Fingerprint Identification of Generative Models Using a MultiFormer Ensemble Approach

Notebook for ImageCLEF Lab at CLEF 2024

Md. Ismail Siddiqi Emon[1,†], Mahmudul Hoque[1], Md Rakibul Hasan[1], Fahmi Khalifa[2,†] and Md Mahmudur Rahman[1,*]

[1]Department of Computer Science, SCMNS School, Morgan State University, Baltimore, Maryland 21251, USA
[2]Electrical & Computer Engineering Dept., School of Engineering, Morgan State University, Baltimore, Maryland 21251, USA

Abstract

In the ever-changing realm of medical image processing, ImageCLEF brought a new dimension with the Identifying GAN Fingerprint task, catering to the advancement of visual media analysis. This year, the author presented the task of detecting training image fingerprints to control the quality of synthetic images for the second time (as task 1) and introduced the task of detecting generative model fingerprints for the first time (as task 2). Both tasks are aimed at discerning these fingerprints from images, on both real training images and the generative models. The dataset utilized encompassed 3D CT images of lung tuberculosis patients, with the development dataset featuring a mix of real and generated images, and the test dataset. Our team 'CSMorgan' contributed several approaches, leveraging multiformer (combined feature extracted using BLIP2 and DINOv2) networks, additive and mode thresholding techniques, and late fusion methodologies, bolstered by morphological operations. In Task 1, our optimal performance was attained through a late fusion-based reranking strategy, achieving an F1 score of 0.51, while the additive average thresholding approach closely followed with a score of 0.504. In Task 2, our multiformer model garnered an impressive Adjusted Rand Index (ARI) score of 0.90, and a fine-tuned variant of the multiformer yielded a score of 0.8137. These outcomes underscore the efficacy of the multiformer-based approach in accurately discerning both real image and generative model fingerprints.

Keywords

GAN Fingerprint, Multiformer, BLIP2, Generative Model Fingerprint, Training Data Fingerprint, DINOv2, Late Fusion, Thresholding, Reranking, ARI Scoring, CT image denoising

1. Introduction

In 2004 ImageCLEF [1] was established for medical imaging which has pioneered advancements in this field. After it advanced to its second edition [2], it marked a significant evolution by the inclusion of medical Generative Adversarial Networks (GANs) tasks. And, the first iteration of its kind came in 2023. This task aimed to explore a specific hypothesis which was that the GANs might embed "fingerprints" of real images within the synthetic medical images they generate. Confirming this hypothesis could have significant implications. It might lead to a reconsideration of the copyright status of synthetic images. This would challenge the conventional view that synthetic images are entirely artificial. In recent years, there has been a substantial surge in the application of GANs and diffusion models within the medical domain [3, 4, 5]. These sophisticated architectures are capable of generating synthetic images and facilitating their translation across different modalities. This burgeoning utilization underscores the transformative potential of GANs and diffusion models in enhancing medical imaging and diagnostics. Medical imaging professionals have been exploring various applications of GANs in medical image

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
*Corresponding author.
†These authors contributed equally.
mdemo1@morgan.edu (Md. I. S. Emon); mahoq1@morgan.edu (M. Hoque); mdhas1@morgan.edu (M. R. Hasan); fahmi.khalifa@morgan.edu (F. Khalifa); md.rahman@morgan.edu (M. M. Rahman)
https://github.com/ismailEmonFu (Md. I. S. Emon); https://github.com/HoqueMahmudul (M. Hoque); https://github.com/Hasan-MdRakibul (M. R. Hasan); https://mdrahmanlab.com/ (M. M. Rahman)
0000-0003-0595-229X (Md. I. S. Emon); 0009-0006-5532-4135 (M. Hoque); 0000-0002-6179-2238 (M. R. Hasan); https://orcid.org/0000-0003-3318-2851 (F. Khalifa)

analysis, such as creating artificial medical images and distinguishing between real and fake images. They have developed effective architectures like "Attention GAN" [6] and "ABC GAN" [7] to produce lifelike medical images, which assist with various tasks, including training AI models and protecting patient privacy. However, despite these advancements [8, 9], a major challenge remains: differentiating between real and synthetic medical images. This is an area where scientists continue to focus their efforts. Generative models [10, 11, 12, 13], a recent AI innovation, have driven significant advancements across multiple domains [14, 15, 16], including generative medical imaging [17, 18, 19], such as authors from [17] shows that synthesizing high-resolution images of skin lesions with Generative Adversarial Networks (GANs) can address the lack of labeled data and skewed class distributions in skin image analysis. Using progressive growing, they produce realistic dermoscopic images that are difficult for expert dermatologists to distinguish from real ones, outperforming other GAN architectures like DCGAN and LAPGAN. Synthetic biomedical images serve vital roles in research, healthcare professional training [20], and patient care enhancement from anisotropic diffusion [21] to AnoGAN, an unsupervised deep convolutional GAN to identify anomalies in imaging data for disease markers, demonstrated by accurately detecting anomalies in retinal OCT images [22]. Additionally recent advancement demonstrates that [23] discriminating between malignant and benign lung nodules remains challenging, necessitating CAD systems to assist radiologists. Using unsupervised learning with Deep Convolutional-Generative Adversarial Networks (DC-GANs), they aim to generate realistic lung nodule samples, hypothesizing that difficult-to-differentiate imaging features will be highly discriminative, thereby improving diagnostic accuracy, training radiologists, and generating realistic samples for deep network training.They address challenges such as data scarcity, cost, and ethical considerations associated with real patient data acquisition.

The 2024 ImageCLEFmedical GANs Task provides a forum to investigate the influence of GANs on the generation of artificial biomedical images, facilitating the examination of the potential advantages and ethical concerns of their application. This includes two basic objectives: (1) how to identify fingerprints of training data in synthetic biomedical images (inspection), and (2) how to find fingerprints of different generative models on images that they are designed to generate (differentiation). This essentially allows researchers to compare models and highlight the characteristics, patterns, or features present in synthetic images that would separate the models other than that they might not be alike. This dataset includes axial slices of 3D CT images of around 8000 lung tuberculosis patients and acts as a great resource for research.

