Reverse Engineering Generative Fingerprints in Medical Images: A Deep Learning Approach to Training Data Attribution

Notebook for the ImageCLEF Lab at CLEF 2025

Mitra Barve[1,†], Nikita Bhedasgaonkar[1,†], Isha Shah[1,†], Sara Nambiar[1,*,†], Atharva Date[1,†] and Geetanjali Kale[1,†]

[1]*Pune Institute of Technology, Dhankawadi, Pune, Maharashtra, India*

Abstract

The growing concern of data privacy in AI models trained on sensitive medical data has led to increasing usage of synthetically generated medical data for this purpose. ImageCLEFmedical GANs 2025 investigates whether such synthetic data contains fingerprints that might be used to identify the real images that were implicitly used to generate these synthetic images thereby posing a threat to patient privacy. We used multiple approaches to identify whether a given image was part of the training set of a generative model whose outputs we had access to. The central idea was self-supervised training of Auto-Encoders and GANs on the synthetic images and performing clustering / classification on the encoder / critic features. These findings suggest that encoder-based feature representations can retain some training signal from generative models, highlighting potential risks to patient privacy. We also observed that Vision Transformers, especially when pretrained on domain-specific data, help models learn more informative representations.

Keywords

Autoencoders, DCLGANs, GANs, Latent Fingerprint Detection, Training Data Attribution, Synthetic Biomedical Images, Medical Image Privacy, Deep Representation Learning

1. Introduction

In recent years, the rise of generative models has enabled the creation of highly realistic synthetic medical images, providing valuable assistance for applications such as data augmentation. This is particularly beneficial in scenarios where there is limited access to real medical data. However, this advancement introduces a critical privacy concern: can synthetic images unintentionally leak sensitive information about the real training data used in their generation?

Our team, **Neural Nexus**, investigates this question through our participation in the ImageCLEFmedical GANs 2025 challenge, specifically *Subtask 1: Detect Training Data Usage* [1, 2].

To explore potential training data leakage, we use self-supervised representation learning methods, including both Vision Transformer (ViT)-based and CNN-based autoencoders, trained directly on synthetic image sets. We also evaluate the use of feature extractors derived from GAN critics, specifically Dual Contrastive Learning GAN (DCLGAN). Additionally, we examine the effects of supervised pretraining on external labeled tuberculosis dataset to guide encoder feature learning.

To determine whether these extracted features carry fingerprints of training data, we apply both supervised classification methods and unsupervised clustering techniques, including KMeans, Gaussian Mixture Models (GMM), and Spectral Clustering.

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

*Corresponding author.

†These authors contributed equally.

barve.mitra@gmail.com (M. Barve); nikitaedu7@gmail.com (N. Bhedasgaonkar); ishahmshah@gmail.com (I. Shah); nambiar.sara@gmail.com (S. Nambiar); atharva2718@gmail.com (A. Date); gvkale@pict.edu (G. Kale)

https://github.com/mbarve117 (M. Barve); https://github.com/NikitaAB7 (N. Bhedasgaonkar); https://github.com/ishahmshah1025 (I. Shah); https://github.com/saranambiar (S. Nambiar); https://github.com/Atharva9621 (A. Date)

Our findings demonstrate that, under specific experimental setups, it is indeed possible to distinguish between synthetic images generated from "real_used" training samples and those that were not. These results raise important concerns about data traceability and privacy in synthetic medical image generation workflows. The code required for this is publicly available in our Github Repository.[1]

2. Background

This work addresses the problem of detecting whether real medical images were used in the training of generative models, with a focus on synthetic lung CT slices produced by a Generative Adversarial Network (GAN). Specifically, Subtask 1, *Detect Training Data Usage*, aims to determine whether a given real image contributed to the training of a GAN that produced synthetic CT images. Identifying such instances is critical for assessing privacy risks, as the presence of training data "fingerprints" in generated images can indicate potential data leakage and privacy violations. The complete dataset comprises standardized lung CT scan slices in PNG format, each of size 256×256 pixels and encoded at 8 bits per pixel. The training data is divided into three folders:

- **generated**: 5,000 synthetic CT images generated by a GAN trained on real lung CT data.
- **real_used**: 100 real images that were included in the training set of the GAN.
- **real_not_used**: 100 real images that were excluded from the GAN's training process.

