AI Multimedia Lab at ImageCLEFmedical GANs 2025: Identifying Real-Image Usage in Generated Medical Images

Notebook for the ImageCLEF Lab at CLEF 2025

Alexandra-Georgiana Andrei[1,*], Mihai Gabriel Constantin[1], Mihai Dogariu[1], Liviu-Daniel Stefan[1] and Bogdan Ionescu[1]

[1]AI Multimedia Lab, National University of Science and Technology Politehnica Bucharest, Romania

Abstract

This paper presents the participation of AI Multimedia Lab in the third edition of the 2025 ImageCLEFmedical GANs task, which investigates privacy and security concerns around generated synthetic medical images. This edition, the challenge comprises two complementary subtasks: (1) Detect which real images were used to train a GAN based on given synthetic outputs and (2) Attribute each synthetic image to its specific real-image subset of origin. We present our team's approach, which combines traditional deep learning techniques on two-step pipeline - feature extraction followed by clustering - for subtask 2, and Siamese Neural Networks for both subtasks. Evaluated on benchmark testing datasets of real and synthetic lung CT slices, our Siamese-based method achieved a Cohen's kappa of 0.036 for Subtask 1 and 99.04% accuracy for Subtask 2. Finally, we discuss the strengths and limitations of our methods and outline directions for improving the detection of training-data "fingerprints" in GAN-generated medical images. All code used in this study is available in Github [1].

Keywords

synthetic medical data, Generative Adversarial Networks, data augmentation, ImageCLEFmedical GANs, ImageCLEFbenchmarking lab,

1. Introduction

Building on the foundation laid by previous editions [1, 2], part of ImageCLEF evaluation campaign [3, 4, 5], the third edition of the 2025 ImageCLEFmedical GANs [6] task continues to explore the intersection of generative models and medical imaging, with a particular focus on data privacy and security. This edition introduces two complementary subtasks: (1) detecting which real images were used to train a generative model based on its synthetic outputs, and (2) identifying the specific real-image subset from which a given synthetic image was generated. These subtasks aim to investigate whether synthetic medical images carry identifiable traces - "fingerprints" - of the data they were trained on, raising important implications for privacy-preserving data generation.

This paper presents the participation of the AI Multimedia Lab – task organizing team – in the third edition of the ImageCLEFmedical GANs task. The paper is structured as follows: Section 2 presents the tasks and the datasets, Section 3 presents the proposed methods and the results are presented and discussed in Section 4. Finally, the paper closes with Section 5, where we present the conclusions.

[1]https://github.com/AlexandraAndrei/ImageCLEFmedical2025_GANs_AIM

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

*Corresponding author.

alexandra.andrei@upb.ro (A. Andrei); mihai.constantin84@upb.ro (M. G. Constantin); mihai.dogariu@upb.ro (M. Dogariu); liviu_daniel.stefan@upb.ro (L. Stefan); bogdan.ionescu@upb.ro (B. Ionescu)

Table 1
Summary of the training and testing datasets available for subtask 1 – "Identify Training Data Usage"

Training dataset		Testing dataset	
Real images	*Generated images*	*Real images*	*Generated images*
100 (used) 100 (not used)	5,000	500 (used and not used)	2,000

2. The 2025 ImageCLEFmedical GANs Task

2.1. Subtask 1 – "Detect Training Data Usage"

This task introduced in the first edition has been continued in both following editions in the same setup. The objective of the task is to detect "fingerprints" within the generated images to determine which of the real images were used to train the GAN that generated the synthetic images. The data set provided for this task consists of real and synthetic images generated, as described in Table 1.

2.2. Subtask 2 – "Identify Training Data Subsets"

To address Subtask 2, the goal was to predict which training subset was used to generate each image in the testing set. The training dataset was structured into two main folders: one containing real images and another containing synthetic (generated) images as described in Table 2. Each of these folders had five subfolders (t1–t5 for real, and gen_t1–gen_t5 for generated), where each pair of corresponding subfolders shared a one-to-one mapping. For example, real images in t1 were used to generate synthetic images in gen_t1. The testing dataset comprised 25,000 generated images, each of which needed to be assigned to one of the five original real-image subsets.

