The Journal of Supercomputing (2023) 79:2850–2881
https://doi.org/10.1007/s11227-022-04775-y

How much BiGAN and CycleGAN-learned hidden features are effective for COVID-19 detection from CT images? A comparative study

Sima Sarv Ahrabi[1] · Alireza Momenzadeh[1] · Enzo Baccarelli[1] · Michele Scarpiniti[1] · Lorenzo Piazzo[1]

Accepted: 10 August 2022 / Published online: 26 August 2022

Abstract

Bidirectional generative adversarial networks (BiGANs) and cycle generative adversarial networks (CycleGANs) are two emerging machine learning models that, up to now, have been used as generative models, i.e., to generate output data sampled from a target probability distribution. However, these models are also equipped with encoding modules, which, after weakly supervised training, could be, in principle, exploited for the extraction of hidden features from the input data. At the present time, how these extracted features could be effectively exploited for classification tasks is still an unexplored field. Hence, motivated by this consideration, in this paper, we develop and numerically test the performance of a novel inference engine that relies on the exploitation of BiGAN and CycleGAN-learned hidden features for the detection of COVID-19 disease from other lung diseases in computer tomography (CT) scans. In this respect, the main contributions of the paper are twofold. First, we develop a kernel density estimation (KDE)-based inference method, which, in the training phase, leverages the hidden features extracted by BiGANs and CycleGANs for estimating the (*a priori* unknown) probability density function (PDF) of the CT scans of COVID-19 patients and, then, in the inference phase, uses it as a target COVID-PDF for the detection of COVID diseases. As a second major contribution, we numerically evaluate and compare the classification accuracies of the implemented BiGAN and CycleGAN models against the ones of some state-of-the-art methods, which rely on the unsupervised training of convolutional autoencoders (CAEs) for attaining feature extraction. The performance comparisons are carried out by considering a spectrum of different training loss functions and distance metrics. The obtained classification accuracies of the proposed CycleGAN-based (resp., BiGAN-based) models outperform the corresponding ones of the considered benchmark CAE-based models of about 16% (resp., 14%).

Extended author information available on the last page of the article

Keywords COVID-19 detection · Hidden feature extraction · BiGAN · CycleGAN · CAE · Unsupervised-vs.-weakly supervised learning · Complexity-vs.-accuracy comparisons

1 Introduction

The chest computed tomography (CT) scan is generally regarded as beneficial in diagnosing COVID-19 diseases and is especially useful when it is used in tandem with clinical examinations [1–5]. Due to the effective use of deep learning (DL) in computer vision and biomedical domains, researchers have explored the efficiency of DL-based methods to recognize COVID-19 from lung CT scans. The current DL approaches can be categorized as supervised, unsupervised, or weakly supervised methods.

1.1 Supervised learning approaches

A large number of research papers adopt supervised learning methods for the reliable detection of COVID-19 diseases [6–16] [17–20]. However, due to the lack of publicly available CTs on COVID-19 patients, researchers have been triggered to consider this deficiency, especially at the beginning of the spread of COVID-19. For instance, the authors of [21–30] adopt *transfer learning* methods to address the lack of large-sized data sets. In [31], the authors utilize GoogleNet and ResNet for supervised COVID-19 classification. The authors of [32] propose a statistical method to address issues, like as huge computational complexity and large datasets required by deep networks. In [33], a segmented CT scan is used as the input of a random forest classifier approach. The authors of [21] used an inception network on CT scans, but the resulting classification accuracy was below average. In [34], the authors propose a contrast enhancement scheme for CT scans, followed by a pre-trained VGG16 and AlexNet classification, reporting good accuracy.

However, the accuracy performance of supervised-trained models typically crashes when CT scans are used in the test phase that belong to unseen classes (that is, classes of test data which are not present in the training data sets). In principle, this loss of robustness suffered by supervised DL models may be effectively by-passed by resorting to unsupervised or weakly supervised DL models, which are trained only on data sets of the COVID-19 class. So doing, it expected that an unsupervised/weakly supervised trained model may differentiate the COVID-19 class (i.e., the target class) from any other type of unseen chest images (i.e., the novelties) in a reliable way.

1.2 Unsupervised learning approaches

Autoencoders (AEs) have been employed in [35–40]. Specifically, the work in [37] focuses on a two-stage learning method and a triple classification task. The authors train their AE model on classes of COVID-19, pneumonia, and normal cases separately. After obtaining the hidden feature vectors of all classes, a feature classifier is trained. The authors of [38] build up a robust statistical target histogram by exploiting the feature representations, which are generated by an unsupervised-trained denoising convolutional AE (DCAE). The proposed method estimates the statistical distance between unknown and target histograms to classify the images according to suitably set decision thresholds. The DCAE proposed in [36] is trained on COVID-19, pneumonia, and a few other types of chest X-rays. Then, the hidden feature vector of a test image is compared to the features of the selected training data sets. The so-trained AE exhibits good test performance. However, unlike our work, this approach relies on training the considered model over each decision class and, then, does not guarantee to instances of unseen class.

