Overview of the 2024 ImageCLEFmedical GANs Task – Investigating Generative Models' Impact on Biomedical Synthetic Images

Notebook for the ImageCLEF Lab at CLEF 2024

Alexandra-Georgiana Andrei[1,*], Ahmedkhan Radzhabov[2], Dzmitry Karpenka[2], Yuri Prokopchuk[2], Vassili Kovalev[2], Bogdan Ionescu[1] and Henning Müller[3]

[1]AI Multimedia Lab, National University of Science and Technology Politehnica Bucharest, Romania
[2]Belarusian Academy of Sciences, Minsk, Belarus
[3]University of Applied Sciences Western Switzerland (HES-SO), Sierre, Switzerland

Abstract

The 2024 ImageCLEFmedical GANs task Controlling the Quality of Synthetic Medical Images created via GANs is in its second edition. It comprises two sub-tasks which address the security and privacy concerns related to personal medical image data in the context of generating and using synthetic images in different real-life scenarios. The first sub-task is an extension of the task presented in the previous edition, focusing on examining the hypothesis that generative models (e.g., GANs, Diffusion Models) generate medical images containing certain "fingerprints" of the original images used for network training. The second sub-task, new this year, explores the hypothesis that generative models imprint unique fingerprints on generated images. The focus is on understanding whether different generative models or architectures leave discernible signatures within the synthetic images they produce. Ground truth data was made available to the participants. This paper presents the overview of systems and runs submitted by describing the datasets, the evaluation metrics, and discussing the methods proposed by the participating teams and their results.

Keywords

generative models, medical synthetic data, medical imaging, Artificial Intelligence and deep learning, ImageCLEF benchmarking lab

1. Introduction

ImageCLEF [1, 2], part of the CLEF initiative[1], offers a range of multimedia information retrieval challenges. Starting from its second edition in 2004, ImageCLEF has consistently featured medical tasks annually. The 2024 ImageCLEFmedical GANs task is the second edition of the task, delving more in the privacy and security concerns related to generated medical data.

Biomedical imaging has advanced significantly in recent years, as a result of the convergence of machine learning (ML) and artificial intelligence (AI) technologies. Among these, generative models – particularly Generative Adversarial Networks (GANs) – have shown to be effective tools for producing synthetic images that mimic real biomedical images. The development of these models has created new opportunities for study and application, high-quality synthetic images have been produced in a variety of disciplines using generative models. The synthetic images produced by these models have several potential advantages in the biomedical domain. They can augment existing datasets, thereby addressing issues related to data scarcity and imbalances. This is especially helpful in the medical domain, where it can be difficult, costly, and time-consuming to obtain huge amounts of labeled data. Furthermore, AI algorithms can benefit greatly from synthetic images, reducing the dependency on real patient data and mitigating privacy concerns.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
*Corresponding author.
alexandra.andrei@upb.ro (A. Andrei); vassili.kovalev@gmail.com (V. Kovalev); bogdan.ionescu@upb.ro (B. Ionescu); henning.mueller@hevs.ch (H. Müller)

[1]https://www.clef-initiative.eu/

In this edition, we continue to study the first sub-task — "Detect generative models' "fingerprints" — proposed in the previous edition [3] focused on examining the existing hypothesis that GANs generate medical images containing certain "fingerprints" of the authentic images used for generative network training. We extended the task by investigating this hypothesis for two different generative models. Another sub-task is introduced to this second edition — Detect generative models' "fingerprints". The second sub-task explores the hypothesis that generative models imprint unique fingerprints on generated images and whether different generative models or architectures leave discernible signatures within the synthetic images they produce.

Similar to the previous year, the 2D gray-scale images being provided depict the axial slices of CT scans of tuberculosis patients taken at different stages of their treatment. In 2024, we continue to use the advanced Diffuse Models along with other Generative Adversarial Networks (GANs) for image generation.

In this paper, we present an overview of the 2024 ImageCLEFmedical GANs task, describing the objective of the two sub-tasks, datasets, evaluation metrics and the results and methods proposed by the participant teams. The article is organized as follows: Section 2 introduces the 2 sub-tasks by presenting the extended version of the task presented in the previous edition and the new one introduced for this edition together with the data used for these sub-tasks. Section 3 presents the evaluation metrics, Section 4 and Section 5 present the results obntained by the participant teams and the paper concludes with Section 6.

2. Tasks description

2.1. Sub-task 1. Identify training data fingerprints

2.1.1. Description

We continued to investigate the hypothesis that generative models are generating medical images that are in some way similar to the ones used for training. The task addresses the security and privacy concerns related to personal medical image data in the context of generating and using artificial images in different real-life scenarios. This edition, in addition to the Diffusion Model used in the previous iteration of the task, we have also used a GAN model. The objective of the task was to detect "fingerprints" within the synthetic biomedical image data to determine which real images were used in training to produce the generated images. The task consisted in performing analysis of test image datasets and assessment of the probability with which certain images of real patients were used for training image generators and which were not. The task is formulated as follows:

- *given two sets that contains generated and real images, the participants are requested to employ machine learning and/or deep learning models to determine for each set which of the real images were used to train the model to generate the provided synthetic images.*

2.1.2. Data description

The benchmarking image data consists of axial slices of 3D CT images extracted from a bigger dataset of about 8,000 lung tuberculosis patients. Considering this, some of the slices may appear pretty "normal" whereas the others may contain certain lung lesions including severe ones. These images are stored in the form of 8-bit/pixel PNG images with dimensions of 256×256 pixels. The artificial slice images are 256×256 pixels in size and are obtained using two type of generative models. Examples of real and generated images using the two generative models are provided in Figure 1.

Development dataset: comprises data for the two different generative models organized as follows:

- Model 1 (representing the ground truth for the test dataset of the previous edition [3]) consists of 10k generated images and 200 images annotated as used/not used for training to generate the images. Specifically, 100 images were utilized for training, while the remaining 100 were not.

(a) real images

(b) synthetic images generated using Model 1

(c) synthetic images generated using Model 2

Figure 1: Examples of images provided for the first sub-task "Identify training data fingerprints".

- Model 2 consists of 10k generated images and 6k annotated images marked as used/not used for training to generate the images. Specifically, 3k images were utilized for training, while the remaining 3k were not.

Test dataset: has been structured similarly to the development dataset, with a key distinction. In this iteration, the two subsets of real images have been mixed, with no disclosed proportion between unused and used ones. The dataset is organized as follows:

- Folder 1: 7,200 generated images and 4,000 real images labeled as "real _unknown_1_%6d".
- Folder 2: 5,000 generated images and 4,000 real images labeled as "real _unknown_2_%6d".

