Analyzing the Similarity between Artificial and Training Images in Generative Models: The PicusLabMed Contribution

Notebook for the ImageCLEFmedical GANs Lab at CLEF 2023

Michela Gravina[1,*,†], Stefano Marrone[1,†] and Carlo Sansone[1,†]

[1] *Department of Electrical Engineering and Information Technology (DIETI) of the University of Naples Federico II, Via Claudio 21, 80125, Naples, Italy*

Abstract

Generative models represent one of the most innovative and interesting applications of artificial intelligence (AI), able to generate realistic synthetic data by learning the characteristics of the training samples. In medical imaging, they are widely used to generate high-resolution medical images belonging to different modalities, improving diagnosis and patient care. However, the surprising performance of generative models has raised concerns about the relationship between artificial and real instances in terms of similarity, which may introduce privacy and ethical issues. To this aim, the ImageCLEFmed GAN challenge has been organized, asking participants to evaluate the hypothesis that generative models produce images containing the fingerprints of the samples used during the training. In this paper, we describe the methodology implemented to take part in the competition, exploiting the ability of deep neural networks to provide a high-level representation of the input data.

Keywords

Convolutional Neural Networks, Image Similarity, Generative Models

1. Introduction

In recent years, the emergence of generative models in the field of Artificial Intelligence (AI) has sparked significant interest and innovation. These models, powered by advanced Machine Learning (ML) algorithms, have the remarkable ability to generate new synthetic data samples, through a synthesis process that learns the characteristics of the distribution of the training dataset. Generative models, such as variational autoencoders (VAEs) [1], generative adversarial networks (GANs) [2], and diffusion models [3], have shown immense potential across various domains.

In the realm of medical imaging, AI generative models represent a real revolution by enabling the generation of synthetic images with exceptional realism and accuracy. These models

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

[*] Corresponding author.

[†] These authors contributed equally.

michela.gravina@unina.it (M. Gravina); stefano.marrone@unina.it (S. Marrone); carlo.sansone@unina.it (C. Sansone)

0000-0001-5033-9617 (M. Gravina); 0000-0001-6852-0377 (S. Marrone); 0000-0002-8176-6950 (C. Sansone)

can generate high-resolution images that mimic various medical imaging modalities, such as computed tomography (CT), magnetic resonance imaging (MRI), and X-ray scans. The application of AI generative models in medical imaging holds significant implications for enhancing diagnostic accuracy, data augmentation for training Deep Learning (DL) algorithms, and even aiding in the development of novel imaging techniques. However, as with any advanced technology, it becomes crucial to address the considerations related to data privacy, ethical implications, challenges associated with their use, and the interpretability of the generated images.

In this context, the ImageCLEF2023 [4] conference organized the ImageCLEFmed GAN challenge [5], focused on examining the hypothesis of an existing relationship between the synthetic images and the samples used in the training of the generative model. The competition resonates in the scientific community since, if this hypothesis is correct, synthetic biomedical images may be subject to the same sharing and usage limitations as real data, while, on the other hand, generative networks confirm their potential to create rich datasets free of ethical and privacy regulations. The aim of the task is to identify in the artificially created biomedical images distinctive patterns or characteristics known as "fingerprints", that help in determining the set of images employed during the training phase of the generative model.

In this paper, we present the methodology implemented to take part in the ImageCLEFmed GAN challenge [5]. In particular, we exploit the ability of the DL models to provide a representation of the input data relying on Convolutional Neural Networks (CNNs) to extract the features from the real and generated images. These features represent the fingerprints that we then analyze, adopting a ML model for the identification of the samples used during the development of the generative model among all the real instances. We propose two variants for the features extraction step, introducing Vector-Net, a convolutional network that learns how to map the input image in an efficient representation, and leveraging a Deforming Autoencoder (DAE) [6], that provides a latent vector in an unsupervised manner.

The rest of the paper is organized as follows: Section 2 introduces the implemented methodology; Section 3 descrives the experimental set-up; Section 4 reports the obtained results; finally Section 5 provides some conclusions.

2. Methods

In image processing, the generative model, or generator, is typically a deep network that learns how to map a fixed latent distribution to the distribution of real data p_r. Denoting with p_g the distribution of the artificial instances, the goal is to learn p_g which approximates p_r as closely as possible [2] with the aim of creating new samples preserving the intrinsic characteristics of the real ones. In the ImageCLEFmed GAN [5] challenge we are asked to evaluate the relationship between p_g and p_r, investigating the possibility to distinguish among all the images belonging to p_r, the subset used during the training of the generative network. In other words, we asses the partition of the set of real data R into two subsets, U and NU, corresponding to the images used (U) and not used (NU) during the development of the model, respectively.