Task 1 aims to identify the real images that GANs' images were generated from, so as to address privacy and security concerns associated with the use of artificial images [18]. The datasets were provided by the organizers, with a development set divided into marked artificial and real images with training while mentioning no information on the percentage of the used and unused images from this set is disclosed. This extensive dataset is useful to rigorously test the hypotheses and push the community forward with discoveries in biomedical image synthesis.

Task 2 is to detect the fingerprints of generative models in GANs-generated images. The authors of ImageCLEFmed GANs argue that one way to understand this behavior is to envision that each AI model imparts its own distinct "signature" on the images it generates. So, here, our aim is to reveal these hidden signatures to identify what makes each model unique. It's like reading differently styled handwriting from different authors but in the AI area. We do not simply aim to differentiate models but rather to gain a deeper understanding of them, by examining the hidden patterns and subtleties contained within the synthetic images.

2. Datasets

In the second edition of the ImageCLEFmed GANs challenge, the organizers presented two tasks: "Identify training data fingerprints" and "Detect generative models' fingerprints."

For the *first task*, the hypothesis is that when images are generated using diffusion models, the original real images leave specific fingerprints in the generated models. If this hypothesis is true, it

could impose additional restrictions on publishing or sharing such images publicly, as they would be as sensitive as the original images. Conversely, if the hypothesis is false, it could lead to a vast dataset of artificially generated images using diffusion models, potentially revolutionizing the medical imaging field.

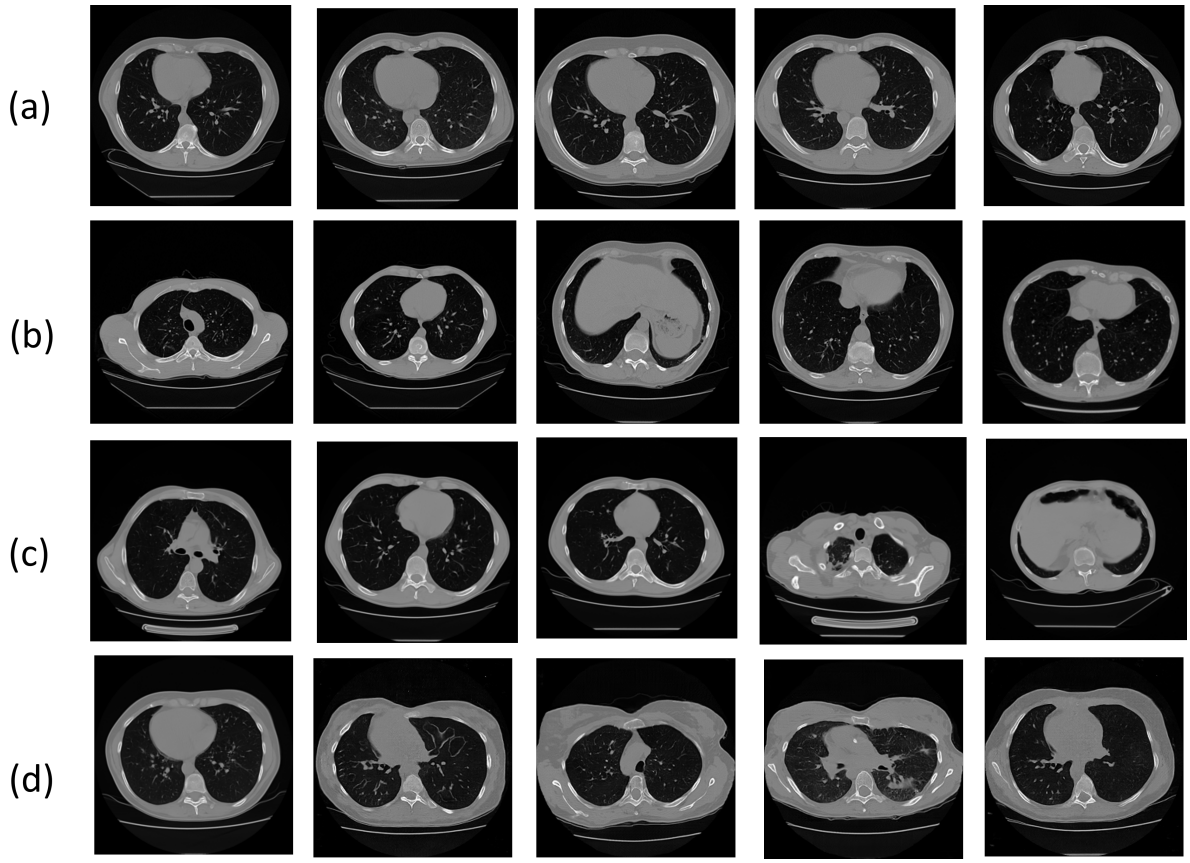

Figure 1: Examples of CT images: (a) from the dataset for identifying training data fingerprints; (b) not from the dataset for identifying training data fingerprints; (c) generated from the dataset for identifying training data fingerprints, and (d) generated from the dataset for detecting generative model fingerprints.

The *second task*, although different in its subjective nature, shares a similar objective: *identifying generative model fingerprints in images produced by various diffusion models*. The challenge is to determine whether these models imprint unique fingerprints in the generated images. The organizers did not disclose the number of diffusion models used, but the approach remains consistent regardless of the quantity.

The author of the ImageCLEFmed GANs task provided both a development set and a training set. For *Task 1*, there were two datasets consisting of axial slices of 3D CT images from approximately 8,000 lung tuberculosis patients. The artificial slice images, sized at 256×256 pixels, were generated using various undisclosed generative adversarial networks and diffusion neural networks. Over 12,000 generative images were included, and the test set contained a total of 8,000 images. For *Task 2*, a dataset comprising 3,000 generated image files was provided. Figure 1 depicts examples of images from the original dataset.

3. Proposed Methodology

Here we are going to explain the details of the approaches utilized in our submission for both tasks: "Identify training data fingerprints" and "Detect generative models' fingerprints". For the task of identifying training data fingerprints, we first performed morphological operations to reduce noise in the CT images. This preprocessing step was crucial for improving the quality of synthetic medical

images generated by GANs. Subsequently, we implemented BLIP and DINOv2 as image signature generators. As illustrated in Figure 2, these morphological operations helped control the quality of the images. After preprocessing, we conducted individual feature rankings for each model and then concatenated the feature rankings from both models. We also performed dimensionality reduction on the concatenated features to enhance the ranking process. Finally, we applied late fusion to combine the results from the previous steps, optimizing the identification of training data fingerprints.