2.2. Sub-task 2. Detect generative models' fingerprints

2.2.1. Description

The second sub-task explores the hypothesis that generative models imprint unique fingerprints on generated images. The focus is on understanding whether different generative models or architectures leave discernible signatures within the synthetic images they produce. By providing a set of synthetic images generated through various generative models, the objective is to identify and detect the distinct "fingerprints" associated with each model. The number of clusters - the number of models used for generating synthetic data - used for the train and development was different, as described below. This task supposes analyzing the characteristics, patterns, or features embedded in the synthetic images. The goal is not only to distinguish between images created by different models but also to uncover the specific traits that define each model's output. This investigation contributes to a deeper understanding of the unique imprint left by generative models on the images they generate, allowing model attribution recognition. The task is formulated as follows:

- *given a set of generated images and the number of generative models used, the participants are required to group the images based on the model that generated them.*

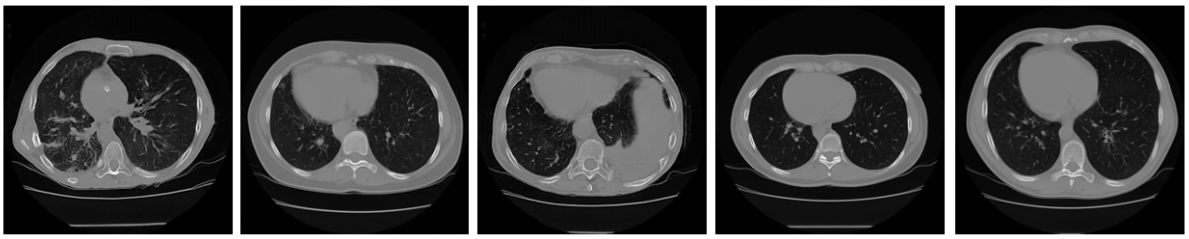
2.3. Data description

The dataset comprise synthetic CT slice images, each with a resolution of 256×256 pixels, generated using various generative models. The data used for training was extracted from the same dataset dataset of approximately 8000 lung tuberculosis patients. Examples of generated images are depicted in Figure 2.

Development dataset: consists of 600 images generated using three different generative models. Each model is represented by 200 images and are organized in annotated folders.

Test dataset: comprises of 3000 generated CT slices generated using four generative models. In addition to the three models used for the development dataset, another GAN was also used.

(a) synthetic images generated using Model 1



(b) synthetic images generated using Model 2,

(c) synthetic images generated using Model 3

(d) synthetic images generated using Model 4

Figure 2: Examples of images from the provided dataset for the second sub-task "Detect generative models' fingerprints".

Table 1
Overview of participating teams that submitted at least one run (* task organizing team)

Team	Sub-task 1	Sub-task 2	Affiliation	Country
SDVA/UCSD	✓	✓	San Diego VA Health Care System	US
Csmorgan	✓	✓	Morgan State University	US
Biomedical Imaging Goa	✓	✓	Goa College of Engineering	India
KDElab	✓	×	Toyohashi University of Technology	Japan
Inoue Koki	✓	×	Toyohashi University of Technology	Japan
Robot	✓	×	Guangxi Key Laboratory of Digital Infrastructure	China
Shitongcao	✓	×	School of Information Science and Engineering	China
KDE-med-lab	✓	✓	Toyohashi University of Technology	Japan
GAN-Amis	×	✓	Pune Institute of Computer Technology	India
AI Multimedia Lab*	✓	✓	Politehnica University of Bucharest	Romania

3. Evaluation Methodology

3.1. Sub-task 1. Identify training data fingerprints

The sub-task was assessed as a binary-class classification challenge, and the F1-score, accuracy, precision, recall, and specificity are used as evaluation metrics. The F1-score serves as the official evaluation metric for this year's edition. The metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$F1 - score = \frac{Precision \cdot Recall}{Precision + Recall} \tag{5}$$

where TP stands for true positive, TN for true negative, FP for false positive, and FN for false negative. The evaluation metrics were computed for each model individually, and the leaderboard was compiled in ascending order of the average F1-score obtained for the two models.

3.2. Sub-task 2. Detect generative models' fingerprints

Adjusted Rand Index (ARI) is the official metric of the sub-task [4, 5]. It computes a similarity measure between two clusterings, the clusters assigned by an algorithm and the ground truth labels, accounting for the possibility of randomness in clustering assignments. On a scale of -1 to 1, an ARI around 1 indicates a high degree of agreement, whilst values near 0 point to random grouping. Scores that are negative indicate a discrepancy between the two groups. When analyzing clustering algorithms in a variety of fields, including the social sciences and biology, ARI is a preferred metric since it provides a dependable assessment of clustering quality by controlling for chance.

4. Participants Runs

Overall, the same 32 teams registered to the both sub-tasks. Among them, 10 teams completed the first sub-task and submitted their runs, while 7 teams completed the second sub-task (including the task organizing team). Notably, 6 teams were common to both sub-tasks. This indicates that 31.25% of the registered teams completed the first sub-task, while 21.87% completed the second sub-task. Table 1 presents a short overview of the participating teams. When it comes to submitting the working notes, one team did not submit them, resulting in an adherence rate of 90.90%. For the first sub-task, 69 runs were submitted, of which 54 are valid and included in this overview. For the second sub-task, 56 runs were submitted, with 46 valid runs included here.

4.1. Sub-task 1. Identify training data fingerprints

Each participant team could submit up to 15 runs. The ranking according to the mean value of the F1-scores for the two models is presented in Table 2. This section briefly describes the methods proposed by the 9 participating teams for the first sub-task and the 6 participating teams for the second sub-task. Further details about each method can be found in the respective team's working note paper.

Inoue Koki [6] used various image processing techniques to enhance the distinct features of generated images, particularly focusing on boundary sharpness, which they state is less clear in synthetic images compared to real ones. The preprocessing steps include binarization, histogram equalization, Laplacian process, and contrast adjustment. Each of these methods aims to highlight different aspects of the images that might help differentiate between real and generated images. ResNet-152 was employed for binary classification. Their proposed method involves training five different models: one without any preprocessing and four with each of the mentioned image-processing techniques. The predictions from these models are then integrated to form a single final prediction. This integration is achieved through two strategies: majority voting and perfect agreement. In majority voting, the final prediction is the most common result among the five models. In perfect agreement, a positive result is accepted only if all five models agree; otherwise, a negative result is assumed. All results obtained by the team are shown in Table 2 and consist in the following methods:

- Submission ID 896: Non-Preprocessed input
- Submission ID 895: Binarization
- Submission ID 894: Constrast adjustment
- Submission ID 892: Histogram equalization
- Submission ID 891: Laplacian process
- Submission ID 893: Majority voting
- Submission ID 890: Perfect agreement.

