Detecting Training Data Usage in Synthetic Images Using Machine Learning Techniques

Notebook for the Challengers Lab at CLEF 2025

Krithikha Sanju Saravanan[1,*,†], Mohanavalli Subramaniam[1,†], R B Anirudh Narayanan[1,†], Bharath P[1,†] and Karunanidhi Ayyamperumal[1,†]

[1]*Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Tamil Nadu, India*

Abstract

This paper presents an investigation into detecting hidden "fingerprints" of real training images within synthetic medical images generated by Generative Adversarial Networks (GANs). The potential for GAN-generated images to retain traces of the source data poses a significant privacy risk, especially in sensitive domains like medical imaging where data sharing is restricted by privacy regulations. While GANs offer a powerful solution to data scarcity for training diagnostic models, it is crucial to ensure their outputs do not compromise patient confidentiality. This study evaluates the effectiveness of various machine learning strategies in identifying whether a specific real image was used in a GAN's training process. We explore a range of methods, from supervised deep learning classifiers to a comprehensive unsupervised framework that uses dimensionality reduction via PCA and clustering techniques. By measuring the feature-space proximity between real and synthetic images, we aim to uncover signatures of training data usage. The primary objective is to determine the privacy implications of employing GAN-generated data in medical applications, thereby providing a clearer understanding of the risks and benefits.

Keywords

Synthetic Data, GAN, Medical Imaging, Privacy, Fingerprint Detection, PCA, Clustering, Training Data Leakage, Supervised Learning, Unsupervised Learning

1. Introduction

The proliferation of deep learning has significantly advanced the capabilities of Computer-Aided Diagnosis (CAD) systems in medical imaging. However, the performance of these data-intensive models is often hampered by the scarcity of large, diverse, and publicly available datasets, a direct consequence of stringent patient privacy regulations and the high cost of data annotation. Generative Adversarial Networks (GANs) have emerged as a promising technology to mitigate this issue by producing high-fidelity synthetic images, which can be used to augment datasets for training more robust and accurate models.

Despite their potential, GANs introduce a critical and unresolved privacy concern: the risk that synthetic images may implicitly retain signatures or "fingerprints" of the real patient data used to train them. The existence of such fingerprints would mean that synthetic images carry similar privacy liabilities as the original data, thereby undermining their utility as a shareable, anonymized resource. This paper addresses the central question: can we reliably detect if a specific real image was used in the training of a GAN?

This working note details the participation of the "Challengers" lab in the ImageCLEFmedical 2025 GANs challenge, specifically addressing Subtask 1: "Detect Training Data Usage" [1]. We present a systematic evaluation of various machine learning strategies, comparing a fully supervised deep learning approach against a comprehensive unsupervised framework. Our goal is to rigorously assess the feasibility of fingerprint detection and contribute to a clearer understanding of the privacy guarantees

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

†These authors contributed equally.

krithikhasanjus@ssn.edu.in (K. S. Saravanan); mohanas@ssn.edu.in (M. Subramaniam); anirudh2210940@ssn.edu.in (R. B. A. Narayanan); bharath2210788@ssn.edu.in (B. P); karunanidhi2210188@ssn.edu.in (K. Ayyamperumal)

of GAN-generated data in the broader context of the ImageCLEF 2025 initiative [2]. To enhance reproducibility, our code is available online.[1]

2. Related Works

The application of Generative Adversarial Networks (GANs), first introduced by Goodfellow et al. [3], has expanded rapidly, particularly within the domain of medical imaging. Early research demonstrated their profound utility for data augmentation, a critical task where acquiring large, labeled datasets is often prohibitive. For instance, Frid-Adar et al. [4] successfully utilized Deep Convolutional GANs (DCGANs) to synthesize CT images of liver lesions, thereby augmenting their training set to improve CNN classification performance. Similarly, other works have explored GANs for augmenting datasets of lung nodules and chest X-rays, establishing them as a powerful tool for enhancing the robustness and accuracy of diagnostic models when faced with limited data [5, 6].

However, the power of GANs to learn and replicate complex data distributions also introduces significant privacy concerns. The very process that enables high-fidelity image generation raises questions about whether the models are inadvertently memorizing and exposing sensitive information from the training data. This has led to a growing body of research on the privacy risks of generative models, including the threat of membership inference attacks, where an adversary aims to determine if a specific data point was part of a model's training set [7, 8]. The concept of "training data fingerprinting"—the central theme of our work—is a nuanced form of this risk, suggesting that unique, traceable signatures of training samples may be embedded within the synthetic outputs.

