Evaluating of the Privacy of Images Generated by ImageCLEFmedical GAN 2025 Using Similarity Classification Method Based on Image Enhancement and Deep Learning Model

Notebook for the ImageCLEF Lab at CLEF 2025

Haojie Zuo, Xiaobing Zhou*

School of Information Science and Engineering, Yunnan University, Kunming 650504, Yunnan, China

Abstract

The ImageCLEFmed GAN 2025 task aims to detect whether the synthesized medical images contain "fingerprints" from the training data. In this paper, we adopted a similarity classification method based on image enhancement and deep learning models to determine which real images were used in the training process by comparing the similarity between real and synthetic images. We preprocessed the images using multiple image enhancement techniques (such as Gaussian filtering, Hessian matrix, Laplacian operator, and bilateral filtering). Then, we used convolutional neural networks (CNN) and ResNet50 models to extract image features and calculate the similarity between images. Through experiments on the validation set, our similarity classification method achieved excellent accuracy and F1 score performance. In the submitted results, our best F1 score was 0.633.The best kappa score is -0.016.It proves that the method used can effectively distinguish between "used" and "unused" images. Our experimental results show that it is possible to successfully identify the real images used to generate synthetic images through image enhancement and deep learning models.Our code is available at https://github.com/Qqiiiii/ImageCLEF.git.

Keywords

Image Enhancement, Deep Learning, CNN, ResNet50, Similarity Calculation, Medical Imaging

1. Introduction

In the field of medical image analysis, deep learning models[1][2] have shown significant potential in assisting diagnosis and treatment, especially in automating the analysis and interpretation of medical images. However, training these deep learning models usually requires a large amount of data, and obtaining high-quality medical image data often faces privacy and data sharing challenges. To address this problem, generative models such as generative adversarial networks (GANs) have been proposed and widely used to synthesize medical image data to enhance the diversity and quality of data sets, thereby helping to train more powerful models.

Although generative models can significantly improve data diversity when generating synthetic images, synthetic images may inadvertently expose sensitive information in the training data during the process of learning data distribution, thus bringing the risk of privacy leakage.[3][4][5] Recent studies have shown that by analyzing the generated images, hidden "fingerprints" can be identified, which may point to the source of the training images. Therefore, in the field of medical imaging, ensuring that synthetic images do not leak patient privacy has become an important research topic.

In this context, ImageCLEFmed GAN 2025[6][7] introduces a new subtask, which aims to determine whether a specific real image has been used to train a generative model by analyzing synthetic medical images. This task requires determining which real images have been used in the generation process by calculating the similarity between synthetic and real images.Our team's username is ZOQ.To address

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

*Corresponding author.

12024215156@stu.ynu.edu.cn (H. Zuo); zhouxb@ynu.edu.cn (X. Zhou*)

0009-0005-2755-7575 (H. Zuo); 0000-0003-1983-0971 (X. Zhou*)

this task, we propose a similarity classification method based on image enhancement and deep learning models.

Our method first performs multiple image enhancement processes on the image, including Gaussian filtering, Hessian matrix, Laplacian operator, and bilateral filtering to enhance the details and features of the image.[8][9][10][11][12] Then, convolutional neural network (CNN) and ResNet50 are used to extract high-dimensional features of the image and calculate the similarity between the real image and the synthetic image. In the experiment, we used similarity calculation based on feature extraction of the deep learning model. This method can effectively distinguish between "used" and "unused" images.

Experimental results on the validation set show that our similarity classification method achieves excellent performance in both accuracy and F1 score, with the best F1 score being 0.633. This proves that the adopted method can effectively identify real images used for training synthetic images and has great potential for application in privacy protection and synthetic image analysis.

This paper is organized as follows: Section 2 introduces the task and the dataset, Section 3 elaborates on our proposed method, Section 4 presents and discusses the experimental results, and finally, Section 5 summarizes the contributions of this paper and proposes future research directions.

2. The 2025 ImageCLEFmed GAN subtask1

The task introduced in ImageCLEFmed GAN 2025 aims to study whether specific real images are used to train the generative model to generate synthetic biomedical images. Participants are required to annotate each real image in the test set to indicate whether it was used to generate the corresponding synthetic image. Specifically, participants need to annotate each real image as "used" (1) or "not used" (0) to determine whether it participated in the training process of generating the image. This subtask aims to detect "fingerprints" in synthetic images, that is, to identify whether there are real image features in the generated image that can be traced back to the training data. The data of the training set and test set are shown in Table 1.