The test data is similarly structured and consists of:

- **generated**: 2,000 new synthetic images produced by the same GAN under the same training configuration.
- **real_unknown**: 500 real CT images, which are a mixture of "used" and "not used" images. The task is to predict, for each of these, whether it was used (label 1) or not used (label 0) during GAN training.

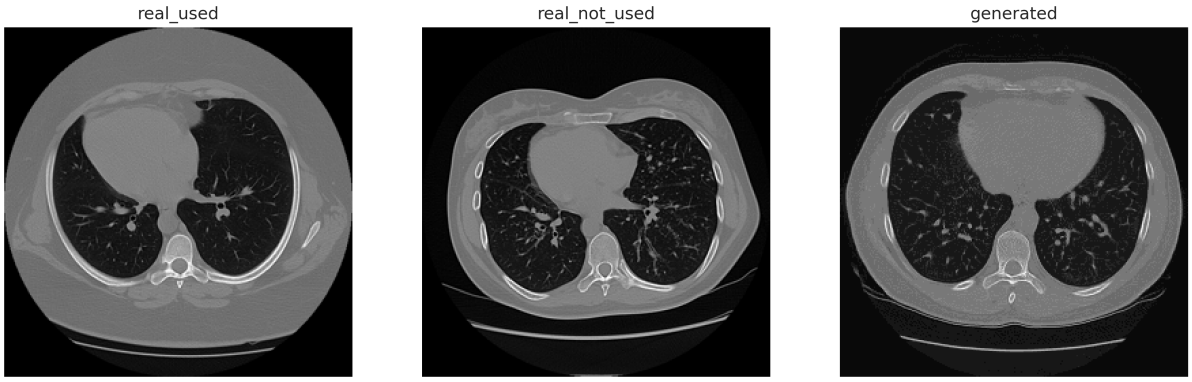

Figure 1: Task Dataset Sample

The goal is to train a model on the provided training set and produce a final output file with 500 binary predictions corresponding to the real_unknown images. A correct prediction (1 for used, 0 for not used) indicates the model successfully identified subtle signatures that distinguish between training and held-out data in the synthetic output space. The implications of this work are substantial, particularly in the context of medical imaging, where the inadvertent exposure of patient-specific information through generative models raises significant ethical and legal concerns.

[1] https://github.com/saranambiar/Neural-Nexus-ImageClef-2025

External Dataset Description

The ViT encoder and the encoder used in the spectral clustering approach was pretrained on a publicly available external tuberculosis classification dataset [3], consisting of lung CT slices in JPG/PNG format. The dataset includes images from four classes: Adenocarcinoma, Large Cell Carcinoma, Squamous Cell Carcinoma, and Normal. It is split into training (70%), validation (10%), and test (20%) sets. Pretraining on this domain-specific dataset helped the encoder learn medically relevant features prior to reconstruction tasks.

3. Methodology

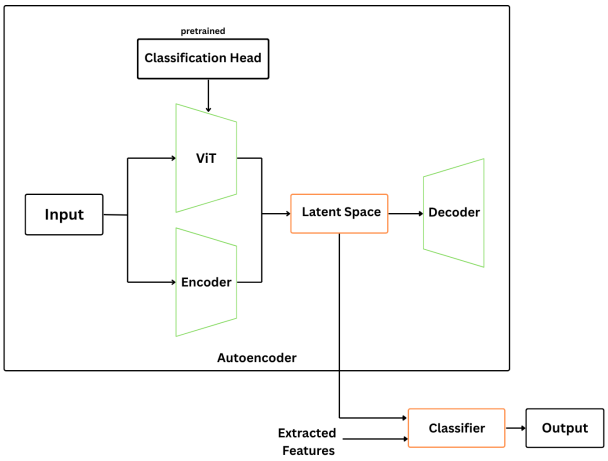

Figure 2: Diagram illustrating the architecture

3.1. Pretrained ViT Autoencoder

The proposed autoencoder architecture is composed of a Vision Transformer (ViT)-based encoder and a convolutional decoder enhanced with residual connections [4]. The encoder utilizes a modified ViT-B/16 model, where the original classification head is removed and replaced with a projection layer that maps the 768-dimensional token embeddings to a lower-dimensional latent space. Since ViT requires a 3-channel input, a lightweight 1×1 convolution is applied to convert grayscale images into a transformer suitable format for training.