The ImageCLEFmedical challenges have been instrumental in formalizing this problem and directing research efforts toward solutions [2, 1]. For instance, throughout the 2023 edition, the DMK SSN team tackled this problem using an integrated approach of Convolutional Neural Networks (CNNs), SIFT with KNN classification, and Perceptual Hashing [9]. They focused on calculating the Hamming distances for actual images and their synthesized counterparts to determine if the GAN outputs were significantly similar to the training data. Their study was an initial confirmation of the fingerprinting hypothesis and attracted attention for efforts toward more robust detection frameworks. To build on such efforts and address the current challenge, our work employs a diverse suite of well-established machine learning techniques.

For deep feature extraction, we utilize several landmark convolutional neural network architectures: ResNet, known for its deep residual learning framework [10]; VGGNet, which established the importance of depth in network design [11]; DenseNet, which introduced dense connectivity patterns [12]; and the highly efficient MobileNetV2 [13] and EfficientNet [14] architectures. For our unsupervised pipeline, we pair these deep learners with classical methods, including Principal Component Analysis (PCA) for dimensionality reduction [15], Local Binary Patterns (LBP) for texture analysis [16], and clustering algorithms like K-Means [17] and DBSCAN [18]. While prior work has identified the fingerprinting problem, our primary contribution is the systematic and rigorous comparative analysis of these varied techniques. By evaluating both supervised and extensively tuned unsupervised pipelines, we provide a clear benchmark on the difficulty of the ImageCLEF 2025 fingerprint detection task and offer practical insights into the privacy guarantees of synthetic medical data.

3. Dataset

Our study utilizes the official benchmarking dataset provided for the ImageCLEFmedical 2025 GANs Task, specifically Subtask 1: "Detect Training Data Usage" [1]. The dataset is composed of grayscale axial CT slices of lungs from tuberculosis patients, saved in PNG format with a uniform resolution of 256×256 pixels. It covers a wide spectrum of pathological conditions, from normal lung tissue to severe lesion cases.

[1] https://github.com/karuna207/ImageCLEF-Subtask1

The dataset is divided into training and test sets, each structured to facilitate the fingerprint detection task.

- Training Set: This set is provided to develop and tune the detection models. It contains three distinct subsets:
  - `real_used`: A folder containing 100 real CT images that were used to train the provided GAN model.
  - `real_not_used`: A folder containing 100 real CT images that were explicitly *not* used in the GAN's training process.
  - `generated`: A collection of 5,000 synthetic images produced by the GAN, which was trained exclusively on the images from the `real_used` set.
- Test Set: This set is used for the final evaluation of the models. It consists of two folders:
  - `real_unknown`: A folder containing 500 real CT images. Some of these images were part of the GAN's training set, while others were not.
  - `generated`: An additional set of 2,000 synthetic images produced by the same GAN architecture under identical training conditions.

The primary objective of the subtask is to classify each image in the `real_unknown` folder with a binary label: '1' for "used" (i.e., the image was part of the GAN's training data) or '0' for "not used".

4. Methodology

To address the challenge of detecting GAN training data "fingerprints" in the ImageCLEF 2025 Medical Task 1, we developed and evaluated two distinct methodological pillars: a comprehensive, semi-unsupervised pipeline based on feature-space distances, and a fully-supervised deep learning classifier.

4.1. Unsupervised Distance-Based Pipeline

The fundamental hypothesis of our unsupervised approach is that real images used to train a GAN will lie closer in a high-dimensional feature space to the manifold of generated images than real images not used in training. This pipeline was designed to systematically find the optimal combination of components to exploit this relationship.

4.1.1. Pipeline Stages

Our approach follows a structured, multi-stage process to ensure that the final models are robust and optimized. The architecture of the pipeline is illustrated in Figure 1.

1. Hyperparameter Tuning: We first perform an extensive search over a large space of models and parameters on a validation set created by splitting the initial labeled data (`real_used` and `real_not_used`). This identifies the most promising model configurations.
2. Model Evaluation and Selection: The best-performing configurations from the tuning phase are then re-instantiated, and models that achieve a performance metric above a predefined threshold (Cohen's Kappa > 0.1) on the full labeled set are selected.
3. Final Inference: The selected, validated models are used to predict labels for the final, unseen test dataset.