Table 1
Description of training and testing datasets made available for Identify training data “fingerprints” sub-task.

Train		Test	
Real images	Generated images	Real images	Generated images

100(used)
100(not used)
	5000	500	2000

This task focuses on the potential privacy leakage and data security issues in the process of generating synthetic images. It explores whether the generative model can generate synthetic images that are highly similar to real patient images, which may lead to the leakage of training data. Participants need to analyze the test image dataset and evaluate whether certain real images are used in the training process of the generative model. To this end, the task’s dataset includes real and synthetic images.

3. Methods

3.1. Image Preprocessing and Enhancement

Gaussian filter enhancement is a common image smoothing method that can effectively reduce the image’s noise and retain the image’s edge information. In the experiment, we applied Gaussian blur to smooth the details of the image and enhanced the image details through histogram equalization. This enhancement method helps the model better identify the key information in the image during image processing.

Laplacian operator enhancement is an edge detection method that uses second-order derivatives to identify edge information in an image. In the experiment, we used the Laplacian operator to extract edge regions in the image. This edge information plays an important role in subsequent feature extraction,

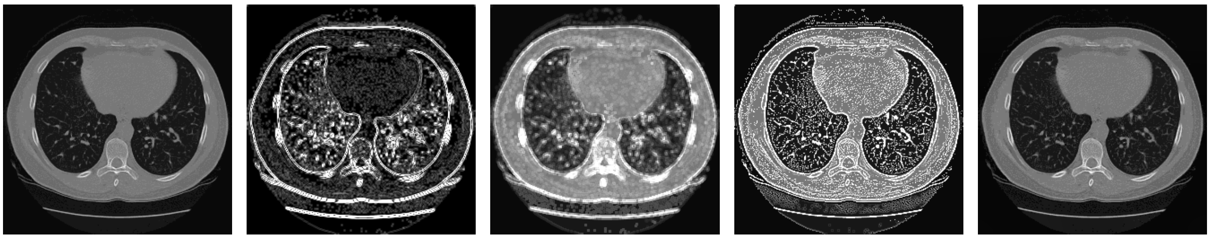

Figure 1: From left to right are the original generated image, Hessian matrix enhanced image, Laplacian enhanced image, Gaussian difference (DoG) and histogram equalization image, and bilateral filtering enhanced image.

helping the model focus on the details in the image, thereby improving the accuracy of similarity calculation.

The Hessian matrix enhancement method captures the local curvature information of the image by calculating the second-order derivative of the image, which is particularly suitable for edge and detail enhancement. We use the Hessian matrix to calculate the edge area of the image and further enhance the details in the image. Through this method, we can enhance the high-frequency information in the image, making the details in the image more prominent, thereby improving the quality of subsequent feature extraction.

Bilateral filtering is a smoothing method that can effectively preserve image edges, especially for images with complex textures. In our experiments, bilateral filtering is used to smooth areas in the image while keeping the edges sharp. By enhancing the details of the image, bilateral filtering improves the image's visual effect and makes the image's key information more obvious.

These four image enhancement methods enhance the image details from different perspectives and processing methods. Gaussian filtering focuses on denoising and preprocessing, the Laplacian operator emphasizes the edge information of the image, the Hessian matrix enhances the structural features of the image through curvature analysis, and bilateral filtering improves the details of the image through edge-preserving smoothing. These enhancement methods cooperate with each other in the feature extraction process, which can effectively improve the quality of the image and help the subsequent feature extraction and similarity calculation to be more accurate. The original generated image and the image after the four enhancement methods are shown in Figure 1. The comparison of image enhancement methods is shown in Table 2.

Table 2
Comparison of Image Enhancement Methods.

Image enhancement methods	Key Parameters	Enhancement effect description
Gaussian filter	kernel size=(5,5)	Smooth the image, remove high-frequency noise, and retain edge information
Laplace operator	Convolution kernel size=3	Extract image edges and enhance image details
Hesse-Matrix	Threshold=0.01*maximum value	Extract image edges and enhance image details
Bilateral Filter	SigmaColor=75,SigmaSpace=75	Extract image edges and enhance image details

3.2. Feature Extraction

Feature extraction is a key step in our task. By extracting effective features from images, we can calculate the similarity between images and perform subsequent classification. In this task, we used convolutional neural networks (CNN) and ResNet50 models to automatically extract image features, taking advantage of the deep learning network's ability to extract deep patterns in images.