More information about both subtasks are available in the overview paper of the task [6].

3. Proposed Methods

3.1. Subtask 2 – "Identify Training Data Subsets"

We started with the second subtask "Identify Training Data Subsets" for which we proposed two different approaches:

- Two step pipeline: feature extraction using pre-trained models and classification using Support Vector Machine (SVM) and clustering using k-means and agglomerative hierarchical clustering;
- Siamese Neural Networks trained on (real, generated) pairs.

3.1.1. Method 1 – Feature Clustering

We adopted the pipeline presented by our team in the previous editions [7, 8] to the current task setup. In this edition, we tested different variations of the pipeline presented in Figure 1 for detecting the real subset of training data used to generate each of the synthetic images.

Feature extraction – we employed four different pretrained models that were originally trained on ImageNet dataset [9]: i) MobileNetV2 [10], ii) ResNet50 [11], iii) EfficientNet [12], and iv) DenseNet [13].

Table 2
Summary of the training and testing datasets available for subtask 1 – "Identify Training Data Subsets"

Training dataset		Testing dataset	
Real images	*Generated images*	*Real images*	*Generated images*
10,000 (2,000 for each real-image subset)	10,000 (2,000 generated from each real subset)	-	25,000

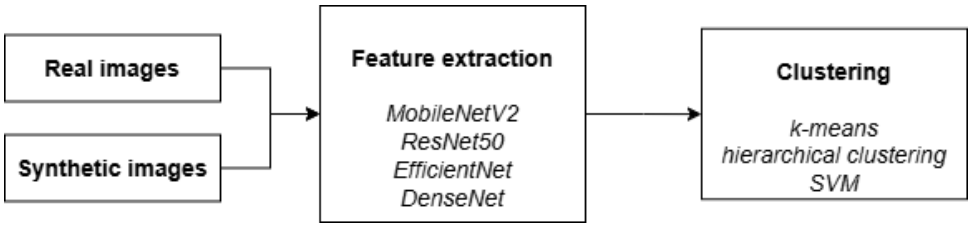


Figure 1: Overview of the proposed feature clustering methods for "Identify Training Data Subsets" subtask.

Clustering – Since the dataset includes five known training subsets, we explored three different approaches to group and classify the data: i) k-means, ii) hierarchical clustering and iii) SVM. K-means and hierarchical clustering are both unsupervised methods that group similar data points, but they work in different ways: k-means clusters data by minimizing distance to cluster centers, while hierarchical clustering builds a hierarchy based on how close points are to each other. SVM, on the other hand, is a supervised method that learns to separate data based on labeled examples. By applying all three methods, we aim to understand whether the extracted features can meaningfully represent the data.

We kept 10% of the training set for the validation of the methods, and the rest of the data provided in the training set was used to train the models in order to bring the pretrained models to train them on our problem.

3.1.2. Method 2 – Siamese Neural Networks

A Siamese neural network [14] is specifically designed to learn similarity between pairs of inputs. It consists of two identical subnetworks (or branches) that share the same weights and parameters. Each subnetwork receives one of the input samples and transforms it into a feature representation, or embedding. These embeddings are then compared using a distance metric such as the Euclidean distance to determine how similar the inputs are. The network is typically trained with contrastive loss, which minimizes the distance for similar pairs and maximizes it for dissimilar ones.

We adopted a Siamese neural network architecture trained with contrastive loss to identify which subset of the training data was used to generate each synthetic image in the testing set. This approach enabled the model to learn a discriminative embedding space where real and synthetic images originating from the same training subset are mapped close together, while those from different subsets are pushed farther apart. The input pairs consisted of one real image and one synthetic image. Positive pairs were formed from images originating from the same training subset, while negative pairs came from different subsets. Each image was preprocessed by and normalizing pixel intensities to the [0, 1] range. These pairs were used to train the Siamese model in different experimental setups.