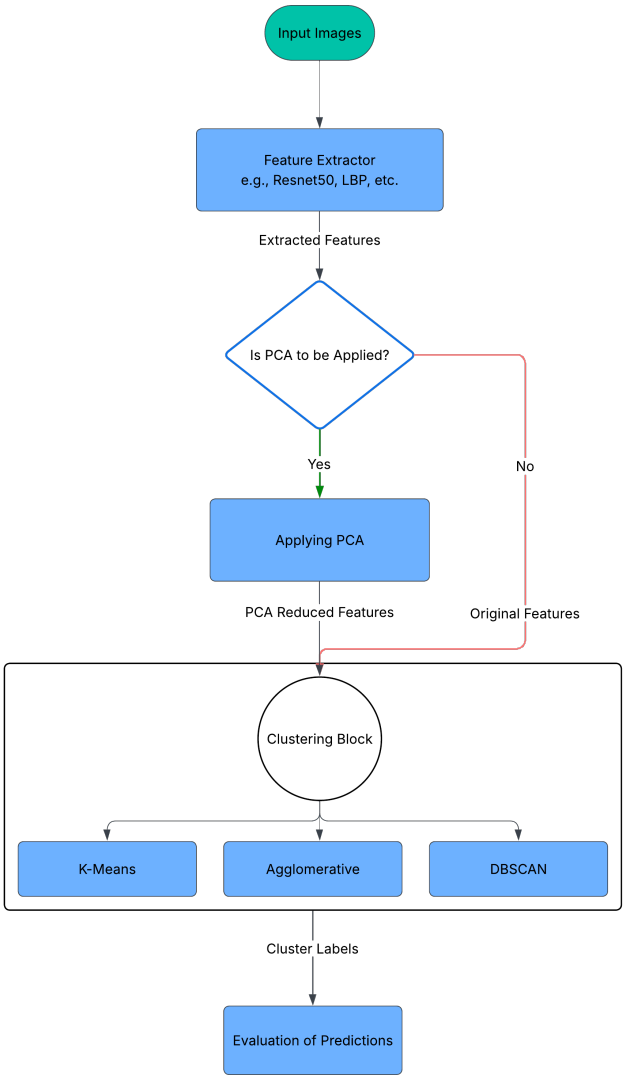

Figure 1: Architectural overview of the unsupervised feature extraction and clustering pipeline.

4.1.2. Component-wise Breakdown

The pipeline is modular, allowing for the systematic testing of different components.

- Feature Extraction: We evaluated a diverse set of pre-trained CNN backbones: ResNet50 [10], DenseNet121 [12], VGG16 [11], MobileNetV2 [13], and EfficientNet-B0 [14]. Additionally, we tested a traditional approach using Local Binary Patterns (LBP) [16].
- Dimensionality Reduction: We integrated an optional step using `StandardScaler` and PCA [15], testing configurations that retained 95% and 90% of the variance.
- Clustering: We applied K-Means [17], Agglomerative Clustering, and DBSCAN [18] to the feature vectors of the `generated` images to find representative centroids.
- Classification Logic: Real images were classified based on the Euclidean distance of their features to the generated centroids, using either the minimum (`'min'`) or average (`'avg'`) distance compared against an optimized threshold.

4.1.3. Systematic Hyperparameter Tuning

The key to this pipeline is the rigorous tuning process. We created a 30% validation split and performed a grid search over all component combinations. The final classification threshold was optimized by

testing 95 percentile values on the validation set's distance distribution. The performance of every unique parameter combination was evaluated using the Cohen's Kappa score, and the best configuration for each feature extractor was selected for the final evaluation.

4.1.4. Step-by-Step Unsupervised Workflow

Our unsupervised pipeline is executed through a systematic, multi-step process for each feature extractor configuration. The entire workflow is automated in a single script, ensuring reproducibility.