SDVAHCS/UCSD [7] solved the first sub-task after the second in order to gain significant insights that helped with proposing the methods. Painters embedding were used. Embeddings were extracted from the synthetic images, and they were paired with embeddings of a sample of the training images (both used and not used for training). Multiple machine learning models were trained. All results obtained by the team are shown in Table 2 and consist in the following methods:

- Submission ID 851: a random number generator was used to assign classification
- Submission ID 850: each generated image was matched with approximately 5 used and 5 not used images
- Submission ID 849: each generated image was matched with approximately 10 used and 10 not used images
- Submission ID 848: each generated image was matched with approximately 50 used and 50 not used images

Robot [8] started by performing data visualization analysis to understand the relationship between the provided real and generated data. Using histogram visualization comparisons, they concluded that the generated images and the real images are highly similar. They proposed a method for calculating feature similarity to enhance image recognition and classification. After extracting features using a pre-trained Masked Autoencoder (MAE) network, they measured the similarity between images in the feature space using metrics such as Euclidean distance, cosine similarity, and Manhattan distance. The process involves acquiring feature vectors through the MAE encoder, selecting an appropriate similarity measure based on the task, calculating similarity scores, and using these scores for classification. For feature extraction, they used pre-trained models including VGG, InceptionNet, ResNet50, ResNet101, MobileNetV2, MobileNetV3, EfficientNet, and MAE, with classification performed using similarity calculations. All results obtained by the team are shown in Table 2 and the following methods were used:

- Submission ID 840: ResNet50
- Submission ID 841: EfficientNet
- Submission ID 842: MobilenetV2
- Submission ID 843: MobileNetV3
- Submission ID 844: ResNet101
- Submission ID 845: MAE
- Submission ID 846: EfficientNet
- Submission ID 847: VGG InceptionNet

KDElab [9] proposed an interesting preprocessing step. They used tools from tensorflow library to colorize the images and geometric augmentation by zooming, rotation, height and width shifts. They tested multiple methods, but the submitted run involved using a MobileNetV2 for feature extraction and classification.

KDE-med-lab [10] proposed fine-tuning deep neural network models using a two-stage transfer learning approach. This dual-stage process aims to leverage diverse data sets to enhance the model's ability to generalize. For the baseline model, DenseNet-121 is employed without applying any masks to lung images. After extracting features using the CNN, these features are processed using k-means clustering to classify the images into two categories. In the proposed model, additional preprocessing was performed by applying masks to lung images using a U-net architecture, which helps in focusing on the relevant regions of the images. This method utilizes a more comprehensive set of deep neural network models, including ResNet18, DenseNet-121, Inception-ResNet V2, EfficientNetB0, and Inception V3. The features extracted by these CNNs are also processed using k-means clustering to predict the two classes of features. All results obtained by the team are shown in Table 2 and consists in the following methods:

- Submission ID 852: ResNet18 + U−net
- Submission ID 853: Densenet 121 (Baseline)
- Submission ID 854: Inception-Resnet V2 + U−net
- Submission ID 855: EfficientNetB03 + U−net
- Submission ID 856: Densenet 121 + U−net
- Submission ID 857: Inception V3 + U−net

Biomedical Imaging Goa [11] assumed that the images used for generation in the training and test datasets are similar. They employed a pre-trained ResNet50 CNN to extract features from the images, resulting in 2048-dimensional vectors. These vectors were then analyzed using a k-means clustering approach, with the Manhattan distance used as a similarity measure to determine the closest cluster center. During testing, each test image is compared with the 2×k cluster centers formed during training, and the label of the closest cluster is assigned to the test image. Multiple runs with different values of k (1, 2, 4, 8, 16, 32) were submitted, and the best performance was achieved with k=4.All results obtained by the team are shown in Table 2 and consist in the following methods:

- Submission ID 898: 4 clusters
- Submission ID 877: 2 clusters
- Submission ID 876: 1 cluster
- Submission ID 875: 8 clusters
- Submission ID 874: 16 clusters
- Submission ID 873: 32 clusters

Csmorgan [12] started with reducing noise in the CT images through morphological operations. This noise reduction is followed by using BLIP and DINOv2 as image signature generators to improve the quality of synthetic images. Features are ranked individually and after concatenation, dimensionality reduction is performed. Late fusion is then employed to refine the fingerprint identification results, combining the strengths of various features and methods. Different methods were proposed: *i) Additive Mode Thresholding* – this technique is implemented to enhance image processing by considering local variations in image intensity. Principal Component Analysis (PCA) is used to reduce the dimension of the feature vector, and the features are then combined and weighted by the total. The mode of this weighted result serves as the threshold. For the test images, a similar weighting approach is applied: if the weighted value is less than the mode, the image is tagged as not used; otherwise, it is tagged as used. *ii) Additive Average Thresholding* – calculates the final result for each subject and averages these results across all subjects to determine the threshold value for classification. This average threshold aims to create a more generalized classification method that can effectively handle the overall distribution of the data, providing a robust approach for classifying images. *iii) Encoder Model with Dual Thresholding* – an encoder model is used to manage the extensive feature set generated by the backbone models. The encoder compresses this concatenated feature set, reducing its dimensionality. With the reduced feature set, both mode and mean thresholding techniques are applied. *iv) Late Fusion with Majority Voting* – employs a late fusion strategy to combine the decisions from the previous methods. Late fusion aggregates results at the decision level rather than at the feature level. Majority voting is used to finalize the classification, ensuring that the combined decisions of the different methods provide a more accurate and reliable result. *v) Reranking with Agglomerative Clustering* – conducts hierarchical clustering with a bottom-up approach, allowing for the specification of parameters such as the number of clusters, distance metric, and linkage criterion. The re-ranking is based on decisions from the previous submissions, further refining the classification results by leveraging hierarchical clustering techniques. All results obtained by the team are shown in Table 2 and consists in the following methods:

- Submission ID 886: Dinov2 model with additive mode thresholding
- Submission ID 884: Blip architecture with additive average thresholding
- Submission ID 883: Concatenated multiformer feature fusion
- Submission ID 881: concatenated multiformer feature fusion
- Submission ID 879: Re-ranking technique
- Submission ID 878: Re-ranking technique

Shitongcao [13] employed a similarity-based classification method, categorizing real images based on their similarity to generated images. They proposed three different methods to calculate similarity. i) they directly computed the similarity between the generated images and real images by comparing their pixel values. ii) Another approch was to apply noise (Gaussian, salt and pepper) to the original images and then calculated the similarity between the noisy images and the real images. iii) Extracted features from the images using advanced deep learning models to obtain high-dimensional features and then calculated the similarity between these features. All results obtained by the team are shown in Table 2 (there was no reference in team's working notes to the method used to obtain the results provided for the other submitted runs) and consists in the following methods:

- Submission ID 834: Cosine simiarity
- Submission ID 836: Euclidian distance
- Submission ID 838: Structual similarity index