In our methodology, we rely on CNNs to extract the features from the real and generated images, thanks to their ability to autonomously learn the data representations well-suited for the specific task to be solved. In particular, we explore two different approaches, where the former leverages the features extracted by Vector-Net, a convolutional network aware bout the hypothesis to be tested, while the latter considers the latent representation provided by a Deforming Autoencoder (DAE) [6] trained in an unsupervised manner. The extracted features vector represents the "fingerprint" that we propose to use to analyze the relationship between p_g and p_r by exploiting a ML model specifically trained for the distinction between elements belonging to U and NU.

2.1. Vector-Net for fingerprint extraction

Vector-Net is the proposed CNN that aims to provide a mapping function f able to project the input images in a latent space where the extracted fingerprints contribute to the identification of the samples used for the training of the generative model among all the real instances. In particular, denoting with x a generic input belonging to one of the sets G, U, and NU, f is applied to generate the representation $\tilde{x}$, where $\tilde{x} = f(x)$, whose characteristics are considered for the distinction of the elements in R. Figure 1 shows an illustrative example of the effects of the mapping function on the samples of G, U, and NU, referred as g ($g \in G$), u ($u \in U$), and nu ($nu \in NU$). Indeed, it highlights that although it is extremely hard to discriminate g, u, and nu, the resulting fingerprints $\tilde{g}$, $\tilde{u}$ and $\tilde{nu}$ can be used to analyze the hidden relationship between the three sets of images.

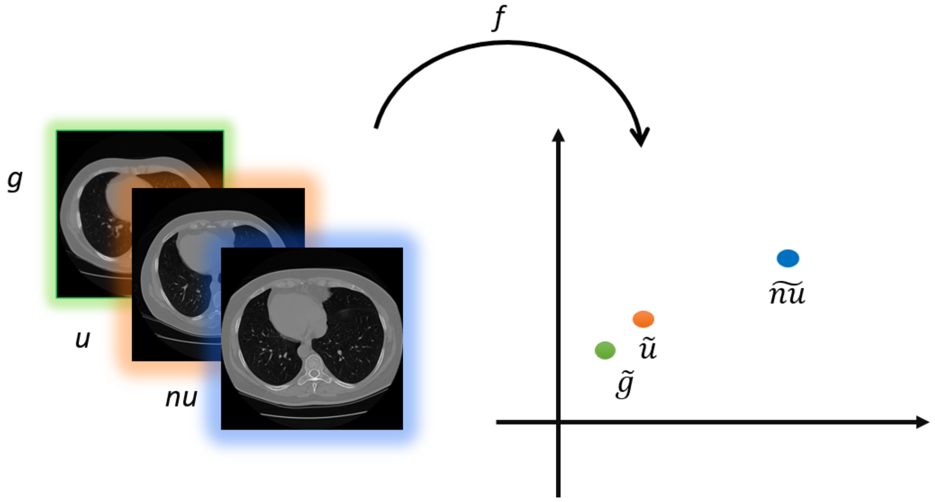

Figure 1: Illustrative example of the application of f to the element g ($g \in G$), u ($u \in U$), and nu ($nu \in NU$).

The architecture of the implemented Vector-Net consists of six convolutional blocks, a Global Average Pooling [7] operation and a fully connected layer. In particular, each convolutional block includes a convolutional layer, with a 4×4 kernel and values of padding and stride set to 1 and 2 respectively, in order to provide a reduction of the dimensionality of the features maps, followed by Batch Normalization and ReLU as activation function. The first convolutional layer presents a one-channel input and 32 output channels, that are doubled by each step in the chain of the six convolutional blocks. The Global Average Pooling is used to generate a features vector which feds the last fully connected layer with 1024 input and 64 output neurons.

The 64-element vector represents the fingerprint $\tilde{x}$ extracted from a generic input x, that embeds an image in a 64-dimensional Euclidean space. As suggested in the work proposed in [8], we constrain this vector to live in a hypersphere ensuring that $\|\tilde{x}\|_2 = 1$ in introducing a normalization criterion among the generated fingerprints.