The structure of the network that was used for the results presented in this paper is depicted in Figure 2 and it was trained over 30 epochs using the Adam optimizer and contrastive loss. The architecture of each Siamese branch included a series of convolutional layers followed by max-pooling, culminating in a dense layer with L2 normalization to produce compact embeddings. Specifically, the network consisted of four convolutional blocks (with 16, 32, 64, and 128 filters), each followed by ReLU activation and max-pooling, and a final dense layer of 256 units with L2 normalization. The final step was a distance computation layer, which calculated the Euclidean (L2) distance between the two embeddings. Validation was performed using a set of positive and negative pairs to ensure that each class was represented. After training, similarity scores were thresholded to compute validation accuracy, distinguishing whether image pairs originated from the same training subset.

For inference on the testing set, we first computed class centroids by averaging the embeddings of real training images for each subset. Each testing image was passed through the embedding network to generate its feature representation, which was then compared to all centroids using Euclidean distance. The label of the closest centroid was assigned as the predicted subset for the testing image.

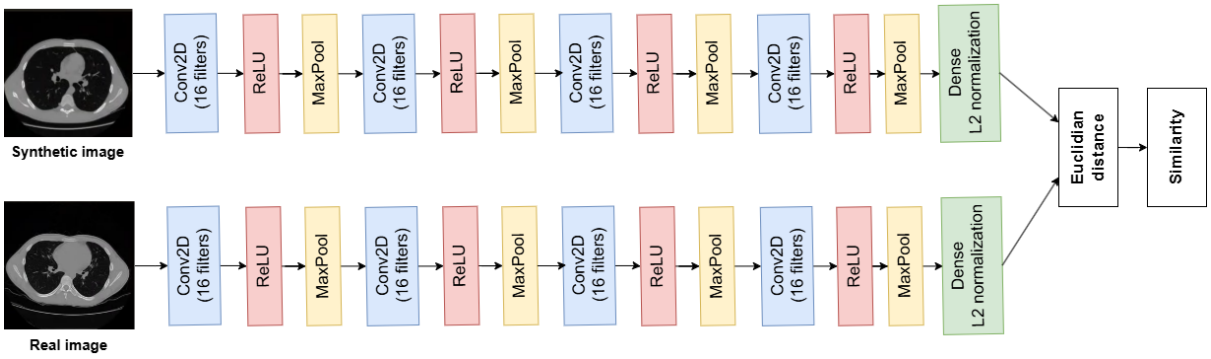


Figure 2: Block diagram of the proposed Siamese neural network.

3.2. Subtask 1 – "Detect Training Data Usage"

We applied the same Siamese neural network architecture used in subtask 2 shown in Figure 2 to address subtask 1. In this setup, both branches of the network received synthetic images from the training dataset. For positive pairs, each synthetic image was matched with a real image that had been used to train the generative model. For negative pairs, the synthetic image was paired with a real image that was not part of the training data for that specific subset.

4. Results and Discussion

4.1. Subtask 1 – "Detect Training Data Usage"

Training dataset

We evaluated the proposed Siamese Neural Network in two different training configurations and obtained Cohen's kappa scores of 0.35 and 0.50, respectively. These scores indicate a fair to moderate agreement with the ground truth in identifying whether a given real image was used to train the generative model. While not indicative of high precision, these results suggest that the model is able to detect subtle patterns or "fingerprints" left by the training data in the synthetic images. A kappa of 0.50, in particular, reflects a meaningful level of distinction between used and unused training data, although there is still room for improvement in terms of discriminative reliability.

Testing dataset

Table 3 presents the results of two configurations of the proposed Siamese NN evaluated on the testing dataset for subtask. The best-performing run (ID 1696) achieved a Cohen's kappa of 0.036 with an accuracy of 51.80%, indicating only marginally better-than-random performance. The second configuration (ID 1492) resulted in a negative kappa score (-0.044), suggesting performance slightly worse than chance.

While both models yielded similar F1-scores ($\sim$0.54), the low kappa values reflect a weak agreement with the ground truth and suggest that the network struggles to consistently distinguish between images that were used in training and those that were not. This outcome highlights the challenging nature of subtask 1, where the distinction between used and unused training images is subtle and may not be reliably captured through visual similarity alone. These results point to the need for more sensitive representations or alternative model strategies capable of detecting the latent "fingerprints" left by GAN training processes.