AI Multimedia Lab [14] proposed two different approaches for identifying training data fingerprints. The first method consists in isolating and analyzing medically relevant regions in target images. The process involves three main stages: medical image segmentation to detect lung areas, deep feature extraction using pre-trained neural networks like ResNet50 and DenseNet121, and clustering to analyze potential clusters of images. The segmentation was performed using a UNet deep neural network, while features are extracted from ResNet50 and DenseNet121, followed by clustering using k-means and hierarchical methods. Their hypothesis was that generative models trained on this data will produce artificial images closely associated with clusters formed during training. Various distances and numbers of clusters are tested to optimize performance on both training and testing sets. This comprehensive approach allows for the identification of subtle fingerprints within the training data. The second approach applied generative models - using autoencoders to capture the reconstruction ability of the training dataset. Comprising an encoder and a decoder, the autoencoder transforms input samples into a condensed feature vector and endeavors to reconstruct the input faithfully. The team employs mean square error (MSE) as a metric to assess reconstruction quality, aiming for minimal MSE values, indicative of a well-adapted autoencoder. They hypothesize that an autoencoder trained on generated samples should exhibit low reconstruction errors for training data and higher errors for samples outside the training set. The team explores two distinct approaches: in the first, they compute the mean MSE at the pixel level to differentiate between used and not-used training samples, while in the second, they compute pixel-level MSE for individual inputs and compare reconstruction errors with centroid-like images of used and not-used training samples using Structural Similarity Index (SSIM). This methodology enables the identification of anomalies in generated data based on reconstruction errors, offering insights into the model's performance and dataset coverage.

All results obtained by the team are shown in Table 2 (there was no reference in team's working notes to the method used to obtain the results provided for the other submitted runs) and consists in the following methods:

- Submission ID 901: Analyzing medically relevant region method using full images and DenseNet
- Submission ID 902: Analyzing medically relevant region method using full images and ResNet
- Submission ID 903: Analyzing medically relevant region method using lung regions and ResNet
- Submission ID 904: Analyzing medically relevant region method using lung regions and DenseNet
- Submission ID 905: Analyzing medically relevant region method using outer regions and ResNet
- Submission ID 908: Analyzing medically relevant region method using outer regions and DenseNet
- Submission ID 908: Anomaly detection applied to generative models method using averge MSE
- Submission ID 909: Anomaly detection applied to generative models method using SSIM

Table 2: Summary on the participant submissions and their results for the first sub-task – Identify training data fingerprints. Acc stands for Accuracy, Prec for Precision, and F1 for F1-score.

				M1					M2			
Rank	Team	ID #	Run	F1	Acc	Prec	Recall	F1	Acc	Prec	Recall	F1
#1	Inoue Koki	892	5	0.666	0.499	0.499	0.998	0.665	0.501	0.5	0.999	0.667
#2	Inoue Koki	896	1	0.663	0.495	0.497	0.987	0.661	0.5	0.5	0.996	0.665
#3	Inoue Koki	891	6	0.663	0.492	0.496	0.979	0.658	0.505	0.502	0.998	0.668
#4	Inoue Koki	894	3	0.66	0.491	0.495	0.973	0.656	0.499	0.499	0.993	0.664
#5	Inoue Koki	895	2	0.638	0.484	0.49	0.838	0.619	0.503	0.501	0.951	0.656
#6	Inoue Koki	890	8	0.631	0.473	0.484	0.805	0.604	0.508	0.504	0.945	0.657
#7	AI Multimedia Lab	909	8	0.627	0.499	0.499	0.977	0.661	0.523	0.517	0.697	0.593
#8	Inoue Koki	893	4	0.626	0.466	0.48	0.819	0.605	0.517	0.509	0.883	0.646
#9	SDVAHCS/UCSD	848	3	0.624	0.51	0.506	0.847	0.633	0.576	0.563	0.676	0.614
#10	SDVAHCS/UCSD	849	4	0.606	0.515	0.509	0.827	0.63	0.538	0.531	0.642	0.581
#11	Robot	844	4	0.603	0.711	0.824	0.538	0.651	0.504	0.503	0.619	0.555
#12	Shitongcao	834	6	0.598	0.614	0.599	0.689	0.641	0.504	0.503	0.622	0.556
#13	Shitongcao	836	4	0.598	0.615	0.6	0.688	0.641	0.504	0.503	0.619	0.555

Table 2: Summary on the participant submissions and their results for the first sub-task – Identify training data fingerprints. Acc stands for Accuracy, Prec for Precision, and F1 for F1-score.