Convolutional Neural Network (CNN) is a powerful deep learning model widely used in computer vision tasks, especially image classification, object detection, and image segmentation. In this task, CNN is used to automatically extract features from images.

CNN extracts features from images through multiple layers of convolution and pooling operations. The convolution layer extracts different features from the image, such as edges, corners, textures, etc., by performing convolution operations with local areas of the image. The pooling layer reduces the spatial size of the image and retains important feature information by performing dimensionality reduction on the convolution results. During the feature extraction process, CNN can gradually extract more advanced features through convolution and pooling operations at each layer, thereby capturing complex patterns and details in the image. We use the optimized CNN model to extract features from the image and use these features for subsequent similarity calculations and classification. The model consists of multiple convolutional, activation, and pooling layers. Through optimized structure and training methods, it can efficiently extract low-level and high-level features of the image.

The advantage of CNN is that it can automatically learn features. CNN can automatically learn image features through the back-propagation algorithm without manually designing feature extractors. It can also connect locally and share weights. The convolution operation of CNN greatly reduces the number of parameters by locally connecting and sharing weights, allowing the model to better handle complex images. It is also translation invariant. The convolution operation is invariant to the translation of the image, which means that CNN can recognize the same objects in the image regardless of their position.

ResNet50 is a deep convolutional neural network, which is designed based on the concept of "Residual Learning" and uses residual blocks to solve the gradient vanishing problem in deep networks. Due to its depth and effective structure, ResNet50 performs very well in image classification and feature extraction tasks, especially when processing large-scale image datasets.

ResNet50 consists of a 50-layer deep convolutional neural network, which mainly uses residual blocks to enhance the network's expressiveness. Each residual block contains several convolutional layers but adds "skip connections" that allow signals to jump over certain layers directly, thereby avoiding the gradient vanishing problem that may occur in traditional deep networks. Through this residual learning mechanism, ResNet50 can train deeper networks and capture more complex features. In the feature extraction process, ResNet50 gradually learns the complex patterns in the image through the previous multi-layer convolution operations and finally outputs a set of high-dimensional features through the fully connected layer, which can effectively represent the visual information of the image. In our study, we used a custom ResNet50 model, which was trained on the dataset and can extract high-level features of the image.

The image features extracted by CNN and ResNet50 are all high-dimensional feature vectors, which play an important role in similarity calculation. We use these high-dimensional feature vectors as input for subsequent similarity calculations and judge their similarity by comparing the feature vector differences between different images.

3.3. Similarity Calculation Methods

In the 2025 ImageCLEFmed GAN task, the choice of similarity calculation method is crucial to evaluate the similarity between generated images and real images. We used several common similarity calculation methods, including cosine similarity, structural similarity index (SSIM), and Jaccard similarity. These methods are widely used in image processing, machine learning, information retrieval, and other fields. They can effectively measure the similarity between image features and provide strong support for subsequent classification tasks.

Cosine similarity is commonly used when calculating image similarity, particularly in content-based image retrieval (CBIR) systems. Cosine similarity assesses the similarity between two vectors by measuring the cosine of the angle between them. The core idea is that the closer the directions of the two vectors, the more similar they are, regardless of their magnitudes. Calculating cosine similarity involves converting each image into a vector form. This typically entails flattening the pixel values of the image or features extracted from the image (such as color histograms, texture descriptors, shape features, etc.) into a one-dimensional vector. The cosine similarity value ranges from -1 to 1, where 1 indicates identical directions (very similar), 0 indicates orthogonality (no similarity), and -1 indicates completely opposite directions. Cosine similarity focuses on directional similarity, ignoring magnitude. In some cases, two images might be very similar in terms of certain feature ratios, but the absolute differences in actual pixel values could be significant.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Where $\mathbf{A}$ and $\mathbf{B}$ are two vectors. $\mathbf{A} \cdot \mathbf{B}$ represents the dot product of vectors $\mathbf{A}$ and $\mathbf{B}$. $\|\mathbf{A}\|$ and $\|\mathbf{B}\|$ represent the magnitudes of vectors $\mathbf{A}$ and $\mathbf{B}$.