In our methodology, we evaluate the ability of the function f in producing fingerprints that enhance the supposed similarity between the sets G and U and make clear the distinction with NU. To this aim, we consider the distance of the vector representations as a measure of their similarity, adopting the Euclidean metric. Then, we use the triplet loss [9] to train the implemented Vector-Net to minimize the distance between $\tilde{g}$, and $\tilde{u}$, while maximizing the dissimilarity with $\tilde{nu}$. Indeed, the triplet loss [9] ensures that a reference element, denoted as *anchor (a)*, is located close to *positive (p)* samples and beyond a certain margin (m) from the *negative (n)* ones, guaranteeing that $d(a,p) + m < d(a,n)$, where d denotes the Euclidean distance, as represented in Figure 2.

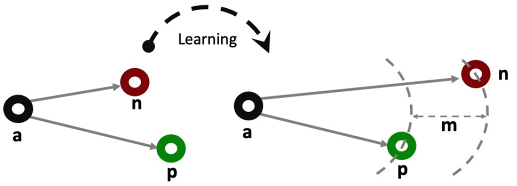

Figure 2: Relation between the samples when the triplet loss is used

In accordance with which of the fingerprints $\tilde{g}$, $\tilde{u}$, and $\tilde{nu}$ is treated as a, p, or n, it is possible to include different constraints during the training of Vector-Net. When the representations of the elements belonging to G are considered as a, we propose to denote the fingerprints $\tilde{u}$, and $\tilde{nu}$ as positive and negative vectors respectively, ensuring that:

$$d(\tilde{g}, \tilde{u}) + m < d(\tilde{g}, \tilde{nu}) \tag{1}$$

$\forall$ $\tilde{g}$, $\tilde{u}$ and $\tilde{nu}$ generated by f when applied to the samples of G, U and NU. The Equation 1 is required to make the network able to search for the supposed similarity between the images u and g, extracting for them fingerprints that are close (or similar) in the Euclidean space, when compared to those belonging to the nu elements. This allows the introduction of another constraint treating the representations $\tilde{u}$, $\tilde{g}$ and $\tilde{nu}$ as the anchor, the positive and the negative samples respectively, and formalized as follows:

$$d(\tilde{u}, \tilde{g}) + m < d(\tilde{u}, \tilde{nu}) \tag{2}$$

We need a CNN able to extract fingerprints from U that allow the recognition of the used images, exploiting their similarity with the representations $\tilde{g}$ and, at the same time, their diversity from the $\tilde{nu}$. The Equation 2 is introduced to explicitly separate the features vectors obtained from u and nu, while further ensuring the similarity between $\tilde{u}$ and $\tilde{g}$. We argue that the diversity between $\tilde{u}$ and $\tilde{nu}$ should be preserved since it is the result of the absence of a relationship between the sets U and NU, which consist of independent samples representing the distribution of the real data ($U \subseteq R$ and $NU \subseteq R$). Indeed, while U and G have been involved in the development of the same generative model, thus explaining the need of testing the hypothesis of their similarity, U and NU are two separate sets extracted from the same distribution.

In general, the presence in the set R, including both U and NU, of images belonging to different and unrelated patients suggests the absence of a relationship among the real instances, which is exploited in the definition of a third constraint focusing on the U set and supporting the training of the CNN for the extraction of the fingerprints. In particular, we propose to bind the function f to extract the vector representations from images in such a way as to minimize the distance between $\tilde{g}$ and $\tilde{u}$, and preserve the independence between the different elements of U. Denoting with u_1 and u_2 two instances belonging to the set of used samples, with $u_1 \in U$, $u_2 \in U \mid u_1 \neq u_2$, we formalize the third constraint as follows:

$$d(\tilde{u_1}, \tilde{g}) + m < d(\tilde{u_1}, \tilde{u_2}) \tag{3}$$

where $\tilde{u_1}$ and $\tilde{u_2}$ are the fingerprint extracted with the application of f. The Equation 3, which considers the representations extracted from U both anchor and negative vectors, is introduced to make the CNN focus only on the search of the similarity between $\tilde{g}$ and $\tilde{u}$, minimizing their distance, while maintaining the distinctiveness among the different and unrelated real images.