1. Data Preparation and Validation Split: The process begins by identifying the paths to all images in the `generated` and real image folders. The set of all available labeled real images (`real_used` and `real_not_used`) is then split into a training/test set and a validation set using a 70/30 ratio, controlled by a `VALIDATION_SPLIT_SIZE` parameter of 0.3. This split is stratified and shuffled with a fixed `RANDOM_SEED` of 42 to ensure consistency across experiments. The smaller validation set (60 images in our case) is used exclusively for the hyperparameter tuning loop.
2. Feature Extraction: For each feature extractor and PCA configuration defined in our experimental setup, the pipeline processes every image. A `UniversalFeatureExtractor` class loads the specified pre-trained model (e.g., `resnet50`), and a dedicated function computes high-dimensional feature vectors for all images in both the `generated` set and the validation set.
3. Dimensionality Reduction: If a PCA configuration is specified (e.g., retain 90% or 95% of variance), a `StandardScaler` and a `PCA` model are employed. Critically, the scaler and PCA models are *fitted only on the feature vectors of the generated images*. This prevents any data leakage from the validation set into the feature transformation process. The fitted models are then used to transform the features of both the generated and validation sets.
4. Synthetic Data Clustering: With the features prepared, a clustering algorithm (`KMeans`, `AgglomerativeClustering`, or `DBSCAN`) is applied. This clustering is performed *exclusively on the feature vectors of the generated images*. The objective is not to classify, but to model the distribution of the synthetic data by finding its cluster centroids. These centroids serve as representatives of the GAN's output manifold.
5. Distance Calculation and Threshold Tuning: The core classification logic is tuned in this step. For each image in the validation set, the pipeline calculates its Euclidean distance to the synthetic data centroids computed in the previous step. Both the minimum and average distances are considered as potential metrics. This produces a set of distances for the validation images. To find an optimal decision boundary, the pipeline iterates through 95 different percentile values of this distance distribution (from 1 to 95). Each percentile is treated as a potential classification threshold.
6. Optimal Configuration Selection: For every unique combination of parameters (feature extractor, PCA, clustering algorithm, distance metric, etc.), a set of predictions is made for the validation set. The quality of these predictions is measured by calculating the Cohen's Kappa score against the ground truth labels. The full set of parameters that results in the highest Kappa score is stored as the single best configuration for that specific experimental branch.

4.2. Supervised Classification Baseline

In parallel to our unsupervised approach, we implemented a fully supervised binary classification model to serve as a high-performance baseline. This model directly learns to differentiate between images that were used in GAN training and those that were not. The architecture of the pipeline is illustrated in Figure 2.

- Architecture: The model uses a pre-trained ResNet50V2 [10] backbone, leveraging transfer learning from ImageNet. We replaced the original classifier with a custom classification head composed of a Global Average Pooling (GAP) layer, a Dense layer with ReLU activation, and a

final Dense layer with a Sigmoid activation function to output a probability score for the "used" class.

- Training and Data Handling: The model was trained directly on the labeled `real_used` and `real_not_used` image sets. Images were resized to 256×256 pixels and normalized. We employed the Adam optimizer [19] and the Binary Crossentropy loss function, which are standard choices for binary classification tasks. Training was accelerated using a GPU.
- Inference: Once trained, the model was applied directly to the `real_unknown` test set to predict a binary label for each image. This approach tests the upper bound of performance when explicit labels are available for training.