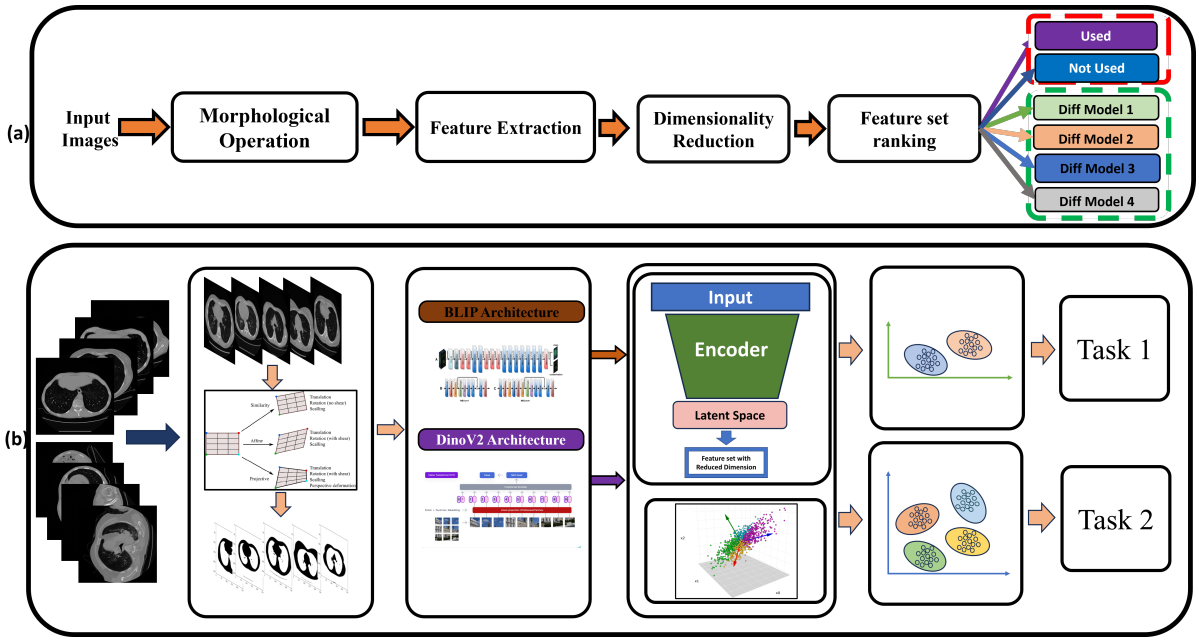

Figure 2: A block diagram of the sequential processing steps (a) and detailed schematic of the the developed system for identifying training data and detecting generative models' fingerprints (b). The process in (b) begins with morphological operations to reduce noise in CT images, followed by using BLIP and DINOv2 for image signature generation. After concatenating and reducing the dimensionality of features using an encoder and PCA, late fusion is applied to optimize the identification of training data.

3.1. Morphological operations

During image processing image opening is widely utilized for effective morphological operation. It primarily aimed to remove small objects from 3D axial CT images. But at the same time preserves the size and shape of larger structures. Which is an effective noise-releasing mechanism. As illustrated in Figure 3, electronic noise in CT images often originates from the combination of the detector system and the reconstruction kernel, with sharper kernels typically resulting in noisier images. According to [24], this noise is a consequence of efforts to enhance image quality without increasing the radiation dose. Our hypothesis is that applying image opening will eliminate small, noisy, and irrelevant details, thereby potentially enhancing the efficiency of fingerprint detection in medical images. By preserving key features, image opening maintains the integrity of essential image details, ensuring that the important structures remain intact while the noise is reduced.

Image opening consists of two main steps erosion (Eq. 1) followed by dilation (Eq. 2). Image opening (3) can be represented as:

$$A \ominus B = \{z \in E | B_z \subseteq A\} \tag{1}$$

$$A \oplus B = \bigcup_{b \in B} A_b \tag{2}$$

$$A \circ B = (A \ominus B) \oplus B \tag{3}$$

Here in equation (1) above image erosion has been performed, where A and B are input image and structuring elements respectively. And B_z is the translation of B by z. Then in equation (2), image