Here, the encoder is pretrained on an external tuberculosis classification dataset, enabling it to extract domain-specific features[3]. Unlike conventional CNNs, the ViT encoder employs self-attention to capture long-range dependencies and global context at an early stage. This mechanism is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Here, Q, K, and V are the query, key, and value matrices derived from the token embeddings, and d_k is the dimensionality of the key vectors. This attention mechanism allows ViT to extract semantically rich and discriminative features, which are particularly crucial in medical imaging and anomaly detection, where subtle, spatially distributed patterns across the image are important [5].

The decoder reconstructs the input image from the latent representation using a sequence of transposed convolutional layers and residual blocks. These residual connections promote efficient feature propagation and training stability [6]. The decoder progressively upsamples the compact representation back to the original image resolution. Finally, a Tanh activation is applied to produce pixel values in a normalized range. After analyzing the reconstruction error distribution on the validation set, a threshold is empirically set to classify the images.

3.1.1. Experimental Setup

The experiment involved two phases: supervised pretraining of the ViT-based encoder for classification, followed by unsupervised training of a full autoencoder using the pretrained encoder.

In the first phase, the encoder, based on the ViT-B/16 architecture, was pretrained for 100 epochs on a multi-class classification task using cross-entropy loss. The model was optimized using the Adam optimizer with a StepLR scheduler, and accuracy and loss were tracked on both training and validation sets. Training was conducted using PyTorch with GPU acceleration.

In the second phase, the pretrained encoder weights were reused to initialize an autoencoder, which was trained to minimize the Mean Squared Error (MSE) between input and reconstructed CT images. The decoder consisted of transposed convolutions and residual blocks to progressively upsample the latent representation back to the image resolution. Training was conducted for 10 epochs using the Adam optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}, and convergence was monitored using average reconstruction loss per epoch.

3.2. Spectral Clustering on AutoEncoder Features

We implemented a convolutional autoencoder to learn compact representations of the given generated images. The encoder was first pretrained in a supervised fashion on an external tuberculosis classification dataset [3], allowing it to extract domain-relevant features. This encoder was then integrated into a full autoencoder and fine-tuned on the available data using reconstruction loss. Specifically, for an input image $x \in \mathbb{R}^{C \times H \times W}$, the encoder $E(\cdot)$ produced a latent representation $z = E(x)$, which was then passed through the decoder $D(\cdot)$ to reconstruct the image: $\hat{x} = D(z)$. The training objective was to minimize the reconstruction MSE:

$$\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|_2^2. \tag{2}$$

After training, the encoder was used to extract latent features z for all images in the dataset. We then applied various clustering algorithms, including KMeans, Gaussian Mixture Models (GMM), and Spectral Clustering, to group these features into two clusters corresponding to the *real_used* and *real_not_used* labels. Among these, Spectral Clustering yielded the highest performance in terms of both accuracy and F1-score.

Spectral Clustering [7] operates by computing a similarity graph $G = (V, E)$ over the features, forming the graph Laplacian $L = D - A$, where A is the affinity matrix and D is the degree matrix. The eigenvectors corresponding to the k smallest eigenvalues of the normalized Laplacian $L_{\text{sym}} = D^{-1/2} L D^{-1/2}$ are used to embed the data, followed by KMeans:

$$L_{\text{sym}} = I - D^{-1/2} A D^{-1/2}. \tag{3}$$

Clustering on the features extracted from the encoder indicated significant differences between the used and unused images.

3.2.1. Experimental Setup

We used an encoder with 4 Residual Blocks (having 32, 64, 128, 256 filters) each consisting of multiple Convolutional, BatchNorm, Dropout and Pooling layers. We initially pretrained it on an external classification dataset by appending a linear classification head to this encoder and training it for 100 epochs against Cross Entropy Loss using the Adam Optimizer and Step LR Scheduler.

We built the autoencoder using this pre-trained encoder and a decoder with 4 Residual Blocks (having 256, 128, 64, 32 filters). The Auto-Encoder was then trained for 100 epochs using L1 loss to model reconstruction error and the Adam optimizer. Spectral clustering was performed on the features produced by the trained encoder to generate final results.

3.3. ResNet Autoencoder and Feature-Based Detection Framework

3.3.1. ResNet Autoencoder

We adopted a ResNet-based convolutional autoencoder with residual connections to enable stable training and deep feature extraction. The encoder consists of a total of four Conv2D layers with increasing filters (32→64→128→256) and stride 2, each followed by a residual block comprising two 3×3 Conv2D layers, BatchNorm, and LeakyReLU activation, along with a shortcut connection.