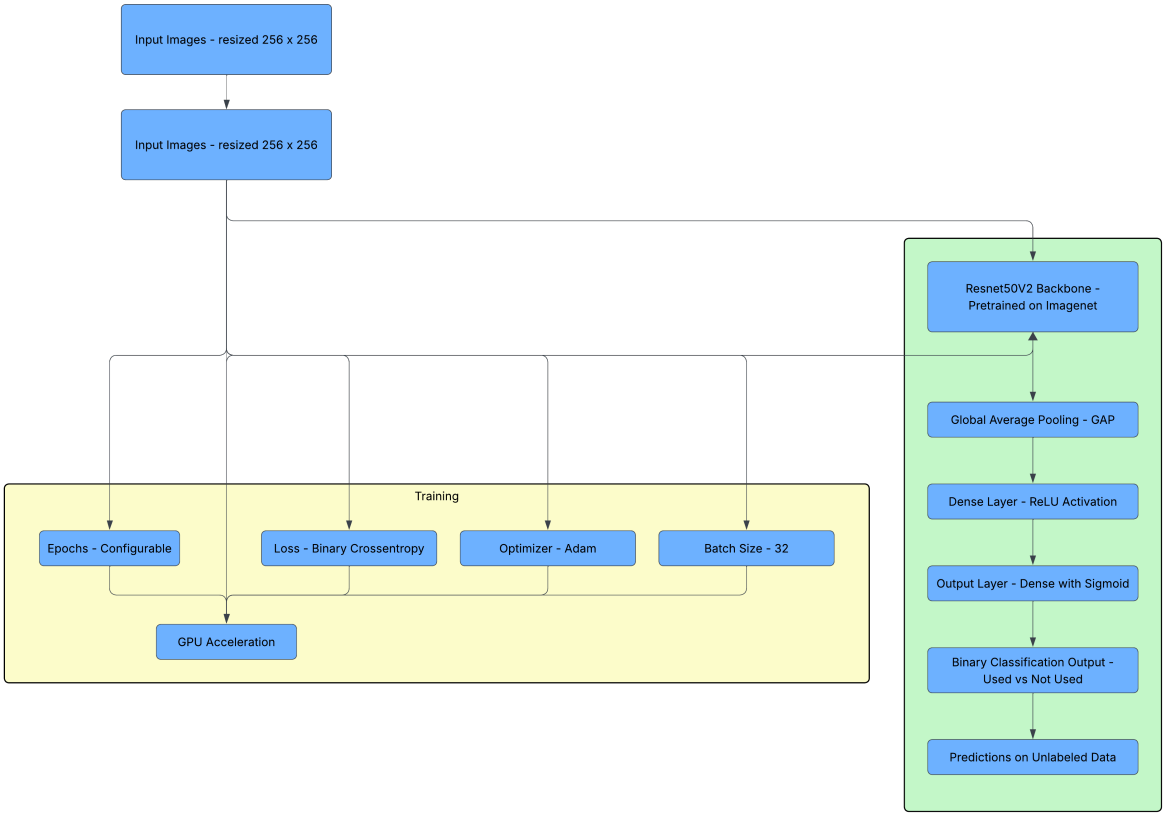

Figure 2: Architectural Overview of the supervised binary classification pipeline.

4.2.1. Step-by-Step Supervised Workflow

Our supervised baseline model was implemented using the TensorFlow and Keras frameworks. The workflow is detailed below.

1. Data Preparation and Labeling: The file paths for all images in the `real_used` and `real_not_used` directories were loaded. Crucially, each image path was explicitly assigned a corresponding integer label: '1' for images in `real_used` and '0' for images in `real_not_used`. This labeled dataset was then split into training and validation sets for model training and evaluation.
2. Model Architecture and Transfer Learning: The architecture is built upon a `ResNet50V2` model pre-trained on ImageNet, leveraging the power of transfer learning. The convolutional base of the pre-trained model was frozen (`base_model.trainable = False`) to act as a fixed feature extractor. A custom classification head was then stacked on top, consisting of:

- A `Conv2D` layer to adapt the single-channel grayscale input to the three-channel format expected by ResNet50V2.
- The frozen `ResNet50V2` base.
- A `GlobalAveragePooling2D` layer to reduce the feature maps to a vector.
- A `Dense` layer with 128 units and a ReLU activation function.
- A `Dropout` layer with a rate of 0.5 for regularization.
- A final `Dense` output layer with a single unit and a `sigmoid` activation for binary probability prediction.

3. Data Augmentation and Training: To improve model generalization, on-the-fly data augmentation was applied to the training set using Keras's `ImageDataGenerator`. Augmentations included random rotations, width and height shifts, horizontal flips, and zooming. The model was compiled with the Adam optimizer and `binary_crossentropy` as the loss function. Training was performed for 20 epochs with a batch size of 32.
4. Evaluation and Final Inference: After training, the model's performance was assessed on the held-out validation set, achieving a final validation Kappa score of 0.05. The fully trained model was then used to make predictions on the final, unlabeled test dataset provided by the task organizers, with the results saved to a submission file.

5. Results and Analysis

In this section, we present the empirical results of our work. We first detail the performance of our models on our internal validation set, which guided our selection process. We then report the final results for our four submitted runs on the official ImageCLEF 2025 test dataset and provide an analysis of the key findings.

5.1. Internal Evaluation and Model Selection

Our methodology involved two primary streams of experimentation: a systematic, unsupervised pipeline and a supervised baseline. The top-performing configurations from both streams were selected for final submission based on their performance on our internal labeled validation data.

5.1.1. Unsupervised Pipeline Validation

The extensive hyperparameter search across our 16 unsupervised configurations yielded a wide range of results. The performance of the best configuration for each feature extractor and PCA setting on the full 200 labeled images is shown in Table 1. While most models struggled, three configurations achieved a Cohen's Kappa score greater than our selection threshold of 0.1: ResNet50 (None), ResNet50 (PCA 0.9), and LBP (None). These were selected for submission as runs `1778`, `1779`, and `1777`, respectively.