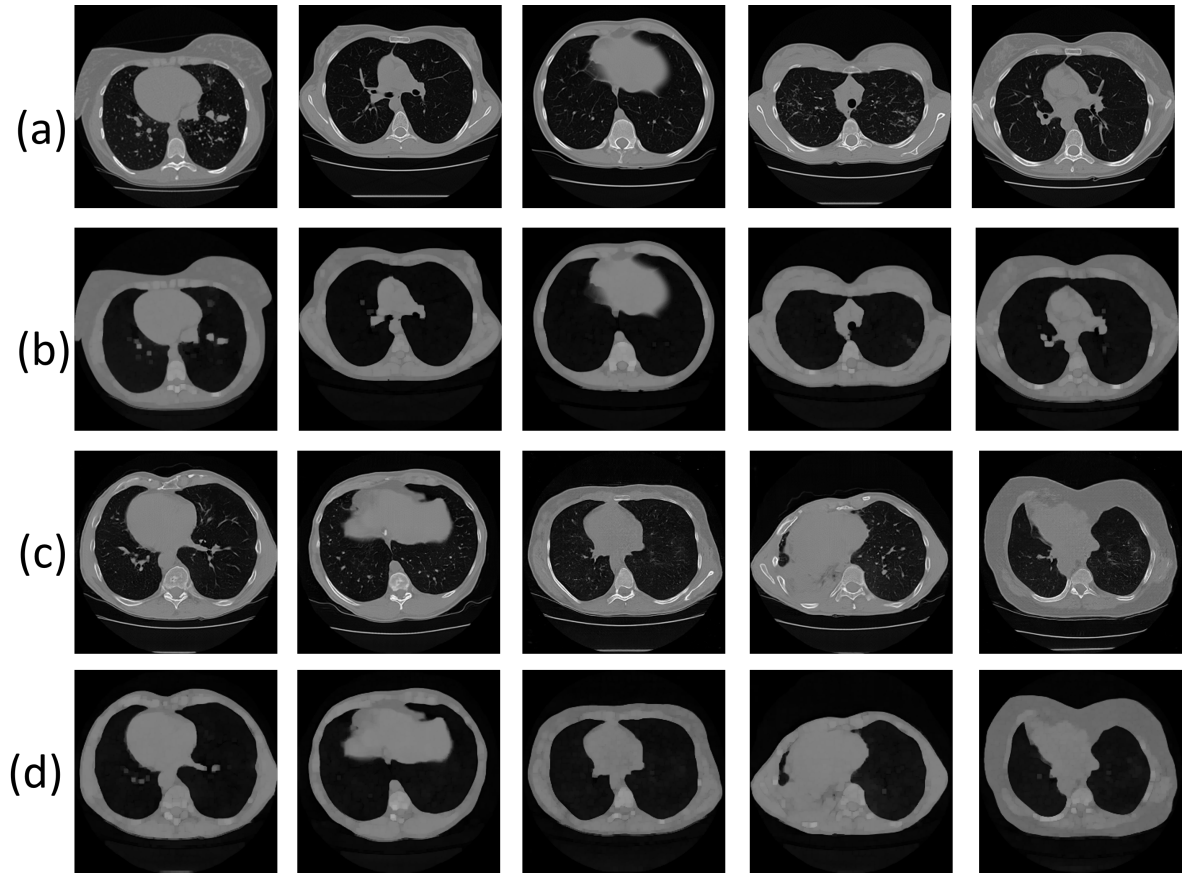

Figure 3: (a) Noisy CT images sampled from the ImageCLEF GAN dataset for task training image fingerprint identification; (b) Processed images after applying the image opening operation, which involves an erosion step followed by dilation. (c) Noisy CT images sampled from the ImageCLEF GAN dataset for the task of generative model fingerprint identification; (d) Processed images after applying the image opening operation for task 2, which also involves an erosion step followed by dilation.

dilation is performed where A_z is the translation of A by z. In the equation (3), $\bigcirc$ is the image opening operation.

3.2. Multiformer

To build a multiformer we have chosen two foundation models which are BLIP and DINOv2 as backbone architecture, see Fig. 4, for which details are given below.

3.2.1. BLIP Architecture

Bootstrapping Language-Image Pre-training 2 (BLIP2) [25, 26] utilize a Visual Transformer (ViT) [27] for image encoding, dividing images into patches and encoding them into a sequence of embeddings with a [CLS] token representing the global image feature. This method is computationally efficient compared to traditional object detectors. BLIP is a multimodal Mixture of Encoder-Decoder (MED), operates in three modes: unimodal encoder [28] (similar to BERT for text), image-grounded text encoder (incorporates visual information via cross-attention layers), and image-grounded text decoder (uses causal self-attention layers for text generation). During pre-training, we jointly optimize three objectives: two for understanding and one for generation. Each image-text pair undergoes one forward pass through the ViT and three through the text transformer for different tasks. The Image-Text Contrastive Loss (ITC) inspired from [29, 30] to align visual and textual feature spaces, improving vision-language understanding by encouraging positive image-text pairs to have similar representations.

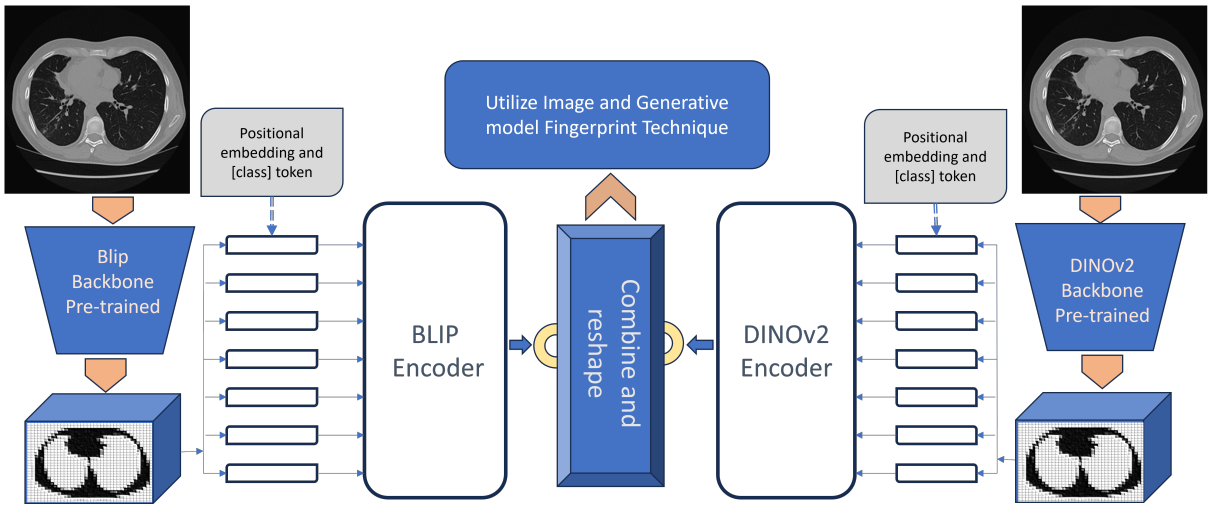

Figure 4: A thematic block diagram of multiformer leveraging the powerful architecture of BLIP2 and DINOv2.

The Image-Text Matching Loss (ITM) [30] learns fine-grained multimodal representations through a binary classification task to determine whether image-text pairs are matched, utilizing a hard negative mining strategy. The Language Modeling Loss (LM) trains the model to generate textual descriptions from images using cross-entropy loss, enhancing the model's capability to convert visual information into coherent captions. To maximize efficiency and leverage multi-task learning, the text encoder and decoder share parameters except for the self-attention layers, which capture the differences between encoding and decoding tasks. This shared architecture benefits from improved training efficiency and effective multi-task learning.

The BLIP2 feature extractor is a multi-modal model designed to extract and integrate features from both images and text. It begins with patch embedding for images, converting each image X into patches P using a convolutional layer, $P = \text{Conv}(X)$. These patches are then processed through a transformer encoder, $F_v = \text{Transformer}_v(P)$, to capture visual relationships. Image data I is tokenized and embedded into high-dimensional space, $E_t = \text{Embedding}(I)$, and passed through another transformer encoder, $F_t = \text{Transformer}_t(E_t)$. The cross-modal attention mechanisms enhance its performance to state of the art, $F_{vt} = \text{CrossAttention}(F_v, F_t)$, are used to align and integrate visual features. The model is pre-trained on large datasets with paired images and text to learn these representations. The pre-trained model can be fine-tuned for downstream tasks to improve performance on specific datasets. The BLIP2 feature extractor thus provides robust, high-level features suitable for tasks such as image classification, object detection, and text-image matching.