A GlobalAveragePooling2D layer reduced the spatial dimensions, and a Dense layer maped the result to a 512-dimensional latent vector $z = E(x)$. Following Wickramasinghe et al. [8], this residual design prevented performance degradation as the depth increased, enabling deeper networks with better generalization.

The decoder inverts this process: the latent vector z is passed through a Dense layer and reshaped to $16 \times 16 \times 128$, followed by three Conv2DTranspose layers (128→64→32), each paired with a residual block. A final Conv2DTranspose layer with 3 filters and sigmoid activation reconstructed the image. The network was trained end-to-end using the mean absolute error (MAE) loss.

Optimization is performed with Adam optimizer, allowing the encoder to effectively learn compact representations of GAN-generated samples. A similar multi-scale residual autoencoder design was successfully applied by Li et al. [9] for CT lung nodule classification, further supporting its applicability in capturing fine-grained medical image features.

3.3.2. Feature-based classification

The ResNet encoder, trained on synthetic images produces 512-dim latent embeddings. However, these features must be enriched. We accomplish this using handcrafted descriptors such as first-order radiomic statistics, GLCM-based texture measures [10], wavelet subband energies [11], Gabor filter responses [12] and morphological characteristics.

Next, we train a Random Forest classifier on the labeled real images using the enriched features. We compute Mahalanobis distances [13] from each real image to the synthetic feature distribution, based on the top-K most imformative features. This is done to encourage the model to better distinguish used and unused images. These distances are used as a meta anomaly signal.

As per the classifier's probability outputs, predictions are generated using thresholds selected via cross-validation methods. This approach helps optimize F1-score and Cohen's Kappa.

3.3.3. Mathematical Expressions for Handcrafted Features

GLCM Contrast

$$\text{Contrast} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 \, P(i,j) \tag{4}$$

where $P(i,j)$ is the normalized co-occurrence probability of gray levels i and j, and N_g is the number of gray levels.

Wavelet Subband Energy

$$E_s = \sum_{m=1}^{M} \sum_{n=1}^{N} |W_s(m,n)|^2 \tag{5}$$

where $W_s(m,n)$ denotes the wavelet coefficient at position (m,n) in subband s, and E_s is the total energy of that subband. M and N are the dimensions of the subband.
Multiple subbands (e.g., horizontal, vertical, diagonal details at various scales) can be used to form a feature vector of subband energies.

Gabor Filter Response

$$G(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \cos\left(2\pi \frac{x'}{\lambda} + \phi\right) \tag{6}$$

$$x' = x\cos\theta + y\sin\theta, \quad y' = -x\sin\theta + y\cos\theta \tag{7}$$

where λ is the wavelength, θ is the orientation, ϕ is the phase offset, σ is the Gaussian standard deviation, and γ is the spatial aspect ratio.

Mahalanobis Distance

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})} \tag{8}$$

where $\mathbf{x}$ is the feature vector, $\boldsymbol{\mu}$ is the mean of the distribution, and $\boldsymbol{\Sigma}$ is the covariance matrix.

3.3.4. Experimental Setup

The above-mentioned ResNet-based convolutional autoencoder was implemented using the TensorFlow deep learning framework. We trained the model for 50 epochs using the Adam optimizer with default hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a fixed learning rate of 1×10^{-4}. The training was performed with a batch size of 32 on normalized RGB images resized to 128×128 pixels. The mean absolute error loss function was used to improve pixel-wise reconstruction accuracy.

The software environment included Python 3.8, TensorFlow 2.12, and CUDA 11.8. We observed the model convergence within 40 to 45 epochs. GPU acceleration was utilized throughout the training process to efficiently handle high-dimensional image data and support deep network training.

The meta-classification piece was made using `Python 3.11` and `scikit-learn 1.4`, `NumPy 1.26`, `Pandas 2.2`, and `OpenCV 4.9` for preprocessing and feature extraction. Latent features were extracted from the previously mentioned autoencoder. Gabor filters and wavelet transforms were calculated on default settings. The Random Forest classifier was trained with 100 estimators and default depth, and with Gini impurity as the split criterion. Mahalanobis distances were calculated using the empirical covariance matrix corresponding to the synthetic image features. All experiments were executed on the Kaggle GPU runtime with CUDA 11.8 support.