5.1.2. Supervised Baseline Validation

The supervised ResNet50V2 model was trained and evaluated on a separate train/validation split of the labeled data. On its validation set of 40 images, it achieved a Kappa score of 0.05. Although modest, this positive result justified its submission as a strong baseline, submitted as run `1811`.

5.2. Final Test Set Performance

The four selected models were submitted to the official challenge platform for evaluation against the hidden test set. The official results, linking each model to its submission run ID, are detailed in Table 3.

Table 1
Performance of the Best Unsupervised Configurations on the Full Labeled Evaluation Set (200 Images). The table shows the final metrics for the optimal parameter set found during the hyperparameter tuning for each feature extractor (FE) and PCA combination. Configurations marked with an asterisk (*) were selected for final submission based on their Kappa score > 0.1.

Feature Extractor (FE)	PCA Setting	Accuracy	Precision	Recall	F1-Score	Cohen's Kappa
ResNet50*	None	0.565	0.610	0.370	0.460	0.130
ResNet50	95%	0.550	0.679	0.190	0.297	0.100
ResNet50*	90%	0.555	0.762	0.160	0.264	0.110
DenseNet121	None	0.520	0.571	0.160	0.250	0.040
DenseNet121	95%	0.520	0.521	0.430	0.472	0.040
DenseNet121	90%	0.535	0.542	0.450	0.492	0.070
VGG16	None	0.525	0.521	0.730	0.606	0.050
VGG16	95%	0.495	0.497	0.890	0.638	-0.010
VGG16	90%	0.495	0.497	0.890	0.638	-0.010
MobileNetV2	None	0.515	0.516	0.490	0.503	0.030
MobileNetV2	95%	0.520	0.750	0.060	0.111	0.040
MobileNetV2	90%	0.515	0.667	0.060	0.110	0.030
EfficientNet-B0	None	0.520	0.522	0.590	0.551	0.040
EfficientNet-B0	95%	0.485	0.492	0.940	0.646	-0.030
EfficientNet-B0	90%	0.485	0.492	0.940	0.646	-0.030
LBP*	None	0.565	0.587	0.440	0.503	0.130

Table 2
Overall Performance of the Supervised ResNet50V2 Model on the Validation Set. The metrics are derived from the detailed classification report.

Feature Extractor (FE)	PCA Setting	Accuracy	Precision	Recall	F1-Score	Cohen's Kappa
ResNet50V2 (Supervised)	None	0.525	0.53	0.53	0.52	0.050

Table 3
Official Test Set Results for Submitted Runs.

Model Description	Run ID	Submission File	Acc.	Prec.	Recall	Cohen's Kappa (κ)
ResNet50 (Unsupervised)	1778	`resnet1_sorted.zip`	0.442	0.446	0.480	-0.116
ResNet50-PCA(0.9) (Unsupervised)	1779	`resnet2_sorted.zip`	0.484	0.435	0.108	-0.032
ResNet50V2 (Supervised)	1811	`resnet3.zip`	0.506	0.506	0.492	0.012
LBP (Unsupervised)	1777	`lbp_sorted.zip`	0.412	0.376	0.268	-0.176

5.3. Analysis

The supervised approach was the only method to achieve a positive Cohen's Kappa score (0.012). While this score is only marginally better than random chance, it clearly indicates that directly training a classifier on labeled examples was the most effective strategy for capturing the subtle differences between the 'used' and 'not used' classes.

Second, the unsupervised models failed to generalize from the validation set to the official test set. Models 1, 2, and 4, which had shown some promise with positive Kappa scores during internal validation,

all yielded negative Kappa scores on the final test data. This suggests that the feature-space proximity hypothesis, while appealing, is not a robust signal on its own and may be highly sensitive to the specific data distribution, leading to poor generalization.

Finally, the low scores across all submissions underscore the extreme difficulty of the GAN fingerprinting task. The signatures left by the GAN training process are incredibly subtle and not easily detected by either standard deep feature comparison or direct supervised classification, and therefore this shall remain as a challenging and open area for future research.

6. Conclusion

In this paper, we presented the work of our team, the Challengers, in the ImageCLEFmedical 2025 GANs Task 1, addressing the critical challenge of detecting training data "fingerprints" in synthetic medical images. To this end, we developed and evaluated two distinct methodological pillars: a fully supervised binary classifier using a ResNet50V2 architecture and an extensive, semi-unsupervised pipeline designed to find an optimal configuration for distance-based detection.