3.2.2. DINOv2 Architecture

DINOv2 [31] is an enhanced version of DINO [32], integrating various improvements and using a larger, more diverse dataset to accelerate and stabilize training at scale. It utilizes the LVD-142M curated dataset, which includes data from sources like ImageNet [33], Google Landmarks [34], Mapillary SLS [35], and Food-101 [36]. The dataset is de-duplicated using FAISS [37] batch searches and embeddings. Training combines DINO and iBOT losses with SwAV centering, involving a learnable student and an EMA teacher. Key techniques include multi-crop cross-entropy for global image representation, patch-level masking, and separate weights for image and patch objectives. The teacher's softmax-centering is replaced by Sinkhorn-Knopp batch normalization, and the KoLeo regularizer ensures batch uniformity. High-resolution images are used towards the end of pre-training. DINOv2's implementation features FlashAttention [38, 39] for efficiency, nested tensors from xFormers, stochastic depth, and mixed-precision PyTorch FSDP. Distillation [40, 41] into smaller models from a larger teacher model is also included. Ablation studies cover model selection, data curation strategies, model scaling, and

loss objectives. DINOv2 achieves results comparable to weakly supervised text models like EVA-CLIP on ImageNet and performs better than SSL methods like Mugs, EsViT, and iBOT. It shows strong performance in domain generalization, image and video classification, instance recognition, and semantic segmentation. For depth estimation, DINOv2 uses a linear layer on frozen tokens and experiments with concatenation of ViT [27] layers/blocks and regression over a DPT decoder, achieving superior results on datasets like NYUd, KITTI, and SUN-RGBd. Qualitative results demonstrate effective semantic matching and foreground extraction using PCA on patch features.

We utilized the DINO Vision Transformer because it is a sophisticated image processing model capable of handling and analyzing images through a systematic series of steps. Below is a description of how its architecture operates. Initially, the input image is divided into smaller patches, which are then embedded into a higher-dimensional space using a convolutional layer. This embedding captures the local features of each patch. These embedded patches pass through multiple nested tensor blocks, each consisting of several components: layer normalization to standardize inputs, memory-efficient attention mechanisms to focus on different parts of the image, and multi-layer perceptrons for feature refinement. Each block also includes layer scaling and dropout layers to improve training stability and prevent overfitting. The model incorporates 12 of these blocks, creating a deep network capable of learning complex image representations. After processing through all blocks, a final normalization layer is applied, followed by an identity layer that prepares the features for subsequent tasks. This architecture allows the DINO ViT to effectively learn and process detailed image features, making it suitable for various image processing applications.

3.3. Autoencoder for feature reduction

Initially, we employed raw features for clustering to assign labels to the test set. Subsequently, we integrated robust features from both the BLIP and DINOv2 models. To enhance the clustering results, we implemented an Autoencoder to encode this extensive feature set into a more meaningful and reduced representation. This dimensionality reduction facilitated improved clustering performance. Autoencoders (AEs) have been quite popularised in AI research over recent years and consequently, there have been many studies and advancements made in this subject, There are mainly two types, they are Fundamental AEs And Variants. The simplest ones, as used in our working example, one can find, are auto-associative neural networks, connected to a multi-layer perceptron, where input is being reconstructed. The encoders consist of an encoder that compresses an input vector with recognition weights into a code vector and a decoder that decompresses the input vector from a code vector with generative weights. This structure allows each layer of a deep network to be trained separately using the basic AE as a building block. The encoder activation function (sf) may be, for example, sigmoid or hyperbolic tangent, a weight matrix (W) is a bias vector (b), and x Wine are used in the computation of the hidden representation of an input vector x with a $y = f_\Theta(x) = sf(Wx + b)$, where W is a weight matrix, b is a bias vector, and sf is the encoder activation function (e.g., sigmoid or hyperbolic tangent). The hidden representation y is then decoded back to a reconstruction vector z using $z = g_\Theta(y) = sg(W'y + b')$, where sg is the activation function of the decoder. The aim is to make the reconstruction(z) as close to the input(x) as possible. To simplify training, the weight matrix W' is often constrained to be the transpose of W (tied weights), reducing the number of free parameters. This mapping process ensures that each input is transformed into a hidden representation and then reconstructed, enabling effective learning and data representation.

4. Experimental Result Analysis

4.1. System Specification and Parameter Settings

The proposed model is implemented on a Google Cloud Vertex AI instance, utilizing a NVIDIA V100 Tensor Core GPU. This GPU boasts 5120 CUDA cores, 640 Tensor cores, up to 32 GB of HBM2 memory, and a memory bandwidth reaching up to 900 GB/s. These specifications allow for highly efficient

processing and accelerated computation, which are essential for handling the complex tasks and large datasets involved in our model's execution.

For the BLIP architecture, the parameters used make it better in feature extraction. The input image size was normalized at 256×256 pixels, followed by data augmentations of random cropping, horizontal flipping with probability $p = 0.5$, rotation within $\pm 20^{\circ}$, and color jittering with brightness adjustment factors $[0.75, 1.25]$. Feature Extraction used the pre-trained ViT large model as the backbone. The value chosen for the learning rate η was 0.0005 and a batch size B of 16 was employed. The AdamW optimizer where weight decay factor was added was used for loss function minimization $L(\theta)$ Normalization layers were used for scaling the features within the same scale of any particular signal for the dataset.

For Dinov2 architecture, a number of key parameters were defined to maximally optimize their performance. The model was pre-trained on a large companys own well curated set of images, making it a great base for later fine-tuning. The size of the input image was kept uniform during training, 224×224 pixels, where several data augmentations like center cropping, random affine transformations, resizing, and normalizing, were performed. We used a learning rate η of 0.001 and batch size (B) of 32. Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ was used to minimize the loss function $L(\theta)$ where θ denotes the model parameters. We applied a dropout with a probability of p = 0.5 and layer normalization to prevent overfitting and maintain training stability.

For the Dinov2, BLIP we then tried to tune these parameters carefully to get as high performance as possible to identify generative model fingerprintfs and test image fingerprints. I executed these procedures while training the item co-occurrence model with data augmentation, learning rates, and optimization technique and achieved high quality and stable item embeddings.