4.2. Subtask 2 – "Identify Training Data Subsets"

Training dataset

This section presents the results obtained using both proposed methods to address this subtask. Table 4 summarizes the performance on the training dataset using feature clustering approaches. Based on the validation accuracy, evaluated on a held-out portion of the training data. Among the three classification strategies, SVM consistently achieved the highest validation accuracy across all feature types, with ResNet50 + SVM yielding the best overall performance (52.59%). This suggests that supervised classification is more effective in capturing the discriminative patterns embedded in the pretrained features, while unsupervised clustering methods were less reliable for this task.

To further analyze the classification behavior of the SVM with different feature representations, we include confusion matrices (Figures 3a, 3b, 3c, and 3d) for the top-performing configurations: DenseNet + SVM, ResNet50 + SVM, EfficientNet + SVM, and MobileNetV2 + SVM. These matrices provide insight into class-wise performance and highlight specific confusion patterns. DenseNet + SVM (Figure 3a): The classifier performed strongly for class t3, with 1,573 correct predictions, and showed good performance for t1 as well. However, misclassifications were observed: a large number of t4 samples were predicted as t3 (576), and t2 samples as t1 (335). ResNet50 + SVM (Figure 3b): This configuration yielded the most balanced performance across all classes. It achieved high accuracy for t1 (1,292), t2 (1,070), and t3 (1,041), with relatively lower off-diagonal confusion compared to other models. Some confusion persisted between t4 and both t2 and t3, but to a lesser extent than in other configurations. The diagonal dominance in the confusion matrix suggests that ResNet50 features are more discriminative and lead to more consistent classification. EfficientNetB0 + SVM (Figure **??**): EfficientNetB0 showed high accuracy for t1 and t3 (1,364 and 1,164 respectively), but was less consistent for t4, where 706 samples were incorrectly predicted as t1. There was also mild confusion between t2 and t1. The model seems to capture certain class-specific features well but lacks robustness for classes with overlapping characteristics, particularly those positioned in the middle of the label set. MobileNetV2 + SVM (Figure 3d): MobileNetV2 demonstrated generally stable performance across all classes, with good classification of t3 (1,379 correct). However, t2 and showed notable confusion with t1 and t3.

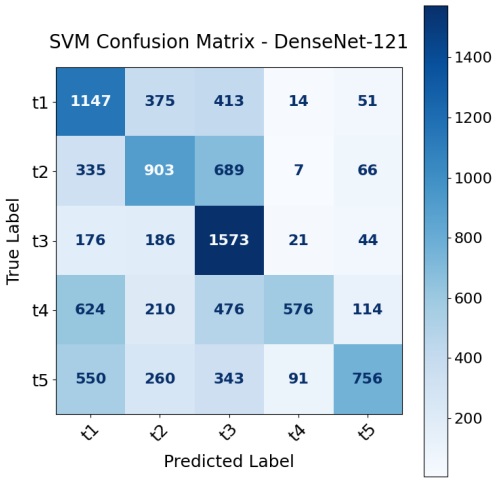
In order to visually analyze the clusters obtained with k-means and hierarchical clustering methods, we plotted the PCA-2D projections for both real and synthetic images from the training dataset.

Figures 4–7 display the features projected into a 2D space using PCA with two components. These visualizations correspond to different feature extractors (MobileNetV2, ResNet-50, DenseNet-121, and EfficientNet) and provide insight into how well the clustering algorithms separate and group data.