Our final results on the official test set reveal the profound difficulty of this task. The supervised baseline was the only approach to achieve a performance marginally better than random chance, yielding a positive Cohen's Kappa score of 0.012. In contrast, our systematically-tuned unsupervised models, which were based on feature-space proximity, failed to generalize from our internal validation set to the final test set, all yielding negative Kappa scores. This finding strongly suggests that the subtle traces left by the training process are not easily captured by standard feature distance metrics alone.

The primary limitation of our study is the low absolute performance across all attempted models, which confirms that the signals of data usage are extremely faint. Plus, our extensive unsupervised pipeline relied on a labeled validation set for its hyperparameter tuning, a resource that may not be available in real-world scenarios, limiting its practical applicability as a truly unsupervised method. Therefore, despite our task remaining as a challenge for future research to tackle, we hope our work provides a comprehensive benchmark for this task, highlighting the limitations of current standard approaches and charting clear paths for future research into ensuring the privacy of generative models in medicine.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini 2.5 and Grammarly in order to correct grammatical errors and review the intelligibility of the paper. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A.-G. Andrei, M. G. Constantin, M. Dogariu, A. Radzhabov, L.-D. Ştefan, Y. Prokopchuk, V. Kovalev, H. Müller, B. Ionescu, Overview of ImageCLEFmedical 2025 GANs task: Training data analysis and fingerprint detection, in: CLEF2025 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Madrid, Spain, 2025.

[2] B. Ionescu, H. Müller, D.-C. Stanciu, A.-G. Andrei, A. Radzhabov, Y. Prokopchuk, Ştefan, Liviu-Daniel, M.-G. Constantin, M. Dogariu, V. Kovalev, H. Damm, J. Rückert, A. Ben Abacha, A. García Seco de Herrera, C. M. Friedrich, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, C. S. Schmidt, T. M. G. Pakull, B. Bracke, O. Pelka, B. Eryilmaz, H. Becker, W.-W. Yim, N. Codella, R. A. Novoa, J. Malvehy, D. Dimitrov, R. J. Das, Z. Xie, H. M. Shan, P. Nakov, I. Koychev, S. A. Hicks, S. Gautam, M. A. Riegler, V. Thambawita, P. Halvorsen, D. Fabre, C. Macaire, B. Lecouteux, D. Schwab, M. Potthast, M. Heinrich, J. Kiesel, M. Wolter, B. Stein, Overview of imageclef 2025: Multimedia retrieval in medical, social media and content recommendation applications, in: Experimental

IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 16th International Conference of the CLEF Association (CLEF 2025), Springer Lecture Notes in Computer Science LNCS, Madrid, Spain, 2025.

[3] 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 27 (NIPS 2014), 2014.

[4] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification, Neurocomputing 321 (2018) 321–331.

[5] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, M. Michalski, Medical image synthesis for data augmentation and anonymization using generative adversarial networks, arXiv preprint arXiv:1807.10225 (2018).

[6] A. Salehi, A. Sari, D. Erdogmus, Generative adversarial network with handcrafted features for data augmentation of medical images, in: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), IEEE, 2017, pp. 706–710.

[7] J. Hayes, L. Melis, G. Danezis, E. De Cristofaro, Logan: Membership inference attacks against generative models, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 579–595.

[8] B. Hilprecht, M. Härterich, D. Bernau, Monte carlo and anom-based membership inference attacks against generative models, in: 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), IEEE, 2019, pp. 174–183.

[9] D. Subburam, S. M. SathyaNarayanan, B. Anand, K. Srinivasan, M. Subramaniam, Dmk-ssn at imageclef 2023 medical: Controlling the quality of synthetic medical images created via gans using machine learning and image hashing techniques, in: CLEF 2023 Working Notes, CEUR-WS.org, 2023.

[10] 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 (CVPR), 2016, pp. 770–778.

[11] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[12] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708.

[13] 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 (CVPR), 2018, pp. 4510–4520.

[14] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning (ICML), PMLR, 2019, pp. 6105–6114.

[15] I. Jolliffe, Principal component analysis, Springer series in statistics, 2002.

[16] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 971–987.

[17] Some methods for classification and analysis of multivariate observations, volume 1, 1967.

[18] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise., in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996, pp. 226–231.

[19] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).