2.2. DAE for unsupervised fingerprint extraction

The Deforming Autoencoder (DAE) [6] is an unsupervised encoder-decoder architecture designed to disentangle an image in its main components of shape and texture. It exploits the basic notion that creating an image involves the combination of two processes: a synthesis of appearance on a coordinate system with no distortion (referred to as a "template"), followed by a second deformation that includes shape diversity. The network architecture is reported in Figure 3, consisting of an Encoder (E) and a set of two decoders (D_s and D_t), for the synthesis of the shape (S) and the texture (T), respectively.

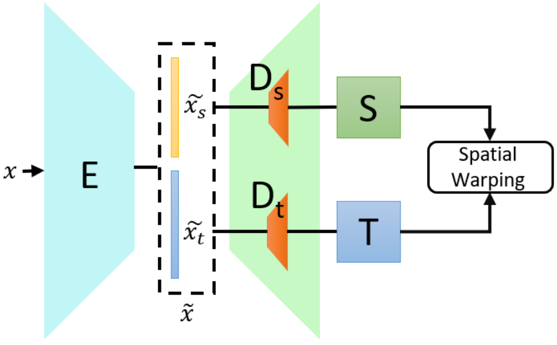

Figure 3: DAE architecture consisting of an Encoder (E) and a two decoders (D_s and D_t)

Denoting with x a generic input image, E implements a function providing the latent representation $\tilde{x}$ ($\tilde{x} = E(x)$) which is then used by D_s and D_t. In particular, $\tilde{x}$ is then split into two different parts $\tilde{x}_s$, and $\tilde{x}_t$ ($\tilde{x} = [\tilde{x}_s, \tilde{x}_t]$), representing the latent shape and texture, thus introducing a clear image decomposition. Each of these parts is fed to a specific decoder generating the disentangled components of the image. The DAE is trained according to its ability to reconstruct the input starting from its main components using the loss function (L_{DAE}) proposed in [6], which consists of the sum of two elements: the reconstruction loss (L_{rec}), implemented as the standard l_2 norm, and the warping loss (L_{warp}) used to create visually realistic samples [6].

The encoder architecture consists of an initial convolutional layer with a 4×4 kernel, values of padding and stride set to 1 and 2 respectively, followed by a LeakyReLU activation function, a chain of n_e encoding blocks, and a last convolutional layer with a 4×4 kernel and a Sigmoid function. Each encoding block includes a convolutional operation, with a 4×4 kernel and values of padding and stride set to 1 and 2 respectively, a Batch Normalization, and a LeakyReLU function. The first layer presents an input channel set to 1, with 32 output channels, that are doubled in each step of the chain of the encoding blocks. The last convolutional operation provides a 128-element vector corresponding to the latent representation of the input.

The D_s and D_t present the same basic architecture, but differ for the activation functions. In particular, both decoders consist of a chain of n_d decoding blocks, followed by a transposed convolution operation with a 3×3 kernel, and activation function that is the HardTanh in D_t and the Sigmoid in D_s. Each decoding block includes a transposed convolution layer with a 4×4 kernel, a Batch Normalization, and a ReLU or hyperbolic (tanh) function for D_t and D_w respectively.

In our methodology, we train the DAE considering the samples belonging to the sets of G, U, and NU. The encoder E is then used to generate a 128-element vector for each input image that is treated as the "fingerprint" of each instance, extracted in an unsupervised manner. Indeed, the information about the source of the images is not exploited, with the aim of testing if the decomposition in the shape and texture components generates elements highlighting the similarity between G, and U.

3. Experimental Set-Up

The datasets involved in the competition consist of axial 3D computed tomography (CT) images of about 8000 lung tuberculosis patients, stored in the form of 8 bit/pixel PNG images with dimensions of 256x256 pixels. The synthetic images are 256x256 pixels in size and are generated using a Diffusion Model [3]. The training test provided to test the hypothesis of the similarity between the artificial and the real samples includes 500 synthetic images (G), 80 real instances not used for the development of generative neural networks (NU) as well as 80 real images taken from the image set used for training corresponding generative model (U). In the test set, a total of 10000 generated images and 200 real samples are provided.

After their training, the Vector-Net and the encoder of the DAE are applied to the all the elements $g \in G$, $u \in U$, and $nu \in NU$, generating the representation $\tilde{g}$, $\tilde{u}$, and $\tilde{nu}$. The extracted fingerprints are analyzed by assessing their effectiveness in the distinction between elements belonging to U and NU. We leverage two ML models, namely the Support Vector Machines (SVM) [10] and the K-Nearest Neighbours (KNN) [11], to detect among the real instances, those used during the development of the generative model. In other words, we perform a binary classification task, exploiting the fingerprint extracted from the networks, associating the label "1" to those belonging to used images $\tilde{u}$ and "0" to the representations $\tilde{nu}$. The choice of the SVM and KNN relies on the fact that they represent two different models, where the former aims to determine a hyperplane to separate the classes, while the latter applies the distance metric that can be interpreted as a similarity score.