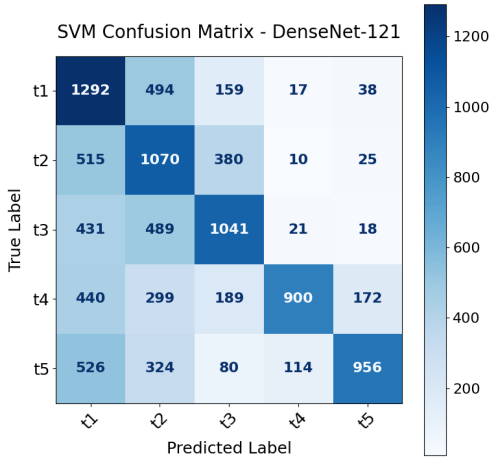
By comparing these figures alongside the quantitative metrics in Table 4, we observe that the overall cluster structures produced by k-means and hierarchical clusteringare generally similar, confirming the stability of clustering results across methods. Figure 4–a and 4–b show the PCA-2D projections using the features extracted using DenseNet-121 network. Both methods show a broader spread in the PCA space, especially along the first principal component. Despite this dispersion, both clustering algorithms continue to group real and synthetic samples in a similar manner. Figures 5–a and 5–b illustrate the clustering results based on EfficientNet features. These projections show high overlap between real and synthetic samples across most clusters. The results obtained using MobileNetV2, depicted in Figures 6–a and 6–b show the PCA-2D projections using features extracted from MobileNetV2. Both clustering methods yield clearly defined cluster boundaries, with a moderate degree of overlap between real and synthetic images. The visual coherence of clusters across both methods suggests MobileNetV2 provides a relatively disentangled feature space for both domains. Finally, Figures 7–a and 7–b illustrate the clustering results based on ResNet-50 features. Both clustering methods capture similar structural divisions in the embedding space, though the k-means clusters appear slightly more isotropic and

Table 3
Results of the proposed methods on the testing dataset for "Detect Training Data Usage" subtask

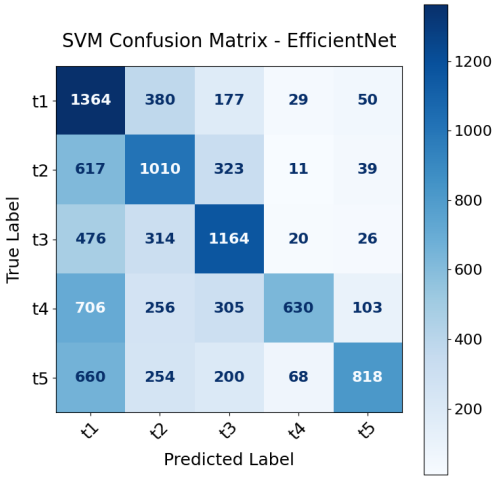
ID #	Cohen's Kappa	Accuracy	Precision	Recall	F1-score
1696	0.0360	0.5180	0.5162	0.5720	0.5426
1492	-0.0440	0.4780	0.4828	0.6200	0.5429



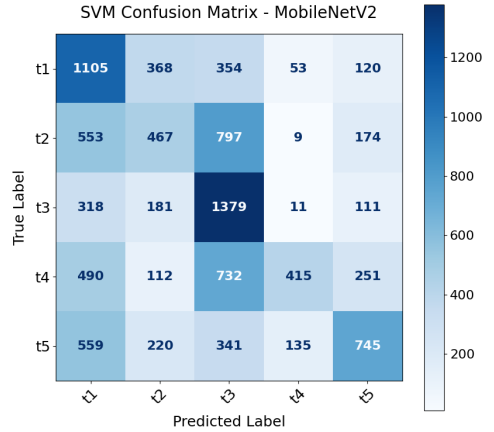
(a) DenseNet121 + SVM



(b) ResNet50 + SVM



(c) EfficientNetB0 + SVM



(d) MobileNetV2 + SVM

Figure 3: Confusion matrices for different CNN feature extractors used with SVM.

Table 4
Results of the proposed feature clustering methods on the training dataset for "Identify training data subsets" subtask

#	Feature extraction method	# features	Classification method	Validation accuracy
1	MobileNetV2	1280	k-means	0.2019
2	MobileNetV2		hierarchical clustering	0.1952
3	MobileNetV2		SVM	0.4111
4	ResNet50	2048	k-means	0.1970
5	ResNet50		hierarchical clustering	0.2003
6	ResNet50		SVM	0.5259
7	EfficientNet	1280	k-means	0.2072
8	EfficientNet		hierarchical clustering	0.2062
9	EfficientNet		SVM	0.1986
10	DenseNet	1024	k-means	0.1916
11	DenseNet		hierarchical clustering	0.2376
12	DenseNet		SVM	0.4955