1.2.1 Generative Adversarial Networks (GANs)-based approaches

Motivated by the aforementioned considerations, we are interested in deep generative models, because learning COVID-19 patterns can be viewed as learning the distribution of the available training data. According to a recent taxonomy in medical image classification [41], we adopt the weakly supervised terminology for indicating the exploitation of two sets of unpaired images. Being unsupervised/weakly supervised models, deep generative models (DGMs) aim to unveil meaningful patterns in raw data. DGMs enable the approximation of statistical data distributions through density estimation. Deep neural networks (DNNs) are based only on point estimates and make deterministic predictions by using suitable feature vectors. Most works on DNNs do not pay much attention to the complexity of these models. On the other hand, probabilistic models typically rely on statistical hypothesis tests, which are more simple to implement through the computation of suitable distances in the latent space [42]. The actual capability of GANs to generate data makes them attractive for anomaly detection under two perspectives [43]. First, GANs can potentially help to generate hard-to-acquire anomalous data points. Second, they can be used to learn the distribution of data for normal operating conditions and, then, can be exploited as anomaly or outlier detectors [44]. A conditional GAN-based model, called CovidGAN, is proposed in [45], which generates synthetic chest X-ray images to augment the available training set. The authors of [46] develop a Dense GAN and a multi-layer attention-based segmentation method for the generation of higher quality images. GANs are also utilized in [47], in order to generate X-rays data sets from 307 images of four different types. The method employed in [48] utilizes auxiliary classifier generative adversarial network (AC-GAN) to generate COVID-19 CT scans. Then, the authors of [48] compare their approach against competing DL models using transfer learning. The authors of [49] introduce a Mean Teacher plus Transfer GAN (MTT-GAN) model, in order to generate COVID-19 chest X-ray images of high quality. Inception-Augmentation GAN (IAGAN), a semi-supervised GAN-based augmentation method, is introduced in [50], in order to improve the detection capability of pneumonia and COVID-19 in chest X-ray images. The authors of [51] present the QuNet model to classify the COVID-19-infected patients by using X-ray images. In [52] an Enhanced Super Resolution GAN (ESRGAN) is used in order to improve the CT scan quality, before feeding it to a Siamese Capsule network.

Additionally, in [53], MESRGAN+ is derived by implementing a connected nonlinear mapping between noise-contaminated low-resolution input images and deblurred and denoised HR images using the building blocks of GAN. A summarizing overview of the main literature on GAN-based models for COVID-19 detection is provided in Table 1.

Overall, differently from the proposed work, all these approaches do not exploit the use of an additional encoder (BiGAN) or a second generator (CycleGAN) that, ideally, learn to invert the mapping performed by the first generator. We argue that a trained BiGAN encoder and a pair of generators/discriminators, respectively, could provide useful feature representation for related tasks of scan classification. Although the increased computational cost with respect to a standard GAN architecture, we can expect that, by considering the performance-vs.-complexity tradeoff, the proposed method can represent a promising approach for the robust classification of the COVID-19 disease from unlabeled CT scans.

1.3 Paper contributions and roadmap

Motivated by the performed review, in this contribution, we aim at exploiting how and at which extend the hidden features learned by weakly supervised BiGAN [55] and CycleGAN [56] models could be effectively exploited for robust classification of COVID-19 diseases from unlabeled CT scans. In fact, both BiGAN and CycleGAN allow to efficiently extract meaningful features of the target class from the encoded vector, which can be successfully used to construct a statistical representation suitable to detect scans of COVID-19 patients from the others. Specifically, the main contributions of this paper are the following ones:

- We exploit the kernel density estimation (KDE) approach for deploying an inference method that utilizes the hidden features generated during

Table 1 Synoptic view of recent work on GAN-based COVID-19 detection

Ref	Approach	Goal	Training class
[45]	CovidGAN	X-ray synthetic augmentation	COVID
[47]	GAN +transfer learning	X-ray synthetic augmentation	Normal+COVID+Pneumonia
[46]	Dense GAN+ U-Net	Enhancing the quality of CT	COVID
[54]	Convolutional GAN	X-ray synthetic augmentation	Normal+COVID+Pneumonia
[48]	AC-GAN	CT synthetic augmentation	COVID
[49]	MTT-GAN	X-ray synthetic augmentation	COVID
[50]	IAGAN	X-ray synthetic augmentation	COVID
[51]	GAN	X-ray synthetic augmentation	Normal+COVID+Pneumonia
[52]	Siamese-CapsNet	Image resolution enhancement	COVID+NON-COVID
[53]	GAN	Image resolution enhancement	COVID+NON-COVID
This contribution	BiGAN & CycleGAN	CT Classification	COVID

the weakly supervised training of BiGANs and CycleGANs for estimating the underlying PDF of CT scans of COVID-19 patients, namely the target COVID-PDF. Afterward, in the test phase, the trained BiGAN/CycleGAN encoder is used for extracting the hidden features from the corresponding COVID/Non-COVID CT test scan, and, then, the distance among the target COVID-PDF and the corresponding PDF of the hidden features extracted from each test image is used for binary classification. For this purpose, a suitably designed binary detector is employed, which is equipped with a tunable decision threshold;