4.2. Identify Training Data Fingerprints Experiments

For identifying training data fingerprints, we first reduced noise in the CT images through morphological operations. We then used BLIP and DINOv2 as image signature generators. As shown in Figure 2, these steps improved the quality of synthetic medical images generated by GANs. We ranked features individually and after concatenation, performed dimensionality reduction, and used late fusion to refine our fingerprint identification results.

For submission 1, we implemented the "additive mode thresholding" technique, which considers local variations in image intensity to enhance image processing. First, we reduced the dimension of the feature vector using Principal Component Analysis (PCA). We then combined all the features and weighted them by the total. The mode of this final weighted result was used as the threshold. For the test images, we applied a similar weighting approach: if the weighted value was less than the mode, the image was tagged as not used; otherwise, it was tagged as used. This method allowed us to account for local intensity variations, improving the accuracy of our thresholding.

In submission 2, which we titled "additive average thresholding," we took a different approach. We calculated the final result for each subject and then averaged these results across all subjects. This average became the threshold value for classification. By using the average, we aimed to create a more generalized threshold that could effectively classify the images based on the overall distribution of the data.

For submission 3, we used an encoder model to handle the extensive feature set generated by the backbone models. The encoder compressed this concatenated feature set, reducing its dimensionality. With the reduced feature set, we applied both mode and mean thresholding techniques. This dual approach allowed us to leverage the strengths of both thresholding methods, providing a robust classification mechanism.

In submission 5, we employed a late fusion strategy to combine the decisions from the previous four methods. Late fusion involves aggregating the results at the decision level rather than at the feature level. We used majority voting to finalize the classification, ensuring that the combined decisions of the different methods provided a more accurate and reliable result. This ensemble approach helped to mitigate the weaknesses of individual methods and improved the overall performance of our classification system.

For the final submission 6, we performed reranking using the Agglomerative Clustering algorithm. This algorithm conducts hierarchical clustering with a bottom-up approach, allowing us to specify parameters such as the number of clusters, distance metric, and linkage criterion. The reranking was based on decisions from the previous submissions.

4.3. Identify Generative Model Fingerprints Experiemnts

To identify generative models' fingerprints, we initially reduced noise in the CT images using morphological operations. We then employed the pre-trained BLIP2 and DINOv2 architectures for feature extraction. As illustrated in Figure 2, our objective was to accurately label each subject with the corresponding model number, determining which generative or diffusion model produced each image.

For submissions 1 and 2, we utilized a combination of feature sets from BLIP and DINOv2 titled 'multiformer' architecture with different augmentation techniques. In submission 1, we applied center cropping and random affine transformations, along with resizing, normalizing, and other standard setups. In submission 2, we used random cropping, random horizontal flipping, random rotation, and color jittering. These augmentations introduced variations in size, orientation, and brightness to the training dataset, enhancing the model's robustness and accuracy. We then used k-means and agglomerative clustering to assign labels to each subject.

Submissions 3 and 4 followed a similar feature extraction method, but we applied PCA and autoencoder was applied for combined feature dimensionality reduction, respectively. The same clustering algorithms were then used for label assignment.

In submissions 5 and 6, we leveraged solely the BLIP architecture. Submission 5 used a BLIP base model for feature extraction, while submission 6 utilized the BLIP pre-trained ViT large model. The normalized feature sets were subsequently fed into clustering algorithms for labeling.

For submissions 7 and 8, we performed ensemble voting and reranking based on the decisions from previous submissions. Ensemble voting combined results at the decision level rather than the feature level, employing majority voting to determine the final classification, ensuring a more accurate and reliable outcome. For reranking, we applied Density-Based Spatial Clustering (DBSCAN). This algorithm identifies clusters by ensuring each point within a cluster has a neighborhood defined by a specified radius, containing at least a minimum number of points, thereby separating dense regions from areas with fewer points.

4.4. Results and Discussions

The results presented in Table 1 show the metrics for datasets 1 and 2, referred to as db1 and db2, for the task of identifying training data fingerprints. Initially, we applied morphological opening operations, which first erode the image and then dilate the eroded image using the same structuring element. After performing these operations, we passed the images to our vision transformer models for further processing.

Table 1
Submission Results for the *Task 1:* Identifying Training Data Fingerprints.

		Dataset 1 (db1)				Dataset 2 (db2)			
Submission ID	Approach	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
1	Additive Mode Thresholding	0.502	0.502	0.489	0.495	0.491	0.491	0.487	0.489
2	Additive Mean Thresholding	0.504	0.504	0.515	0.509	0.499	0.499	0.483	0.491
3	Autoencoder Mode	0.492	0.492	0.486	0.489	0.5	0.5	0.509	0.504
4	Autoencoder Mean	0.495	0.495	0.512	0.503	0.497	0.497	0.496	0.496
5	Reranking	0.51	0.511	0.449	0.478	0.495	0.494	0.436	0.463
6	Reranking v2	0.495	0.495	0.485	0.49	0.506	0.506	0.45	0.476

Submission 1 utilized the Dinov2 model with additive mode thresholding. This model employs task-agnostic and cognitive approaches through self-supervised learning and features a multipurpose backbone derived from a pre-trained, extensive, and well-curated image set, powered by a visual

transformer model. Additive mode thresholding was then used to assign labels to each image. This method achieved (shown in Table 1) an accuracy and precision of 0.5025 for dataset 1, and an accuracy and F1 score of 0.491 for dataset 2. As shown in Table 2 the overall accuracy was 0.492.
Submission 2 used the Blip architecture with additive average thresholding. This model employs an unimodal decoder to extract pattern signatures from images. Subsequently, additive average thresholding was applied to assign labels to each image. This approach resulted in an accuracy and precision of 0.50425 for dataset 1, and an accuracy and F1 score of 0.4995 for dataset 2. As shown in 2.As shown in Table 2 the overall accuracy for submission 2 was 0.501875.
Submissions 3 and 4 are based on the concatenated multiformer feature fusion. We subsequently applied PCA and autoencoder for dimensionality reduction to both feature sets derived by the Dinov2 and Blip models. This approach yielded our best results, with submission 4 achieving an accuracy and precision of 0.49575 for dataset 1. For dataset 2, the highest accuracy and F1 score of 0.5005 were attained, making it the best among these submissions. As shown in Table 2 the overall accuracy for submissions 3 and 4 was 0.496357 and 0.4957 respectively.