				M1				M2				
Rank	Team	ID #	Run	F1	Acc	Prec	Recall	F1	Acc	Prec	Recall	F1
#14	AI Multimedia Lab	905	5	0.538	0.529	0.529	0.529	0.529	0.547	0.547	0.547	0.547
#15	Biomedical Imaging Goa	898	3	0.531	0.538	0.543	0.488	0.514	0.533	0.531	0.569	0.549
#16	Shitongcao	838	2	0.529	0.711	0.824	0.538	0.651	0.593	0.751	0.279	0.407
#17	AI Multimedia Lab	906	6	0.527	0.524	0.524	0.524	0.524	0.529	0.529	0.529	0.529
#18	Robot	841	7	0.524	0.615	0.6	0.688	0.641	0.593	0.751	0.279	0.407
#19	AI Multimedia Lab	903	3	0.515	0.496	0.496	0.496	0.496	0.534	0.534	0.534	0.534
#20	Biomedical Imaging Goa	875	4	0.515	0.529	0.538	0.408	0.464	0.56	0.559	0.572	0.565
#21	SDVAHCS/UCSD	850	1	0.511	0.509	0.509	0.51	0.509	0.513	0.513	0.512	0.512
#22	KDE-med-lab	852	6	0.51	0.493	0.495	0.652	0.562	0.502	0.502	0.42	0.457
#23	AI Multimedia Lab	904	4	0.51	0.514	0.514	0.514	0.514	0.506	0.506	0.506	0.506
#24	Robot	840	8	0.503	0.503	0.504	0.409	0.451	0.504	0.503	0.619	0.555
#25	AI Multimedia Lab	902	2	0.502	0.476	0.476	0.476	0.476	0.528	0.528	0.528	0.528
#26	SDVAHCS/UCSD	851	2	0.501	0.524	0.531	0.401	0.457	0.525	0.523	0.57	0.545
#27	csmorgan	881	4	0.5	0.495	0.495	0.512	0.503	0.497	0.497	0.496	0.496
#28	csmorgan	884	2	0.5	0.504	0.504	0.515	0.509	0.499	0.499	0.483	0.491
#29	AI Multimedia Lab	901	1	0.499	0.48	0.48	0.48	0.48	0.519	0.519	0.519	0.519
#30	Biomedical Imaging Goa	874	5	0.499	0.533	0.549	0.373	0.444	0.557	0.558	0.55	0.554
#31	Biomedical Imaging Goa	873	6	0.497	0.508	0.51	0.396	0.446	0.538	0.536	0.564	0.549
#32	csmorgan	883	3	0.496	0.492	0.492	0.486	0.489	0.5	0.5	0.509	0.504
#33	csmorgan	886	1	0.492	0.502	0.502	0.489	0.495	0.491	0.491	0.487	0.489
#34	KDE-med-lab	854	4	0.488	0.504	0.504	0.422	0.459	0.527	0.528	0.505	0.516
#35	csmorgan	879	5	0.483	0.495	0.495	0.485	0.49	0.506	0.506	0.45	0.476
#36	csmorgan	878	5	0.47	0.51	0.511	0.449	0.478	0.495	0.494	0.436	0.463
#37	Shitongcao	833	7	0.462	0.615	0.6	0.688	0.641	0.559	0.761	0.174	0.283
#38	KDE-med-lab	857	1	0.46	0.468	0.47	0.505	0.487	0.479	0.475	0.398	0.433
#39	KDE-med-lab	853	5	0.455	0.506	0.509	0.331	0.401	0.497	0.497	0.521	0.509
#40	KDElab	897	1	0.454	0.484	0.476	0.317	0.38	0.481	0.484	0.579	0.527
#41	Shitongcao	835	5	0.451	0.584	0.803	0.222	0.348	0.504	0.503	0.617	0.554
#42	Shitongcao	839	1	0.448	0.583	0.812	0.216	0.341	0.504	0.503	0.619	0.555
#43	KDE-med-lab	856	2	0.443	0.502	0.503	0.314	0.386	0.49	0.49	0.511	0.5
#44	Biomedical Imaging Goa	876	1	0.43	0.505	0.511	0.239	0.325	0.527	0.526	0.544	0.534
#45	Robot	845	3	0.429	0.503	0.504	0.409	0.451	0.593	0.751	0.279	0.407
#46	Biomedical Imaging Goa	877	2	0.385	0.504	0.507	0.29	0.369	0.501	0.501	0.335	0.402
#47	Robot	846	2	0.35	0.615	0.6	0.688	0.641	0.504	0.579	0.031	0.058
#48	Robot	842	6	0.314	0.583	0.812	0.216	0.341	0.559	0.747	0.178	0.287
#49	Robot	843	5	0.312	0.583	0.812	0.216	0.341	0.559	0.761	0.174	0.283
#50	Robot	847	1	0.312	0.583	0.812	0.216	0.341	0.559	0.761	0.174	0.283
#51	Shitongcao	837	3	0.255	0.503	0.504	0.409	0.451	0.504	0.579	0.031	0.058
#52	AI Multimedia Lab	908	7	0.235	0.5	1	0.001	0.002	0.497	0.497	0.442	0.468
#53	Shitongcao	832	8	0.2	0.583	0.812	0.216	0.341	0.504	0.579	0.031	0.058
#54	KDE-med-lab	855	3	0.019	0.5	0.545	0.003	0.005	0.497	0.443	0.017	0.033

4.2. Subtask 2. Detect generative models' fingerprints

Each participant team could submit up to 10 runs. The results are presented in Table 3, and arranged in ascending order according to the ARI value.

SDVAHCS/UCSD [7] used all embedders included with Orange3 (SqueezeNet, Inception V3, VGG-16, VGG-19, Painters, DeepLoc) and k-means clustering and two-dimensional data projection with t-SNE using widgets provided by Orange3 were used for clustering. Another approach proposed by the team was using CNNs as ResNet18, ResNet34 and ResNet50 pre-trained models. All results obtained by the team are shown in Table 3 and consist in the following methods:

- Submission ID 545: t-SNE clustering on Painters embeddings

- Submission ID 550: ensamble method combining ResNet18, ResNet34 and ResNet50 models trained on pseudo-labeled data
- Submission ID 590: ResNet34 model trained on pseudo-labeled data
- Submission ID 548 and 549: ResNet50 and ResNet18 trained on 224×224 images from the training set, and after pseudo-labeling the test data, they were trained on a combination of the trainig data and 200 pseudo-labeled images
- Submission ID 547 and 225: ResNet34 network trained only on the training data, without incorporating pseudo-labels
- Submission ID 546: k-means clustering algorithm

Biomedical Imaging Goa [11] used a pre-trained ResNet-50 for feature extraction. Furthermore, they are clustered using Gaussian Mixture Model (GMM). The training involved experimenting with GMMs that had 3 components, while during testing, the GMMs were assumed to have 4 components. Different methods were used to initialize the means of the components in the GMMs, including k-means, k-means++, random initialization, and random selection from the data. All results obtained by the team are shown in Table 3 and consists in the following methods:

- Submission ID 307: k–means clustering algorithm for initialization
- Submission ID 321: k—means++ clustering algorithm in which the first clusters are chosen randomly while the cluster centers in the subsequent iterations are chosen based on the maximum squared distance
- Submission ID 324: chooses the component means randomly
- Submission ID 323: chooses random data points as component means

KDE-med-lab [10] employed the K-means algorithm to cluster features extracted from various CNNs, utilizing unsupervised learning. The baseline method employs DenseNet-121, while the proposed method combines five different deep neural network models: ResNet18, DenseNet-121, Inception-ResNet V2, EfficientNetB0, and Inception V3. Both methods use K-means clustering to identify intrinsic groups within the unlabeled dataset and draw inferences from these groups. In the proposed model, lung images are preprocessed using a U-net to apply masks, enhancing the focus on relevant areas of the images before passing the features through K-means clustering. This preprocessing step aims to improve the model's ability to identify meaningful patterns in the data, thereby enhancing the performance of unsupervised learning. The results achieved by the team are presented in Table 3 and consist in the following methods:

- Submission ID 270: Densenet 121 (baseline method)
- Submission ID 237: Densenet 121 + U–net
- Submission ID 257: Inception-Resnet V2 + U–net
- Submission ID 480: ResNet18 + U–net
- Submission ID 254: EfficientNetB03 + U–net
- Submission ID 271: Inception V3 + U–net