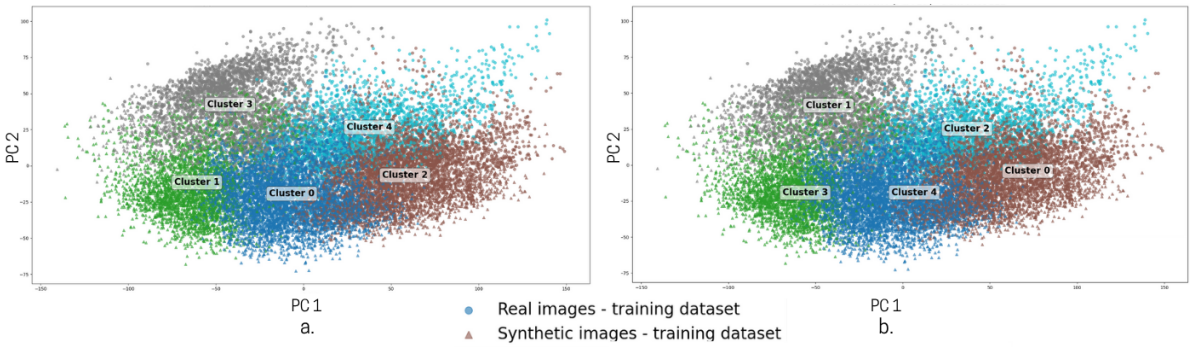


Figure 4: The PCA-2D projections for the features extracted using the Densenet network: a) hierarchical clustering, b) k-means.

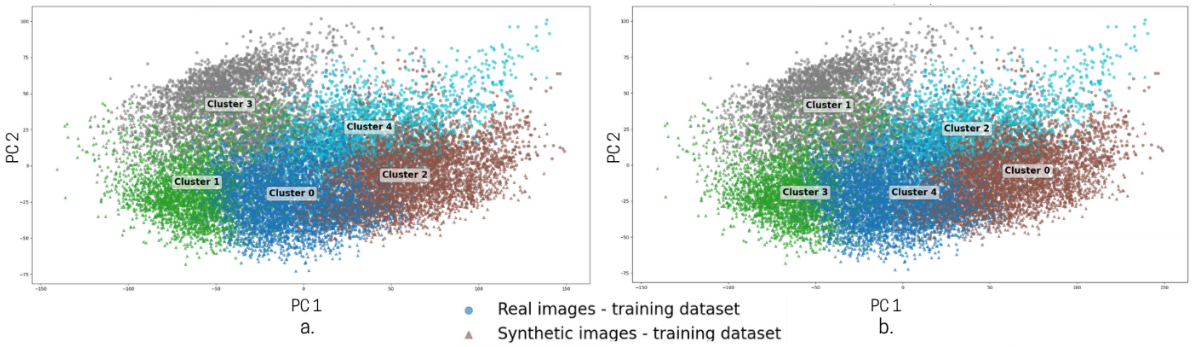


Figure 5: The PCA-2D projections for the features extracted using the EfficientNet network: a) hierarchical clustering, b) k-means.