The experiments implemented for the ImageCLEFmed GAN challenge [5] vary according to different aspects related to the characteristics of the networks used for the extraction of the fingerprints, the involved ML model, and the set of features representations used to train the SVM and KNN.

When the Vector-Net is used for fingerprint extraction, we evaluate the contribution of the proposed constraints by adding them one at a time. As a consequence, we denote as Vector-Net(1), Vector-Net(1,2) and Vector-Net(1,2,3) the approaches including the Equation 1, Equations 1 and 2, and Equations 1, 2, 3, respectively.

As aforementioned, the DAE is used to provide an unsupervised features vector definition. We consider three different variants obtained by changing the set of images used during the training of the network, with the aim of evaluating if their presence or absence affects the fingerprints extraction. In particular, we denote as DAE(G,U), DAE(G,NU), and DAE(G,U,NU), the experiments involving the DAE architecture and the sets of samples specified in brackets. It is worth noting that after the training step, all the approaches are applied to all the elements $g \in G$, $u \in U$, and $nu \in NU$.

Two different ML models are used for the identification of the subset of the real images used for the development of the generative network. After a hyper-parameter optimization step, we set the number of neighbors in KNN to 5, and explore both linear and the polynomial (degree = 2) kernel in the SVM to investigate also more complex boundaries among the vector representations, denoting with SVM-Linear and SVM-2 the two variants, respectively.

In our experiments, we also evaluate the impact of the fingerprints extracted from G in the classification task proposing two training strategies, referred as "REAL" and "FULL". The former uses only the fingerprints generated from U and NU, namely $\tilde{u}$, and $\tilde{nu}$, to train the two ML models, while the latter includes also the representations of the elements of G by associating to the vectors $\tilde{g}$ the same label of $\tilde{u}$. In both cases, we aim to evaluate the effectiveness of the CNNs in the generation of features representations that enhance the supposed similarity between G and U, and the dissimilarity with NU. However, in the "REAL" strategy, we explicitly evaluate how the extracted fingerprints are able to determine a separation among the elements of U and NU, without relying on the presence of $\tilde{g}$. When the "FULL" option is explored, we apply the adaptive synthetic sampling approach (Adasyn [12]) to handle the imbalance between the labels "1" and "0".

During the experiments, we train the Vector-Net and the DAE using the loss functions defined in Section 2. The DAE architecture is implemented following the work proposed in [6], but modifying the networks to handle a 256×256 input. In particular, we add encoding and decoding blocks both in the Encoder and Decoder, considering n_e and n_d set to 5 and 7 respectively. The maximum number of epochs is set to 1000 and 500 for the DAE and the Vector-net, respectively. The batch size is 32, the learning rate for the triplet loss is 10^{-5}, and $2 \cdot 10^{-4}$ for the L_{DAE}. Adam optimizer is used with a decay set to 10^{-4}.

Performance is evaluated in terms of accuracy (ACC), F1-score (F1) and Recall (R), as suggested in the competition. It is worth noting that the evaluation step is performed only considering the images belonging to U and NU as test set, thus evaluating the hypothesis of the competition. All the experiments were run in a 10-folds cross-validation (CV) setting, to better assess the generalization ability of each approach, using Pytorch for the training of the Vector-Net and the DAE and MATLAB 2020b for the classification task involving the SVM and the KNN. A Linux workstation equipped with Intel(R) Core(TM) i7-10700KF CPU, 64 GB of DDR4 RAM and a Nvidia RTX 3090 GPU is used.

We took part in the ImageCLEFmed GAN challenge [5] with 10 different submissions. As a consequence, we have a partial evaluation of the experiments using also the real samples of test set provided by the competition.

4. Results and Discussion

This section reports the results of the implemented methodology, including different variants. In particular, Table 1 shows the performance of the experiments in CV setting, thus exploiting the dataset provided by the ImageCLEFmed GAN challenge [5]. It consists of six sections, that differ according to the network used for the fingerprints extraction step as detailed in the column *Net.*, while the training strategy and the ML exploited for the classification are reported in columns *Stretegy*, and *ML Model* respectively. For each vector extractor, the best values are reported in bold.