GAN-Amis [15] developed a methodology to detect fingerprints CNNs and various pre-trained deep learning models. They first constructed and preprocessed the dataset, normalizing pixel values and one-hot encoding the labels for three classes. Their custom CNN architecture included multiple convolutional layers with increasing filters, batch normalization, ReLU activation, and max pooling, followed by fully connected layers and a softmax output layer. The model was trained with the Adam optimizer and categorical cross-entropy loss over 200 epochs. The final layer was removed to use the penultimate layer's activations as feature extractors, which were then clustered using K-means to identify groups corresponding to different generative models. Additionally, the authors employed pre-trained models—EfficientNet, ResNet50, MobileNetV2, VGG19, and Xception—fine-tuning them on the development dataset. They removed the final classification layers to use the extracted features for K-means clustering. This multi-architecture approach aimed to enhance the robustness of their methodology by leveraging the strengths of different models to detect unique fingerprints left by

generative models in the synthetic lung CT images. The results achieved by the team are presented in Table 3 and consist in the following methods:

- Submission ID 520: EfficientNet
- Submission ID 518: MobileNetV2
- Submission ID 517: Xception
- Submission ID 516: ResNet50
- Submission ID 513: VGG19
- Submission ID 277: custom CNN

Csmorgan [12] aimed to identify fingerprints of generative models in CT images through a multi-step approach. Initially, noise was reduced in the CT images using morphological operations. They then employed pre-trained BLIP2 and DINOv2 architectures for feature extraction. The results achieved by the team are presented in Table 3 and consist in the following methods:

- Submission ID 446: a combination of feature sets from BLIP2 and DINOv2, referred to as the 'multiformer' architecture, was used with various augmentation techniques (center cropping, random affine transformations, resizing, and normalizing). K-means and agglomerative clustering were then used to assign labels to each subject.
- Submission ID 447: a combination of feature sets from BLIP2 and DINOv2, referred to as the 'multiformer' architecture, was used with various augmentation techniques (random cropping, horizontal flipping, rotation, and color jittering to introduce variations and improve robustness). K-means and agglomerative clustering were then used to assign labels to each subject.
- Submission ID 451 and 452: a combination of feature sets from BLIP2 and DINOv2, applied Principal Component Analysis (PCA) and autoencoders for feature dimensionality reduction, respectively, before using the same clustering algorithms for label assignment.
- Submission ID 453: BLIP base model for feature extraction. The normalized feature sets were then clustered for labeling.
- Submission ID 454: LIP pre-trained ViT large model for feature extraction. The normalized feature sets were then clustered for labeling.
- Submission ID 456 and 458: ensemble voting and reranking based on the decisions from previous submissions. Ensemble voting combined results at the decision level using majority voting to determine the final classification, enhancing accuracy and reliability. Reranking employed Density-Based Spatial Clustering (DBSCAN) to identify clusters, ensuring each point within a cluster had a dense neighborhood, thereby separating dense regions from sparser areas.

AI Multimedia Lab [14] built upon the method the method proposed in the previous edition [16] and proposed various methods to detect the "fingerprints" of generative models in synthetic images. The approach involved pattern recognition and feature extraction to analyze the embedded features in generated images. They used two main feature extraction methods: transfer learning with pre-trained models and a handcrafted technique. The pre-trained models, originally trained on ImageNet, included VGG-16, ResNet50, MobileNetV2, EfficientNet, and DenseNet-121. These models were selected for their efficacy in capturing complex patterns and hierarchical features. The extracted features were then reduced in dimensionality using Principal Component Analysis (PCA) to manage high dimensionality. Additionally, a handcrafted method using Local Binary Pattern (LBP) was employed for extracting local spatial patterns and grayscale contrast. For clustering, the authors used k-means and hierarchical clustering to group images based on similarity. They compared the clustering outcomes to validate the effectiveness of the feature extraction methods.The results achieved by the team are presented in Table 3 and consist in the following methods:

- Submission ID 327: feature extraction using DenseNet-121 and hierarchical clustering
- Submission ID 326: feature extraction using DenseNet-121 and k–means for clustering
- Submission ID 328: handcrafted feature extraction –LBP – and hierarchical clustering
- Submission ID 329: handcrafted feature extraction –LBP – and k–means for clustering

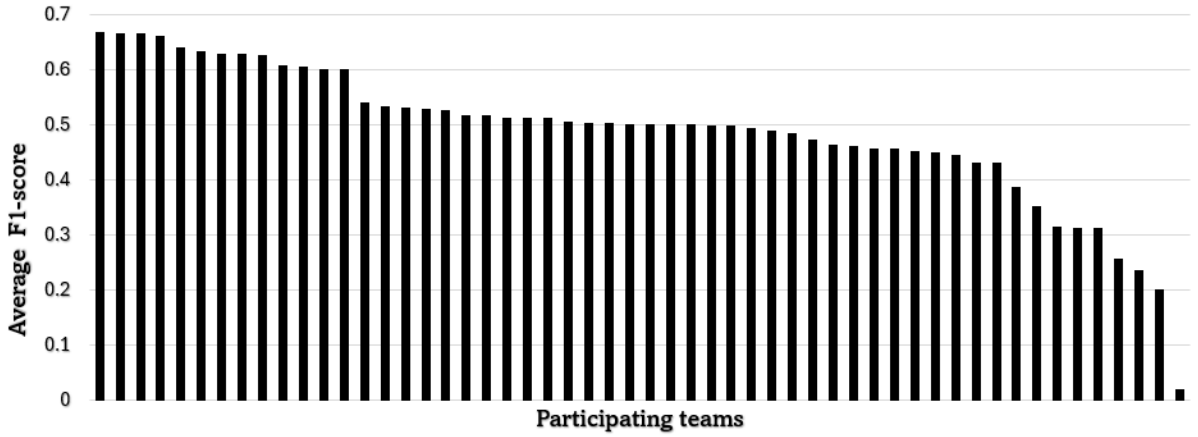


Figure 3: Distribution of the average F1-scores obtained by the participating teams to the first sub-task "Identify training data fingerprints".

- Submission ID 330: feature extraction using MobileNetV2 and hierarchical clustering
- Submission ID 331: feature extraction using MobileNetV2 and k–means for clustering
- Submission ID 332: feature extraction using VGG–16, feature reduction using PCA (number of components = 95%)
- Submission ID 333: feature extraction using VGG–16, feature reduction using PCA (number of components = 95%) and k–means for clustering
- Submission ID 334: feature extraction using ResNet50 and hierarchical clustering
- Submission ID 335: feature extraction using ResNet50 and k–means clustering

5. Discussion

For the first sub-task, "Identify training data fingerprints", a variety of methods were employed, ranging from advanced image preprocessing techniques to deep learning models. Various techniques such as binarization, histogram equalization, feature extraction, noise reduction, noise addition, colorization were used to accentuate distinct features. Different neural network architectures, including ResNet [17], MobileNet [18], Densenet [19], Effiecientnet [20] and autoencoders were used for feature extraction and classification. Additionally, strategies like majority voting, perfect agreement and agglomerative clustering were used. The distribution of the F1–scores for the two models obtained by the participating teams is depicted in Figure 3. The obtained F1-scores range from 0.019 to 0.666 with a median value of 0.5. Inoue Koki team achieved the higher six F1-scores, ranging from 0.657 to 0.667. The top-performing average F1-score of 0.667 was achieved by employing the ResNet-152 model, which used images preprocessed through histogram equalization as input.