Structural Similarity (SSIM) is a more intuitive and effective method for calculating image similarity. SSIM considers images' luminance, contrast, and structural information, allowing it to more accurately reflect the human visual system's perception of image quality. SSIM first calculates the luminance difference between two images. The luminance comparison is achieved by calculating the mean values of the images, which reflects the overall brightness levels of the images. Next, SSIM calculates the contrast difference. The contrast comparison is achieved by calculating the standard deviation of the images; the greater the standard deviation, the higher the image contrast. Finally, SSIM compares the structural information of the two images. This step is achieved by calculating the covariance of the images. Covariance reflects the linear relationship between the images' pixels, capturing the images' structural characteristics.

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_{\mathbf{x}} \cdot \mu_{\mathbf{y}} + C_1)(2\sigma_{\mathbf{xy}} + C_2)}{({\mu_{\mathbf{x}}}^2 + {\mu_{\mathbf{y}}}^2 + C_1)({\sigma_{\mathbf{x}}}^2 + {\sigma_{\mathbf{y}}}^2 + C_2)}$$

Where x and y are corresponding blocks of the two images. μ_x and μ_y are the mean values of image blocks x and y. σ_x and σ_y are the standard deviations of image blocks x and y. σ_{xy} is the covariance of image blocks x and y.

Jaccard Similarity is another commonly used method for measuring image similarity. Jaccard Similarity is a common method for measuring the similarity between two sets, especially for evaluating the overlap between objects or regions in an image. In image processing, Jaccard Similarity is often used to compare the similarity between two binary images, that is, to quantify their similarity by comparing the image's feature regions (such as edges, targets, etc.). In image processing, Jaccard Similarity is often used to compare the features of an image (such as edges, textures, colors, etc.) or the overlapping parts of the pixel level of a binary image. The value of Jaccard Similarity is between 0 and 1, and the larger the value, the more similar the two images are. For this task, we use Jaccard Similarity to evaluate the similarity between synthetic and real images. We first convert the real and synthetic images into high-dimensional feature vectors through a feature extraction model (ResNet50). Then, we obtain the binary representation of the salient regions in the image by binarizing these feature vectors. We set a threshold. When the Jaccard similarity exceeds the threshold, the real image is considered to have participated in the generation process. Otherwise, the image is considered to have not been used. This way, we can label the real image as "used" or "unused". The advantage of Jaccard similarity is that it can intuitively reflect the degree of overlap of image features, especially for images or features after binarization. Therefore, when processing synthetic images, Jaccard similarity can effectively capture the common parts between images and help us identify similar areas between generated and real images. In addition, Jaccard similarity is not affected by the scale or specific size of the image, so it has a certain robustness when processing images of different sizes.

$$\mathrm{Jaccard}(x, y) = \frac{|x \cap y|}{|x \cup y|}$$

where x and y represent the sets of feature regions (such as edges, textures, or salient regions) of the two images, $|x \cap y|$ is the number of common elements (intersection) between the sets, and $|x \cup y|$ is the total number of elements in the union of the two sets.

4. Experiments

4.1. Evaluation Metrics

We use the following evaluation indicators to evaluate the model's performance: Kappa coefficient, accuracy, precision, recall, and F1 score. Since the Kappa coefficient can effectively measure the consistency between the model prediction and the true label, especially in the case of class imbalance, we use the Kappa coefficient as the main evaluation indicator. At the same time, the F1 score is used to provide a balanced evaluation of precision and recall. The definitions of these metrics are as follows:

$$\text{Kappa} = \frac{P_o - P_e}{1 - P_e} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

4.2. Experimental Results

We used two different models (CNN and ResNet50) and three similarity calculation methods (cosine similarity, SSIM, and Jaccard similarity) to conduct experiments, and the experimental results of different models and similarity calculation methods are shown in Table 3. We pay special attention to the Kappa coefficient because it can comprehensively evaluate the classification consistency of the model.

Table 3
The results were obtained using CNN and ResNet50 models and three similarity calculation methods: cosine similarity, SSIM, and Jaccard.