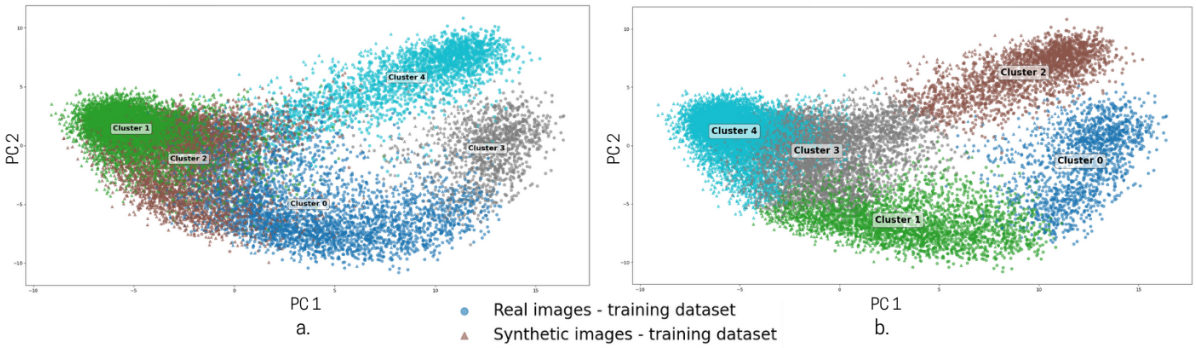


Figure 6: The PCA-2D projections for the features extracted using the MobileNet network: a) hierarchical clustering, b) k-means.

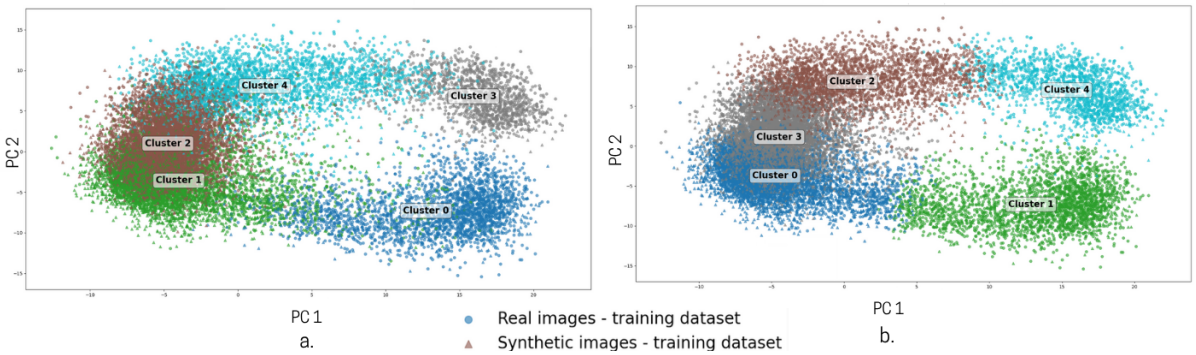


Figure 7: The PCA-2D projections for the features extracted using the ResNet network: a) hierarchical clustering, b) k-means.

Table 5
Results of the proposed methods on the testing dataset for "Identify training data subsets" subtask

Method	ID #	Accuracy	Precision	Recall	F1	Specificity
Siamese Neural Networks	1396	**0.9904**	0.9904	0.9904	0.9904	0.9972
DenseNet + SVM	1271	**0.4904**	0.5691	0.4904	0.4832	0.8752
EfficientNet + SVM	1269	**0.4912**	0.5821	0.4912	0.4934	0.8743
ResNet + SVM	1268	**0.5236**	0.5981	0.5236	0.5326	0.8798
MobileNetV2 + SVM	1267	**0.4111**	0.4644	0.4111	0.3945	0.8547

evenly distributed. Overall, these visual analyses support the conclusion that both clustering methods produce consistent structures across different feature extractors. Moreover, they confirm that synthetic data is generally well aligned with real data in the learned feature spaces.

These results guided our selection of runs submitted – we selected only the best-performing configurations (IDs 3, 6, 9, and 12) for official submission.

When it comes to Siamese neural network described in Section 3, we obtained an accuracy of 100%.

Testing dataset

Table 5 reports the performance of the proposed methods on the official testing dataset for subtask 2. Among all evaluated approaches, the Siamese Neural Network (ID 1396) achieved a significantly higher performance than all feature clustering methods, reaching an accuracy of 99.04%, with equally high precision, recall, F1 score, and a specificity of 99.72%. This confirms the effectiveness of the Siamese architecture in learning fine-grained similarity patterns between real and synthetic images.In contrast, the feature-based SVM classifiers, while substantially less accurate, showed varying degrees of performance depending on the feature extractor used. ResNet + SVM (ID 1268) achieved the best accuracy among the clustering-based methods (52.36%), followed by EfficientNet and DenseNet configurations. Notably, MobileNetV2 + SVM lagged behind with the lowest performance (41.11% accuracy), indicating that its learned features may be less separable for this task. Precision and specificity trends followed similar patterns, reinforcing that deeper architectures such as ResNet and DenseNet yield more discriminative embeddings.