Table 1 highlights that the fingerprints extracted with Vector-Net are able to provide a better separation between U ad NU in comparison with those obtained with the DAE. We argue that this characteristic depends on the introduction of task-specific constraints during network training. Indeed, the results achieved involving the DAE reveal that relying solely on shape and texture information is not enough to determine supposed similarity among artificial images and the samples used during the training of the generative model. Moreover, the presence of the $\tilde{g}$ vectors in the "FULL" strategy negatively impacts performance, as a consequence of the evident difference among the distributions of real p_r and generated p_g data. Some preliminary experiments (not reported in this paper) focusing on the distinction of the elements of G from R showed that it is possible to separate the synthetic images from the real ones with high accuracy (0.9970). Therefore, merging the fingerprints $\tilde{g}$ with $\tilde{u}$ during the training makes the ML model exploit the dissimilarity of p_r and p_g, determining a boundary that includes in the same region the representations of U ad NU. In addition, it is worth noting that in the "FULL" strategy the presence of the features of G generates a very unbalanced dataset. Othe other hand, in the "REAL" strategy, the model is forced to determine a separation among elements belonging to the same distribution p_r. This aspect also explains the reduced gap in the performance when the Vector-Net(1,2) is considered in comparison with the other experiments. Indeed, Equations 1 and 2 ensure that $d(\tilde{g}, \tilde{u}) + m < d(\tilde{g}, \tilde{nu})$ and $d(\tilde{u}, \tilde{g}) + m < d(\tilde{u}, \tilde{nu})$, thus enhancing the separation among U ad NU.

A subset of the conducted experiments has been submitted to the ImageCLEFmed GAN challenge [5]. It is worth noting that only a part of the proposed variants has been implemented before the deadline of the competition. Indeed, we started to address the problem of the relationship between U and NU and we continued after the end of the challenge. As a consequence, we

Table 1
Performance of the implemented experiments evaluated in 10-fold CV setting.

Net.	Strategy	ML Model	Acc	R	F1
Vector-Net (1)	FULL	SVM-Linear	0.5694	0.4028	0.4833
		SVM-2	0.6458	0.6250	0.6383
		KNN	0.6111	0.6250	0.6164
	REAL	SVM-Linear	0.7569	**0.8750**	**0.7826**
		SVM-2	**0.7639**	0.8333	0.7792
		KNN	0.7431	0.8333	0.7643
Vector-Net (1,2)	FULL	SVM-Linear	0.7987	0.9506	0.8221
		SVM-2	0.7014	0.9444	0.7598
		KNN	0.8264	0.9583	0.8466
	REAL	SVM-Linear	0.9583	**0.9444**	0.9578
		SVM-2	**0.9653**	**0.9444**	**0.9645**
		KNN	0.9167	0.8889	0.9143
Vector-Net (1,2,3)	FULL	SVM-Linear	0.6181	0.8333	0.6857
		SVM-2	0.5694	**0.9861**	0.6908
		KNN	0.5764	0.7639	0.6433
	REAL	SVM-Linear	**0.8194**	0.8472	**0.8243**
		SVM-2	0.8056	0.8194	0.8082
		KNN	0.7222	0.8611	0.7561
DAE (G,U)	FULL	SVM-Linear	0.4180	0.2773	0.3227
		SVM-2	0.4805	**0.6016**	0.5366
		KNN	0.5117	0.5781	0.5421
	REAL	SVM-Linear	0.4512	0.4102	0.4277
		SVM-2	**0.6113**	0.5391	**0.5811**
		KNN	0.4824	0.2539	0.3291
DAE (G,NU)	FULL	SVM-Linear	0.4805	0.3477	0.4010
		SVM-2	0.5215	0.6874	**0.5896**
		KNN	0.5020	0.6836	0.5785
	REAL	SVM-Linear	0.4785	0.5391	0.5083
		SVM-2	**0.5449**	0.6133	0.5740
		KNN	0.4805	**0.7031**	0.5751
DAE (G,U,NU)	FULL	SVM-Linear	0.5125	0.6375	0.5667
		SVM-2	0.5875	**0.9750**	0.7027
		KNN	**0.6438**	0.8875	**0.7136**
	REAL	SVM-Linear	0.4625	0.4625	0.4625
		SVM-2	0.4813	0.5000	0.4908
		KNN	0.5000	0.5625	0.5294