Model	Similarity calculation method	Accuracy	Precision	Recall	F1-Score	Kappa
CNN	Cosine similarity	0.466	0.482	0.92	0.633	-0.068
CNN	SSIM	0.482	0.490	0.82	0.614	-0.032
ResNet50	Jaccard	0.492	0.490	0.412	0.448	-0.016

Our team submitted 5 results, with the best kappa value of -0.016 and the best F1 score of 0.633.The results show that the Kappa coefficient of the CNN model is low, indicating that the model's predictions are less consistent with the actual labels. When using SSIM similarity, although the Kappa coefficient has improved, it still shows that the model has greater uncertainty when processing data. This may be related to the dataset's deviation or the model's overfitting. The ResNet50 model has the smallest Kappa coefficient in all experiments, especially under the Jaccard similarity calculation. The Kappa value is close to 0, indicating that its classification results are less consistent with the actual labels. Cosine similarity performs well on the CNN model. Although the Kappa coefficient is low, it performs well regarding recall, indicating that this method can effectively detect "used" images. SSIM similarity is better than cosine similarity, especially on the CNN model, which can balance precision and recall and obtain more stable results. SSIM considers the image's structural information and may be more suitable for such tasks. Jaccard similarity has the worst effect on ResNet50, with an F1 score of 0.448 and a negative Kappa coefficient, indicating that this method may not be suitable for such tasks.

5. Conclusions

This paper proposes a method that combines image enhancement with deep learning models to detect the "fingerprint" of training data in synthetic biomedical images. We evaluate the similarity between real and synthetic images using cosine similarity, SSIM, and Jaccard similarity. Experimental results show that the CNN model performs best when combined with SSIM similarity calculation. Although the performance of the ResNet50 model is weak, it still has the potential for optimization. Future research can optimize the similarity calculation method, further improve the feature extraction ability of the deep learning model, and enhance the performance and robustness of the model. This study provides an effective method for the privacy protection of synthetic images and has good application prospects.

Declaration on Generative AI

During the preparation of this work Chat-GPT-4o and Grammarly were used to check grammar and spelling. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] M. Tsuneki, Deep learning models in medical image analysis, Journal of Oral Biosciences 64 (2022) 312–320.

[2] D. Shen, G. Wu, H.-I. Suk, Deep learning in medical image analysis, Annual review of biomedical engineering 19 (2017) 221–248.

[3] S. M. Bellovin, P. K. Dutta, N. Reitinger, Privacy and synthetic datasets, Stan. Tech. L. Rev. 22 (2019) 1.

[4] X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, A. V. Vasilakos, Privacy and security issues in deep learning: A survey, IEEE Access 9 (2020) 4566–4593.

[5] Z. Kuang, Z. Guo, J. Fang, J. Yu, N. Babaguchi, J. Fan, Unnoticeable synthetic face replacement for image privacy protection, Neurocomputing 457 (2021) 322–333.

[6] A.-G. Andrei, M. G. Constantin, M. Dogariu, A. Radzhabov, L.-D. Ştefan, Y. Prokopchuk, V. Kovalev, H. M"uller, B. Ionescu, Overview of ImageCLEFmedical 2025 – GANs Task, in: CLEF2025 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Madrid, Spain, 2025.

[7] B. Ionescu, H. M"uller, D.-C. Stanciu, A.-G. Andrei, A. Radzhabov, Y. Prokopchuk, Ştefan, Liviu-Daniel, M.-G. Constantin, M. Dogariu, V. Kovalev, H. Damm, J. R"uckert, A. Ben Abacha, A. Garc'ia Seco de Herrera, C. M. Friedrich, L. Bloch, R. Br"ungel, A. Idrissi-Yaghir, H. Sch"afer, 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.

[8] G. Singh, A. Mittal, et al., Various image enhancement techniques-a critical review, International Journal of Innovation and Scientific Research 10 (2014) 267–274.

[9] D. Nandan, J. Kanungo, A. Mahajan, An error-efficient gaussian filter for image processing by using the expanded operand decomposition logarithm multiplication, Journal of ambient intelligence and humanized computing 15 (2024) 1045–1052.

[10] J. Lavín-Delgado, J. Solís-Pérez, J. Gómez-Aguilar, J. Razo-Hernández, S. Etemad, S. Rezapour, An improved object detection algorithm based on the hessian matrix and conformable derivative, Circuits, Systems, and Signal Processing 43 (2024) 4991–5047.

[11] P. Ma, H. Yuan, Y. Chen, H. Chen, G. Weng, Y. Liu, A laplace operator-based active contour model with improved image edge detection performance, Digital Signal Processing 151 (2024) 104550.

[12] N. S. Awarayi, F. Twum, J. B. Hayfron-Acquah, K. Owusu-Agyemang, A bilateral filtering-based image enhancement for alzheimer disease classification using cnn, Plos one 19 (2024) e0302358.