Comparing the results for the two models, it was observed that most teams achieved slightly higher F1 scores for the first model. This indicates that they were better able to detect the training images for this generative model. We are not revealing the two models as they will be featured in future editions. However, these insights will contribute to enhancing the organization of upcoming editions. Analyzing the rest of the metrics obtained by the participant teams, we observed that certain teams achieved notably high true positive rates (recall), meaning that they managed to correctly identify the images used as being used for training. Looking forward on this runs and analyzing the accuracy values, we observed that it doesn't exceed much the random value of 0.5. This suggests that while the proposed methods succeeded in identifying used images as being used, they also misclassified a considerable portion of unused images as used for training.

For the second sub-task, "Detect generative models fingerprints", most teams used pre-trained deep learning models such as ResNet, DenseNet, EfficientNet, MobileNetV2, VGG, and Inception for feature

Table 3
Summary on the participant submissions and their results for the second sub-task. – Detect generative models' fingerprints

rank	Team	ID #	Run ID	ARI
#1	SDVAHCS/UCSD	545	2	1
#2	AI Multimedia Lab	330	5	0.997085
#3	AI Multimedia Lab	327	1	0.996517
#4	AI Multimedia Lab	326	2	0.934709
#5	AI Multimedia Lab	331	6	0.900844
#6	Csmorgan	447	2	0.9000159
#7	SDVAHCS/UCSD	550	6	0.885478
#8	SDVAHCS/UCSD	590	8	0.877797
#9	SDVAHCS/UCSD	548	4	0.851990
#10	SDVAHCS/UCSD	549	6	0.851362
#11	Csmorgan	446	1	0.813749
#12	AI Multimedia Lab	334	9	0.722857
#13	AI Multimedia Lab	333	8	0.654021
#14	AI Multimedia Lab	335	10	0.645386
#15	Biomedical Imaging Goa	307	1	0.638117
#16	SDVAHCS/UCSD	547	1	0.577203
#17	SDVAHCS/UCSD	225	4	0.577203
#18	AI Multimedia Lab	332	7	0.552682
#19	AI Multimedia Lab	329	4	0.5037
#20	Biomedical Imaging Goa	321	2	0.434414
#21	Csmorgan	452	4	0.365604
#22	AI Multimedia Lab	328	3	0.329388
#23	Biomedical Imaging Goa	324	4	0.272975
#24	Csmorgan	451	3	0.267530
#25	Csmorgan	458	8	0.232390
#26	KDE-med-lab	237	2	0.226339
#27	Csmorgan	456	7	0.178545
#28	KDE-med-lab	248	3	0.166582
#29	KDE-med-lab	257	5	0.123426
#30	KDE-med-lab	271	9	0.091818
#31	KDE-med-lab	258	6	0.060058
#32	KDE-med-lab	254	4	0.045286
#33	KDE-med-lab	270	8	0.038242
#34	KDE-med-lab	259	7	0.014388
#35	KDE-med-lab	480	10	0.013856
#36	SDVAHCS/UCSD	546	3	0.003375
#37	Csmorgan	454	6	0.001776
#38	Csmorgan	453	5	0.001313
#39	KDE-med-lab	236	1	0.000816
#40	GAN-Amis	516	5	0.000079
#41	Biomedical Imaging Goa	323	3	0.000046
#42	GAN-Amis	518	7	-0.000010
#43	GAN-Amis	520	8	-0.000546
#44	GAN-Amis	277	1	-0.000615
#45	GAN-Amis	513	4	-0.000993
#46	GAN-Amis	517	6	-0.002019

extraction. These models were chosen for their proven efficacy in capturing complex patterns and hierarchical features in images. A variety of clustering algorithms were employed across the methods. K-means was the most commonly used clustering algorithm, but other techniques like hierarchical clustering, Gaussian Mixture Models (GMM), and t-SNE were also applied to group the extracted

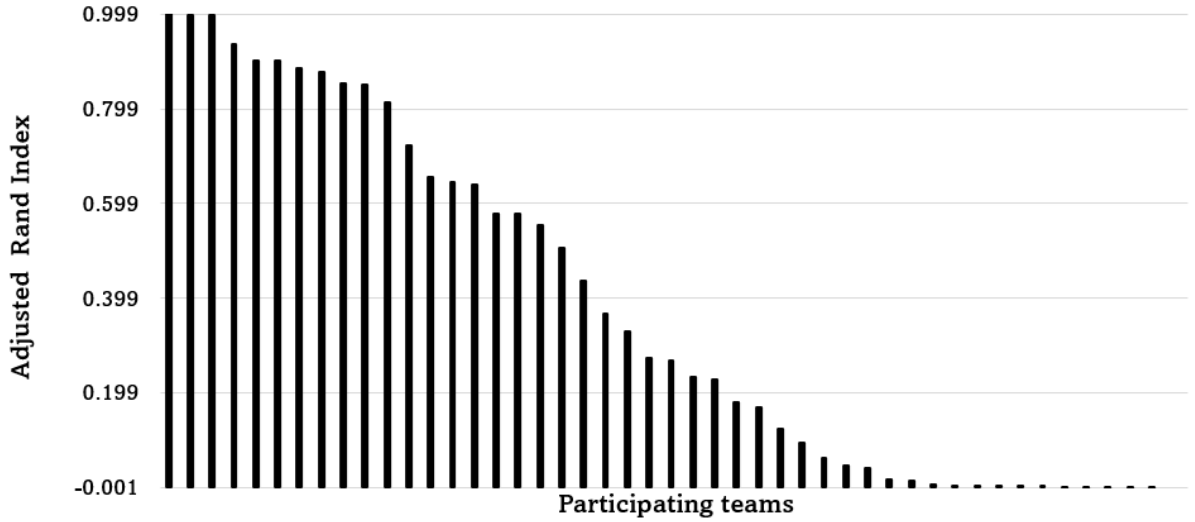


Figure 4: Distribution of the ARI scores obtained by the participating teams to the second sub-task "Detect generative models' fingerprints".

features based on their similarities. Several teams used dimensionality reduction techniques like Principal Component Analysis (PCA) to manage the high dimensionality of the extracted features, ensuring the retention of essential information while reducing computational complexity. Many approaches involved combining multiple models or techniques to enhance robustness. For example, some teams used ensemble methods or combined different neural network architectures to improve feature extraction and clustering accuracy. These commonalities reflect a comprehensive approach to identifying generative model fingerprints by leveraging a combination of advanced deep learning techniques, traditional pattern recognition methods, and thorough experimental validation.