Table 2
Overall Submission Results for the Identifying Training Data Fingerprints task

Submission	Identify training data "fingerprints"-Accuracy	Identify training data "fingerprints"-Precision	Identify training data "fingerprints"-Recall	Identify training data "fingerprints"-F1-score
1	0.492	0.497	0.497	0.497
2	0.5	0.501875	0.501875	0.501875
3	0.496	0.496375	0.496375	0.496375
4	0.5	0.4957	0.4957	0.4957
5	0.47	0.500875	0.500875	0.500875
6	0.483	0.502625	0.502625	0.502625

In the final two submissions, 5 and 6, we employed a reranking technique. Submission 6 provided the highest accuracy for dataset 1, reaching 0.51. Meanwhile, for dataset 2, submission 5 achieved the best accuracy at 0.506. As shown in Table 2 the overall accuracy for submissions 5 and 6 was 0.500875 and 0.502625 respectively.

Moreover for Task 2, we investigated the intriguing notion that generative models might leave distinct marks on the images they create. Our goal was to determine whether different models have unique "fingerprints" within the synthetic images they produce. By closely examining these images, we aimed to uncover the specific characteristics that define each model's output. This time the author of the ImageCLEFmed GANs [8] provided results are represented by the Adjusted Rand Index (ARI), which measures the similarity between two clusters. The ARI improves upon the Rand Index by considering the likelihood of chance agreements between clusters, thus enhancing its reliability.

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{a_{ij}}{2} - \left[\sum_i \binom{a_{i\cdot}}{2}\sum_j \binom{a_{\cdot j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_{i\cdot}}{2} + \sum_j \binom{a_{\cdot j}}{2}\right] - \left[\sum_i \binom{a_{i\cdot}}{2}\sum_j \binom{a_{\cdot j}}{2}\right]/\binom{n}{2}} \tag{4}$$

As shown in Eqn (4) we can see that the ARI is calculated with the formula: $\mathrm{ARI} = \frac{\text{Index} - \text{Expected Index}}{\text{Max Index} - \text{Expected Index}}$. Here, the Index represents the raw agreement index, which counts pairs of elements that are either in the same or different clusters in both the true and predicted clusterings. The Expected Index accounts

for the expected value of the raw index if the cluster assignments were random, while the Max Index is the maximum value of the raw index, indicating perfect clustering. The ARI ranges from -1 to 1, where 1 signifies perfect agreement, 0 indicates a random clustering, and -1 suggests no agreement. The formula leverages a contingency table and binomial coefficients to adjust the Rand Index for change, providing a more accurate measure of clustering similarity.

Table 3
Overall Submission Results for the Generative Model Fingerprint Detection Task

Submission	ARI Score
1	0.8137499357777883
2	0.9000159097044281
3	0.26753081555895303
4	0.36560477207139175
5	0.0013132463035679
6	0.0017768435
7	0.1785452554
8	0.2323909988

Among our total of eight submissions shown in Table 3 Submission 2 stood out with the highest ARI score of 0.900, demonstrating a strong agreement between the predicted clustering and the ground truth, and showcasing its exceptional performance in data clustering. In contrast, Sub5 and Sub6 recorded very low ARI scores of 0.001 and 0.002, respectively, indicating poor alignment between the predicted clusters and the actual data.

Our analysis of the results revealed a spectrum of scores for each submission, illustrating how well they matched the real data. While submissions like Sub2 performed exceptionally well, others such as Sub5 and Sub6 fell short of expectations. Overall, our findings highlight the distinctive marks generative models leave on the images they produce, which could aid in recognizing and attributing these images to specific models in the future.

5. Conclusions

In the dynamic field of medical image processing, ImageCLEFmed GANs has launched a pioneering initiative with the Identifying GAN Fingerprint task, alongside the task of detecting generative model fingerprints. In this paper, we tackled the first task by proposing six approaches, utilizing additive thresholding, autoencoder, and reranking techniques to classify images as used or not used for generating synthetic images. To address the task of detecting generative models' fingerprints, we implemented eight approaches involving Dinov2, Blip, and ensemble feature fusion. These findings highlight the significance of our efforts in advancing medical image analysis techniques. Moving forward, we plan to apply advanced noise-removing techniques to leverage pixel-level connectivity. Additionally, we aim to develop a unified framework that integrates classical and transformer-based architectures to enhance our ability to detect imprint signatures.

6. Acknowledgments

This work was supported by the National Science Foundation (NSF) grant (ID. 2131307) "CISE-MSI: DP: IIS: III: Deep Learning-Based Automated Concept and Caption Generation of Medical Images Towards Developing an Effective Decision Support".

References

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

[2] 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] S. Xun, D. Li, H. Zhu, M. Chen, J. Wang, J. Li, M. Chen, B. Wu, H. Zhang, X. Chai, Z. Jiang, Y. Zhang, P. Huang, Generative adversarial networks in medical image segmentation: A review, Computers in Biology and Medicine 140 (2022) 105063. URL: https://www.sciencedirect.com/science/article/pii/S001048252100857X. doi:https://doi.org/10.1016/j.compbiomed.2021.105063.

[4] A. Anantatamukala, K. M. Krishna, N. B. Dahotre, Generative adversarial networks assisted machine learning based automated quantification of grain size from scanning electron microscope back scatter images, Materials Characterization 206 (2023) 113396. URL: https://www.sciencedirect.com/science/article/pii/S1044580323007556. doi:https://doi.org/10.1016/j.matchar.2023.113396.

[5] A. Makhlouf, M. Maayah, N. Abughanam, C. Catal, The use of generative adversarial networks in medical image augmentation, Neural Comput. Appl. 35 (2023) 24055–24068.

[6] H. Tang, H. Liu, D. Xu, P. H. S. Torr, N. Sebe, Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks, CoRR abs/1911.11897 (2019). URL: http://arxiv.org/abs/1911.11897. arXiv:1911.11897.

[7] D. Mindlin, M. Schilling, P. Cimiano, Abc-gan: Spatially constrained counterfactual generation for image classification explanations, in: L. Longo (Ed.), Explainable Artificial Intelligence, Springer Nature Switzerland, Cham, 2023, pp. 260–282.