3.4. Dual-Contrastive GAN for Feature Extraction

We use a DCLGAN framework to extract and compare discriminator-driven feature maps across three CT image domains. By analyzing attention patterns from intermediate discriminator activations, we estimate semantic similarity to infer which real images influenced the generator's learning. The generator starts with a 7×7 convolution, followed by downsampling (128→256 channels) and ResNet blocks with identity shortcuts [14]. It then upsamples (256→128→64) and ends with a 7×7 convolution and Tanh activation.

The PatchGAN discriminator consists of five Conv2D layers (64→1 channels) with LeakyReLU and Instance Normalization [14]. We attach forward hooks to the deepest post-activation layers to extract feature maps, enabling spatial attention analysis [15]. This supports evaluating feature overlap and understanding the generator's dependency on real images. While this approach is not directly linked to a specific submission ID, initial testing indicated promising performance for feature extraction in CT scan imagery.

3.4.1. Experimental Setup

The model was trained for 200 epochs using the Adam optimizer with TTUR: learning rates for both the generator and discriminator were set to 1×10^{-4} and 4×10^{-4}, respectively. Cosine annealing schedulers were used for both optimizers. Spectral normalization was applied to all Conv2D and Linear

layers in the discriminator to enforce 1-Lipschitz continuity. We conducted training with batch size 1 using CUDA acceleration on PyTorch.

Training includes three loss functions:

- **Hounsfield Unit (HU) Loss:** Preserves clinically relevant CT intensity distributions by aligning histogram distributions of real and synthetic images for each slice. Let $P_r(i)$ and $P_f(i)$ denote the normalized histogram values of the i^{th} bin for real and fake images, respectively. The HU loss is computed as the Kullback-Leibler divergence over N histogram bins:

$$\mathcal{L}_{\text{HU}} = \sum_{i=1}^{N} P_r(i) \log \left(\frac{P_r(i) + \epsilon}{P_f(i) + \epsilon} \right) \tag{9}$$

  where ϵ is a small constant added for numerical stability.
- **PatchNCE Loss:** Enforces localized contrastive alignment between real and generated patches, improving the fidelity of fine-grained features [15].
- **Feature Matching Loss:** Stabilizes adversarial training by minimizing the L1 distance between discriminator feature activations for real and synthetic inputs.

We apply gradient penalty regularization to enhance training stability. This setup ensures that precise attention overlaps are captured between discriminator feature activations, supporting our hypothesis that *real_used* and *generated* images exhibit greater alignment than *real_not_used*.

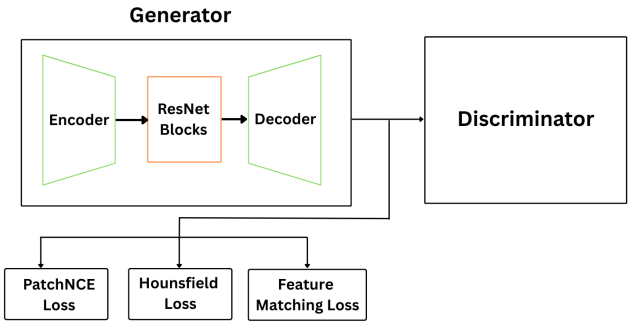

Figure 3: DCLGAN with Patch-Based Discriminator Architecture

4. Results

The quantitative results in Table 1 demonstrate the superiority of the ViT-based encoder, which obtained the highest F1-score (0.568) and Cohen's Kappa (0.148). The encoder learns anatomical priors and pathological patterns unique to thoracic imaging after being pretrained on a lung CT dataset. During reconstruction, the model can extract more significant and medically relevant representations thanks to this initialization. ViT's self-attention mechanism, in contrast to CNNs, enables the model to concentrate on contextually related but spatially distant regions, which is essential for detecting subtle or diffuse abnormalities in chest CT scans. Furthermore, by stabilizing training and maintaining low-level spatial details, the residual decoder architecture makes high-fidelity reconstruction possible. Only the most important features are kept and rebuilt thanks to the compact latent space's function as a semantic bottleneck.The higher Cohen's Kappa score indicates that the model is more in agreement with ground-truth labels as a result of these factors working together to minimize incorrect classifications and

help the model differentiate between structurally similar classes. Reconstructions produced by this architectural synergy are both aesthetically pleasing and diagnostically significant.