only have a partial evaluation of the results considering the real images included in the test set. Table 2 shows the performance obtained by submitting the variants with the SVM classifier, and the "FULL" training strategy. The column *Submission* details the name of the submission file and in the last row the experiment "PicusLabMed_submission10.csv" is generated by implementing a voting strategy among the other results. As we expected from the results reported in Table 1, the presence of the $\tilde{g}$ representations during the training of the ML models does not lead to good results. Moreover, the experiments with the Vector-Net (1,2) present overfitting showing good performance in Table 1, but low values in Table 2. We argue that this characteristic may reflect the weakness of the models when applied to a test set with different characteristics, thus highlighting the need for a more robust approach.

Table 2
Performance of the implemented experiments on the test set provided by the competition using the training strategy "FULL"

Submission	Net.	ML Model	Acc	R	F1
PicusLabMed_submission1.csv	Vector-Net (1)	SVM-Linear	0.5050	0.3800	0.4343
PicusLabMed_submission2.csv		SVM-2	0.5050	0.4400	0.4706
PicusLabMed_submission3.csv	Vector-Net (1,2)	SVM-Linear	0.4750	0.3800	0.4199
PicusLabMed_submission4.csv		SVM-2	0.5150	0.6000	0.5530
PicusLabMed_submission5.csv	Vector-Net (1,2,3)	SVM-Linear	0.4550	0.3900	0.4171
PicusLabMed_submission6.csv		SVM-2	0.5250	0.7900	0.6245
PicusLabMed_submission7.csv	DAE (G,U)	SVM-Linear	0.5150	0.0500	0.0934
PicusLabMed_submission8.csv	DAE (G,NU)	SVM-Linear	**0.5300**	**0.9400**	**0.6667**
PicusLabMed_submission9.csv	DAE (G,U,NU)	SVM-Linear	0.5250	0.6100	0.5622
PicusLabMed_submission10.csv	-	-	0.5050	0.4700	0.4870

5. Conclusions

The surprising performance achieved by generative models in the creation of realistic synthetic images has raised the need to address considerations related to data privacy and ethical implications, especially in sensitive domains, such as the medical one. To this aim the ImageCLEFmed GAN challenge [5] has been organized, asking participants to test the hypothesis of the similarity between the set of real images used during the development of the generative model and the synthetic samples. In this paper, we described the methodology we implemented to take part in the competition, designing experiments that analyze the impact of two diverse CNNs in the fingerprints extraction step and the ability of the ML models in the classification with different training strategies. Indeed, Vector-Net and the DAE represent two approaches where the former explicitly exploits the constraints related to the hypothesis to be tested, while the latter aims to investigate if information about the texture and the shape of the image contribute to the evaluation of the supposed similarity. Despite the poor performance obtained from some experiments, we argue that there is the need to explore other approaches before ruling out all the concerns regarding the application of synthetic images in medical field. To this aim, we will consider the obtained results as a baseline, improving in future works the set of constraints required to extract fingerprints able to reveal the images used during the training of the generative models.

Acknowledgments

This work is part of the POR FESR CAMPANIA 2014-2020 Synergy for COVID project (CUP H69I22000710002).

References

[1] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (2020) 139–144.

[3] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in Neural Information Processing Systems 33 (2020) 6840–6851.

[4] B. Ionescu, H. Müller, A. Drăgulinescu, W. Yim, A. Ben Abacha, N. Snider, G. Adams, M. Yetisgen, J. Rückert, A. García 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, N. Papachrysos, J. Schöler, D. Jha, A. Andrei, A. Radzhabov, I. Coman, V. Kovalev, A. Stan, G. Ioannidis, 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.

[5] 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.

[6] Z. Shu, M. Sahasrabudhe, R. Alp Guler, D. Samaras, N. Paragios, I. Kokkinos, Deforming autoencoders: Unsupervised disentangling of shape and appearance, in: ECCV, 2018.

[7] M. Lin, Q. Chen, S. Yan, Network in network, arXiv preprint arXiv:1312.4400 (2013).

[8] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[9] G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking., Journal of Machine Learning Research 11 (2010).

[10] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[11] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE transactions on information theory 13 (1967) 21–27.

[12] H. He, Y. Bai, E. A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), IEEE, 2008, pp. 1322–1328.