "S." refers to the submission number. AE stands for autoencoder.

S.	Method	Metric	Accuracy	Precision	Specificity	Recall	F1-Score
1	Ranking / Ensemble		0.685	0.637	0.51	0.86	0.731
2	Threshold (MAX)	SSIM	0.810	0.836	0.850	0.770	**0.802**
3	Retrieval		0.590	0.550	0.190	0.990	0.707
5	Ranking / Ensemble	Simple AE	0.635	0.645	0.670	0.600	0.621
6	Ranking / Ensemble	Simple AE	0.635	0.658	0.710	0.560	0.605
7	Ranking / Ensemble	ResNet AE	0.615	0.616	0.620	0.610	0.613
8	Ranking / Ensemble	ResNet	0.460	0.458	0.480	0.440	0.448
4	Simple AE (AVG)	-	0.720	0.854	0.910	0.530	0.654
9	Matching Patches	-	0.500	0.500	0.470	0.530	0.514
10	Replacing Patches	-	0.615	0.693	0.770	0.520	0.594

The method that achieved the best results on the test data was threshold (MAX) using SSIM as a similarity metric, achieving the highest accuracy and F1-score out of all the methods. Furthermore, ranking and retrieval using SSIM also present high F1-score, when compared to all other approaches. Retrieval seems to present high F1-Score, despite its relatively low accuracy, due to classifying most images as used, as can be seen in its high recall and low specificity.

Using a simple autoencoder to perform outlier detection led to the second-highest accuracy, but a relatively low F1-score. Ranking using the embeddings of the simple autoencoder trained with different amounts of data provided very similar results. Nonetheless, this method led to considerably worse results in the test dataset than in the development dataset, where it achieved the best results of F1-score.

The method that replaces patches obtained comparable results to these similarity-based methods using autoencoders. The worst results on the test dataset were obtained through the matching patches method and ranking using the latent representations of images obtained with a pre-trained ResNet model. Both these methods failed to distinguish between used and not used images, achieving low accuracy and F1-score.

The results on the development dataset differ substantially from the results on the test dataset, perhaps due to the difference in the dimensions of the datasets. Despite the differences in results between the test and development datasets, we were capable of developing approaches that achieved high F1-score and accuracy at identifying the used images for both sets. These results support the hypothesis that the synthetic images generated using deep generative models expose the identity of patients.

5. Conclusions and Future Work

This paper described the work developed by the VCMI team for the ImageCLEFmedical GANs task. The goal of the task was to verify whether a set of images generated using a deep generative model contained identifiable properties of the data used to train the network. As such, we proposed various approaches to solve the binary classification task of classifying a set of real images according to whether they were used in the training of the networks that generated a set of synthetic images. The experiments confirmed the hypothesis that synthetic data threatens the privacy of the training data, as some of the proposed methods achieved high accuracy and F1-score on the datasets of the challenge.

One of the limitations of the proposed methods is that they do not consider that the classification task was only applied to a subset of the real images. As such, there may be synthetic images that threaten the privacy of real images not provided to us during the challenge and that may be similar to provided not used images, leading to their misclassification. To prevent this issue, future work considers the separation of the classification process into two steps: identifying which of the generated images threaten patient privacy of the provided subset of images, by obtaining the subset of generated images whose probability distribution is the most similar to the provided real images, and matching each of those generated images to the most similar real images. Future work will also consider the further development of the proposed methods.

To conclude, this paper, as well as the ImageCLEFmedical GANs challenge, serve to raise awareness about the potential privacy risks of using and sharing synthetic medical data in real-world applications. We highlight the importance of implementing privacy-preserving techniques when developing deep generative models on sensitive medical data.

Acknowledgments

This work is financed by National Funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P. (Portuguese Foundation for Science and Technology) within the project CAGING, with reference 2022.10486.PTDC, and within PhD grants 2020.06434.BD, 2020.07034.BD, 2021.06872.BD, 2022.11566.BD, 2022.14516.BD.

References

[1] A. Beers, J. Brown, K. Chang, J. P. Campbell, S. Ostmo, M. F. Chiang, J. Kalpathy-Cramer, High-resolution medical image synthesis using progressively grown generative adversarial networks, arXiv preprint arXiv:1805.03144 (2018).

[2] P.-D. Tudosiu, T. Varsavsky, R. Shaw, M. Graham, P. Nachev, S. Ourselin, C. H. Sudre, M. J. Cardoso, Neuromorphologicaly-preserving volumetric data encoding using vq-vae, arXiv preprint arXiv:2002.05692 (2020).

[3] A. S. Coyner, J. S. Chen, K. Chang, P. Singh, S. Ostmo, R. P. Chan, M. F. Chiang, J. Kalpathy-Cramer, J. P. Campbell, Imaging, I. in Retinopathy of Prematurity Consortium, et al., Synthetic medical images for robust, privacy-preserving training of artificial intelligence:

application to retinopathy of prematurity diagnosis, Ophthalmology Science 2 (2022) 100126.
[4] M. K. Baowaly, C.-C. Lin, C.-L. Liu, K.-T. Chen, Synthesizing electronic health records using improved generative adversarial networks, Journal of the American Medical Informatics Association 26 (2019) 228–241.
[5] K. Packhäuser, S. Gündel, N. Münster, C. Syben, V. Christlein, A. Maier, Deep learning-based patient re-identification is able to exploit the biometric nature of medical chest x-ray data, Scientific Reports 12 (2022) 14851.
[6] L. C. M. Esmeral, A. Uhl, Low-effort re-identification techniques based on medical imagery threaten patient privacy, in: Medical Image Understanding and Analysis: 26th Annual Conference, MIUA 2022, Cambridge, UK, July 27–29, 2022, Proceedings, Springer, 2022, pp. 719–733.
[7] A. Andrei, A. Radzhabov, I. Coman, V. Kovalev, B. Ionescu, H. Müller, Overview of ImageCLEFmedical GANs 2023 Task – Identifying Training Data "Fingerprints" in Synthetic Biomedical Images Generated by GANs for Medical Image Security, in: CLEF2023 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Thessaloniki, Greece, 2023.
[8] B. Ionescu, H. Müller, A.-M. Drăgulinescu, W. wai Yim, A. B. Abacha, N. Snider, G. Adams, M. Yetisgen, J. Rückert, A. G. S. de Herrera, C. M. Friedrich, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, S. A. Hicks, M. A. Riegler, V. Thambawita, A. Storås, P. Halvorsen, N. Papachrysos, J. Schöler, D. Jha, A.-G. Andrei, A. Radzhabov, I. Coman, V. Kovalev, A. Stan, G. Ioannidis, H. Manguinhas, L.-D. Ştefan, M. G. Constantin, M. Dogariu, J. Deshayes, A. Popescu, Overview of ImageCLEF 2023: Multimedia Retrieval in Medical, Social Media and Recommender Systems Applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 14th International Conference of the CLEF Association (CLEF 2023), Springer Lecture Notes in Computer Science LNCS, Thessaloniki, Greece, 2023.
[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, volume 27, 2014, pp. 2672–2680.
[10] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE transactions on image processing 13 (2004) 600–612.
[11] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, Advances in neural information processing systems 32 (2019).
[12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: kdd, volume 96, 1996, pp. 226–231.