The distribution of the ARI scores obtained by the participating teams is depicted in Figure 4. The obtained ARI scores range from -0.00201 to 1. The SDVAHCS/UCSD team achieved the highest possible ARI score of 1, indicating perfect clustering of the synthetic images. This success was attained by applying t-SNE clustering on Painters embeddings, a model trained to predict painters from artwork images. The next four highest ARI scores, ranging from 0.9008 to 0.9970, were achieved by the task organizing team, AI Multimedia Lab. Additionally, Csmorgan also managed to achieve an ARI above 0.9, demonstrating significant accuracy in their clustering results. Negative ARI values indicate that the clustering performed worse than random, suggesting that GAN-Amis grouped the data points in a way that significantly deviates from the ground truth. This result implies that the models proposed by the team may not fully understand the underlying structure of the data, and both the models and the feature extraction methods need refinement. These results offer valuable insights into the complexities and challenges of the datasets. The low ARI value indicates that the clustering performance is close to what would be expected by random chance. This means that the proposed algorithms failed to find meaningful and correct groupings of the features. Essentially, the clustering results are equivalent to randomly assigning data points to clusters, demonstrating that the models did not successfully capture the underlying features of the provided data.

6. Conclusions

The second edition of the ImageCLEFmedical GANs task introduced two sub-tasks for participants: a prediction-based task utilizing both real and generated images, and a clustering task using only generated images. Ten teams participated in the first sub-task, and seven teams participated in the second sub-task. A range of method were proposed for the first sub-task, "Identify training data fingerprints", including advanced image preprocessing techniques and deep learning models. Techniques such as binarization,

histogram equalization, and feature extraction were employed to enhance features, with strategies like majority voting and agglomerative clustering improving results. F1-scores ranged from 0.019 to 0.666, with the Inoue Koki team achieving the highest score of 0.667 using the ResNet-152 model on histogram-equalized images. For the second sub-task, "Detect generative models' fingerprints", the majority of methods included using a pre-trained CNN for feature extraction. Clustering algorithms such as k–means, hierarchical clustering, GMM and t-SNE were applied to group the extracted features, The SDVAHCS/UCSD team achieved a perfect ARI score of 1 using t-SNE clustering on Painters embeddings. High ARI scores from other teams, such as csmorgan, AI Multimedia Lab, further illustrated the effectiveness of combining multiple models and techniques. However, negative ARI values highlighted challenges, indicating that some models failed to understand the data's underlying structure, pointing to areas needing refinement.

The results highlight the complexities and challenges in both sub-tasks, offering valuable directions for enhancing future editions of the task. Future editions of this task will broaden the scope of synthetic medical data studies by varying aspects such as datasets and generation methods. Additionally, based on the insights we gained during the first two editions, we plan to introduce new tasks focused on different aspects of the privacy and security of the generated data. We will also investigate whether any alternative metrics for evaluation could be more suitable for the already proposed tasks.

Acknowledgments

The contribution of Alexandra Andrei, Bogdan Ionescu and Henning Müller to this task is supported under project AI4Media, A European Excellence Centre for Media, Society and Democracy, H2020 ICT-48-2020, grant #951911.

References

[1] B. Ionescu, H. Müller, A. Drăgulinescu, J. Rückert, A. Ben Abacha, A. Garcıa Seco de Herrera, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, C. S. Schmidt, T. M. Pakull, H. Damm, B. Bracke, C. M. Friedrich, A. Andrei, Y. Prokopchuk, D. Karpenka, A. Radzhabov, V. Kovalev, C. Macaire, D. Schwab, B. Lecouteux, E. Esperança-Rodier, W. Yim, Y. Fu, Z. Sun, M. Yetisgen, F. Xia, S. A. Hicks, M. A. Riegler, V. Thambawita, A. Storås, P. Halvorsen, M. Heinrich, J. Kiesel, M. Potthast, B. Stein, Overview of ImageCLEF 2024: Multimedia retrieval in medical applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 15th International Conference of the CLEF Association (CLEF 2024), Springer Lecture Notes in Computer Science LNCS, Grenoble, France, 2024.

[2] B. Ionescu, H. Müller, A. M. Drăgulinescu, A. Idrissi-Yaghir, A. Radzhabov, A. G. S. d. Herrera, A. Andrei, A. Stan, A. M. Storås, A. B. Abacha, et al., Advancing multimedia retrieval in medical, social media and content recommendation applications with imageclef 2024, in: European Conference on Information Retrieval, Springer, 2024, pp. 44–52.

[3] A.-G. 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: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), volume 3497, 2023.

[4] W. M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical association 66 (1971) 846–850.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[6] K. Inoue, T. Asakawa, K. Shimizu, K. Nomura, M. Aono, Prediction of whether an image is a real

or generated image with image processing and integration of predictions, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[7] A. Gentili, Detecting training data and generative model fingerprints in synthetic ct scans using machine learning, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[8] H. Tang, H. Wang, J. Chen, Classification of real and generated images based on feature similarity, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[9] S. Fukuyama, T. Asakawa, K. Shimizu, K. Nomura, M. Aono, Kde lab at imageclefmedical gans 2024, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[10] T. Asakawa, K. Shimizu, K. Nomura, M. Aono, Kde-med-lab at imageclef 2024: Identify data and detect generative models using cnn by lung segmentation based on u-net, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[11] D. Miranda, A. Rane, B. Naik, Analyzing generated images using machine learning, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[12] M. I. Emon, M. Hoque, M. R. Hasan, F. Khalifa, M. Rahman, Fingerprint identification of generative models using a multiformer ensemble approach, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[13] S. Cao, X. Zhou, Evaluation of the privacy of images generated by imageclefmedical gans 2024 based on similarity methods, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[14] A. Andrei, M. G. Constantin, M. Dogariu, B. Ionescu, Ai multimedia lab at imageclefmedical gans 2024: Deep learning approaches for analyzing synthetic medical images, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[15] A. Upganlawa, A. Lad, A. Desai, Evaluating clustering of gan-generated medical images using custom and pre-trained cnn architectures to identify gan fingerprints, in: CLEF2024 Working Notes,CEUR Workshop Proceedings, Grenoble, France, 2024.
[16] A. Andrei, B. Ionescu, Aimultimedialab at imageclefmedical gans 2023: determining "fingerprints" of training data in generated synthetic images, in: CLEF2023 Working Notes, CEUR Workshop Proceedings, Thessaloniki, Greece, 2023.
[17] B. Koonce, B. Koonce, Resnet 50, Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization (2021) 63–72.
[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
[19] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, K. Keutzer, Densenet: Implementing efficient convnet descriptor pyramids, arXiv preprint arXiv:1404.1869 (2014).
[20] B. Koonce, B. Koonce, Efficientnet, Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization (2021) 109–123.