Overall, these results confirm that supervised similarity learning using a Siamese architecture substantially outperforms unsupervised and supervised clustering approaches for identifying the origin of synthetic medical images.

5. Conclusions

In this paper, we presented the participation of the AI Multimedia Lab in the third edition of the ImageCLEFmedical GANs 2025 task, which focuses on identifying traces of real-image usage in synthetic medical images generated by GANs. Our contributions addressed both subtasks of the challenge using a combination of traditional machine learning and deep learning techniques.

For subtask 1, which aimed to detect whether a real image was used in the training of a GAN, we applied a Siamese neural network trained to differentiate used versus unused real images based on paired similarity with synthetic outputs. Although the training results suggested the model could identify subtle training data patterns (kappa up to 0.50), performance on the official test set was limited (a kappa score of 0.036), highlighting the difficulty of the task and the need for more sensitive modeling approaches.

In subtask 2, which required identifying the specific real-image subset used to generate each synthetic image, we proposed two complementary approaches: (1) a feature clustering pipeline using pretrained models and traditional classifiers, and (2) a Siamese neural network trained to learn subset-specific embeddings. Among the evaluated methods, the Siamese network achieved the best performance by far, with an accuracy of 99.04% on the test set, demonstrating its effectiveness in learning discriminative patterns for subset attribution. Feature-based SVM classifiers performed reasonably well, with ResNet50 features achieving over 52% accuracy, but were clearly outperformed by the Siamese approach.

Overall, our results show that while detecting whether an image was used in GAN training remains a challenging problem, identifying the origin subset of synthetic images is feasible with high accuracy using supervised similarity learning.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4o in order to: Grammar and spelling check and improve writing style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

6. Acknowledgments

The work of Alexandra Andrei was supported by a grant of the Ministry of Research, Innovation and Digitization, CCCDI - UEFISCDI, project number PN-IV-P6-6.3-SOL-2024-0049, within PNCDI IV. The work of Mihai Gabriel Constantin was supported by a grant of the Ministry of Research, Innovation and Digitization, CCCDI - UEFISCDI, project number PN-IV-P6-6.3-SOL-2024-0060, within PNCDI IV.

References

[1] 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.

[2] A. Andrei, A. Radzhabov, D. Karpenka, Y. Prokopchuk, V. Kovalev, B. Ionescu, H. Müller, Overview of 2024 imageclefmedical gans task–investigating generative models' impact on biomedical synthetic images, in: CLEF2024 Working Notes, CEUR Workshop Proceedings, CEUR-WS. org, Grenoble, France, 2024.

[3] B. Ionescu, H. Müller, A.-M. Drăgulinescu, W.-W. Yim, A. Ben Abacha, N. Snider, G. Adams, M. Yetisgen, J. Rückert, A. García Seco de Herrera, et al., Overview of the imageclef 2023: multimedia retrieval in medical, social media and internet applications, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2023, pp. 370–396.

[4] B. Ionescu, H. Müller, A.-M. Drăgulinescu, J. Rückert, A. Ben Abacha, A. García Seco de Herrera, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, et al., Overview of the imageclef 2024: Multimedia retrieval in medical applications, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2024, pp. 140–164.

[5] 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.

[6] 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.

[7] 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.

[8] A.-G. Andrei, B. Ionescu, Aimultimedialab at imageclefmedical gans 2023: Determining" fingerprints" of training data in generated synthetic images., in: CLEF (Working Notes), 2023, pp. 1379–1386.

[9] 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, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[10] 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.

[11] B. Koonce, B. Koonce, Resnet 50, Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization (2021) 63–72.

[12] B. Koonce, B. Koonce, Efficientnet, Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization (2021) 109–123.

[13] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, K. Keutzer, Densenet: Implementing efficient convnet descriptor pyramids, arXiv preprint arXiv:1404.1869 (2014).

[14] D. Chicco, Siamese neural networks: An overview, Artificial neural networks (2021) 73–94.