[8] A. 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: CLEF2023 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Thessaloniki, Greece, 2023.

[9] B. Ionescu, H. Müller, A. Drăgulinescu, W. Yim, A. Ben Abacha, N. Snider, G. Adams, M. Yetisgen, J. Rückert, A. Garcıa Seco de Herrera, C. M. Friedrich, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, S. A. Hicks, M. A. Riegler, V. Thambawita, A. Storås, P. Halvorsen, D. J. A. A. A. R. I. C. V. K. A. S. G. I. Nikolaos Papachrysos, Johanna Schöler, H. Manguinhas, L. Ştefan, M. G. Constantin, M. Dogariu, J. Deshayes, A. Popescu, Overview of ImageCLEF 2023: Multimedia retrieval in medical, socialmedia and recommender systems applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 14th International Conference of the CLEF Association (CLEF 2023), Springer Lecture Notes in Computer Science LNCS, Thessaloniki, Greece, 2023.

[10] C. Jang, Y. Lee, Y.-K. Noh, F. C. Park, Geometrically regularized autoencoders for non-euclidean data, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=_q7A0m3vXH0.

[11] T. Rhodes, T. Bhattacharjee, D. D. Lee, Learning from demonstration using a curvature regularized variational auto-encoder (curvvae), in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 10795–10800. doi:10.1109/IROS47612.2022.9981930.

[12] J. Z. Kim, N. Perrin-Gilbert, E. Narmanli, P. Klein, C. R. Myers, I. Cohen, J. J. Waterfall, J. P. Sethna,

γ-vae: Curvature regularized variational autoencoders for uncovering emergent low dimensional geometric structure in high dimensional data, 2024. arXiv:2403.01078.

[13] D. Ahn, H. Cho, J. Min, W. Jang, J. Kim, S. Kim, H. H. Park, K. H. Jin, S. Kim, Self-rectifying diffusion sampling with perturbed-attention guidance, 2024. arXiv:2403.17377.

[14] Y. Skandarani, P.-M. Jodoin, A. Lalande, Gans for medical image synthesis: An empirical study, 2021. arXiv:2105.05318.

[15] W. Ahmad, H. Ali, Z. Shah, S. Azmat, A new generative adversarial network for medical images super resolution, Scientific Reports 12 (2022). URL: http://dx.doi.org/10.1038/s41598-022-13658-4. doi:10.1038/s41598-022-13658-4.

[16] W. Zhang, W.-K. Cham, Hallucinating face in the dct domain, IEEE Transactions on Image Processing 20 (2011) 2769–2779. doi:10.1109/TIP.2011.2142001.

[17] C. Baur, S. Albarqouni, N. Navab, Generating highly realistic images of skin lesions with gans, in: OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis: First International Workshop, OR 2.0 2018, 5th International Workshop, CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16 and 20, 2018, Proceedings 5, Springer, 2018, pp. 260–267.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[19] A. F. Frangi, S. A. Tsaftaris, J. L. Prince, Simulation and synthesis in medical imaging, IEEE transactions on medical imaging 37 (2018) 673–679.

[20] C. Wang, G. Yang, G. Papanastasiou, S. Tsaftaris, D. Newby, C. Gray, G. Macnaught, T. MacGillivray, Dicyc: Gan-based deformation invariant cross-domain information fusion for medical image synthesis, Information Fusion 67 (2021) 147–160. doi:10.1016/j.inffus.2020.10.015.

[21] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on pattern analysis and machine intelligence 12 (1990) 629–639.

[22] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: International conference on information processing in medical imaging, Springer, 2017, pp. 146–157.

[23] M. J. Chuquicusma, S. Hussein, J. Burt, U. Bagci, How to fool radiologists with generative adversarial networks? a visual turing test for lung cancer diagnosis, in: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), IEEE, 2018, pp. 240–244.

[24] M. Diwakar, M. Kumar, Ct image noise reduction based on adaptive wiener filtering with wavelet packet thresholding, 2014 International Conference on Parallel, Distributed and Grid Computing (2014) 94–98. URL: https://api.semanticscholar.org/CorpusID:15070114.

[25] J. Li, D. Li, C. Xiong, S. C. H. Hoi, Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in: International Conference on Machine Learning, 2022. URL: https://api.semanticscholar.org/CorpusID:246411402.

[26] J. Li, D. Li, S. Savarese, S. Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. arXiv:2301.12597.

[27] 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, 2021. arXiv:2010.11929.

[28] J. Devlin, M. Chang, K. Lee, K. Toutanova, Pre-training of deep bidirectional transformers for language understanding, Minneapolis, MN: Association for Computational Linguistics (2019) 4171–4186.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, 2021. URL: https://api.semanticscholar.org/CorpusID:231591445.

[30] J. Li, R. R. Selvaraju, A. D. Gotmare, S. R. Joty, C. Xiong, S. C. H. Hoi, Align before fuse: Vision and language representation learning with momentum distillation, in: Neural Information Processing

Systems, 2021. URL: https://api.semanticscholar.org/CorpusID:236034189.

[31] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, Dinov2: Learning robust visual features without supervision, 2024. arXiv:2304.07193.

[32] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, 2021. arXiv:2104.14294.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255. URL: https://ieeexplore.ieee.org/abstract/document/5206848/.

[34] T. Weyand, A. Araujo, B. Cao, J. Sim, Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval, 2020. arXiv:2004.01804.

[35] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, J. Civera, Mapillary street-level sequences: A dataset for lifelong place recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2020, IEEE, United States, 2020. URL: http://cvpr2020.thecvf.com/,https://ieeexplore.ieee.org/xpl/conhome/9142308/proceeding, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020 ; Conference date: 14-06-2020 Through 19-06-2020.

[36] L. Bossard, M. Guillaumin, L. Van Gool, Food-101 – mining discriminative components with random forests, in: European Conference on Computer Vision, 2014.

[37] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library, 2024. arXiv:2401.08281.

[38] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, C. Ré, Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. arXiv:2205.14135.

[39] T. Dao, Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. arXiv:2307.08691.

[40] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, 2020. arXiv:2004.09813.

[41] N. Brown, A. Williamson, T. Anderson, L. Lawrence, Efficient transformer knowledge distillation: A performance review, 2023. arXiv:2311.13657.