The spectral clustering approach outperformed the ResNet-based framework in F1-score (0.442 vs. 0.437) and Cohen's Kappa (0.072 vs. 0.032), despite being unsupervised, proving the effectiveness of clustering in learned latent spaces. Lower recall, on the other hand, implies sensitivity trade-offs in capturing subtler image characteristics, perhaps as a result of noise in fine-grained structural regions that are important for medical imaging or more difficult-to-cluster borderline cases.

The modest results of the ResNet autoencoder with handcrafted features and anomaly scoring via Mahalanobis distance are probably the result of either domain shift effects between real and synthetic samples or the limited generalization of handcrafted features. Handcrafted features frequently lack the flexibility to adapt to unseen data distributions, especially in complex medical contexts, even though the architecture captures low and mid-level patterns fairly well.

The DCLGAN method demonstrated excellent qualitative performance in capturing intensity distributions and attention overlaps, indicating its potential for targeted feature attribution and interpretable GAN evaluation in CT domains, even though this was not evident in the leaderboard submissions.

It is important to note that not all submissions made during the challenge are discussed in this working note. We have intentionally focused on presenting only those configurations that yielded the most competitive results in terms of quantitative metrics or qualitative insights. Lower-performing or exploratory runs have been excluded to maintain clarity and focus.

Table 1
Final Results of Our Methods

Test ID	Method Name	Cohen's Kappa	Accuracy	Precision	Recall	F1-Score
1878	ViT-Based Encoder	0.148	0.574	0.569	0.604	0.568
1880	Spectral Clustering on Autoencoder Features	0.072	0.536	0.554	0.368	0.442
1881	ResNet Autoencoder + Feature-Based Detection	0.032	0.516	0.522	0.376	0.437

5. Conclusion

In this paper, we explore the ImageCLEF GAN 2025 task of identifying GAN fingerprints on training data. Specifically, we determined whether a particular image had been part of the training set for a generative model used to create the given synthetic images. We used multiple methods where we trained VIT/Resnet-based AutoEncoders or GANs on the provided Synthetic Images to capture the Generated distribution and performed clustering/classification on the features extracted from the Encoder/GAN Critic. Using a classifier on the features extracted from a VIT-based AutoEncoder provided the best results resulting in a Cohen's Kappa Score of 0.148.

Acknowledgments

Thanks to SCTR's Pune Institute of Computer Technology (PICT), Pune, India for their support and the resources provided, which greatly assisted in the research and preparation of this work.

Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-4 and Gramby in order to: Grammar and spelling check. Further, the author(s) used X-AI-IMG for figures 3 and 4 in order to: Generate images.

After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) 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] M. Hany, Chest ct-scan images dataset, https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images/data, 2020. Accessed: 2025-05-30.

[4] C. Prabhakar, H. B. Li, J. Yang, S. Shit, B. Wiestler, B. Menze, Masked autoencoders are effective for medical image reconstruction and synthesis, arXiv preprint arXiv:2301.07382 (2023). URL: https://arxiv.org/abs/2301.07382.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, International Conference on Learning Representations (ICLR) (2021). URL: https://arxiv.org/abs/2010.11929.

[6] 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. doi:10.1109/CVPR.2016.90.

[7] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, Advances in neural information processing systems 14 (2001). doi:10.5555/2980539.2980649.

[8] C. S. Wickramasinghe, D. L. Marino, M. Manic, Resnet autoencoders for unsupervised feature learning from high-dimensional data: Deep models resistant to performance degradation, IEEE Access (2021). doi:10.1109/ACCESS.2021.3064819.

[9] F. Li, S. N. Y. Sherazi, Y. Zhang, Z. Wu, A new multi-scale dilated deep resnet model for classification of lung nodules in ct images, 2022. doi:10.1145/3507971.3507988.

[10] S. K. . D. I. Haralick, R.M., Textural features for image classification., IEEE Transactions on Systems, Man, and Cybernetics (1973). doi:10.1109/TSMC.1973.4309314.

[11] S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation., IEEE Transactions on Pattern Analysis and Machine Intelligence (1989). doi:10.1109/34.192463.

[12] J. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters., Journal of the Optical Society of America A (1989). doi:10.1364/josaa.2.001160.

[13] J.-R. D. . M. D. De Maesschalck, R., The mahalanobis distance., Chemometrics and Intelligent Laboratory Systems (2000).

[14] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi:10.48550/arXiv.1611.07004.

[15] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, in: Proceedings of the International Conference on Machine Learning (ICML), 2019. doi:10.48550/arXiv.1805.08318.