- We numerically evaluate the sensitivity of the achieved accuracies, test times and training times of the implemented BiGANs and CycleGANs on the employed training loss functions and inter-PDF distance metrics. The tested loss training functions are the cross-entropy (CE), least squares (LS) and Wasserstein (W) ones, while the Euclidean, Kullback-Leibler (KL) divergence, Correlation and Jensen-Shannon (JS) divergence are tested as inter-PDF distance metrics;
- The training of the BiGAN and CycleGAN models is, by design, of weakly supervised type. Hence, as a final contribution, we compare the attained BiGAN and CycleGAN performance against the corresponding ones of some recently published methods [38] and [57], which exploit the encoders of unsupervised trained CAEs as feature extractors. In this regard, we anticipate that the implemented CycleGAN model achieves the highest test accuracy, while the tested CAE models attain the lowest test and training times. The corresponding accuracies, test times and training times of the implemented BiGAN models fall somewhat in the middle.

To the best of our knowledge, the exploitation of a KDE estimation of the target COVID-PDF from the feature encoded by the BiGAN and CycleGAN for the classification of COVID/Non-COVID CT scans is novel and not yet investigated in the current literature.

The rest of the paper is organized as follows. In Sect. 2, we describe, at first, the employed training/test data sets, the implemented BiGAN and CycleGAN models and the related training loss functions. Afterward, we present the proposed KDE-based method for test inference. Section 3 is devoted to the presentation of the obtained numerical results and related performance comparisons. Finally, the conclusive Sect. 4 summarizes the main results of the paper and highlights some possible hints for future research.

2 Material and solving method

This section describes the used data sets and the implemented BiGAN and CycleGAN-based architectures for feature extraction, together with the companion PDF-based approach pursued for binary classification of the test images.

2.1 Training and testing data sets

We selected 1000 COVID-19 CT scans related to 500 (anonymous) patients from several multiple open-access data sets [58], in order to generate the training data set. However, before training, a pre-processing step has been carried out, in which the borders of all CT scans have been cropped and all the gray-scale images have been resized to 100×100 pixels, in order to achieve a suitable processing complexity-vs.-image resolution trade-off. Finally, the per-pixel mean of each image has been evaluated and subtracted. In the sequel, we will indicate as y (resp. Y) an input COVID-19 training image (resp., the set of the COVID-19 training images). For illustrative purposes, Fig. 1a reports four examples of COVID-19 training images. Since the considered CAE models require unsupervised learning, only the set Y is utilized for their training. However, for both BiGAN and CycleGAN models that rely on weakly supervised learning [55, 56], a second set X composed by 1000 input features (also referred to as latent feature maps) has been generated for their training. Specifically, according to [55], each training input feature $x \in X$ has been generated by randomly sampling (in an independent and identically distributed way) from a continuous probability density function, which is evenly distributed over the interval $[-100, 100]$. The random procedure adopted for generating the training features assures that the elements of the resulting training sets X and Y are unpaired, as required by the weakly supervised training of BiGAN and CycleGAN models [55, 56]. In this regard, we anticipate also that, although, in our tests, the feature maps $\{\hat{x}\}$ extracted by each model have the same size of the corresponding input feature maps $\{x\}$, nevertheless, their size varies from model to model (see the 6th column of

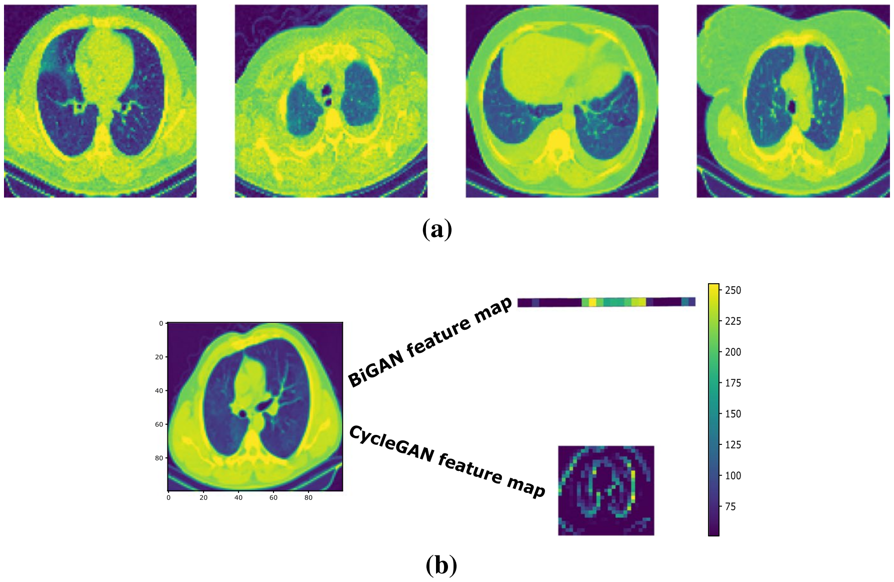


Fig. 1 **a** Four representative samples of lung CT scans from the training sets of size (100×100). **b** Representative samples of lung CT feature maps extracted by the implemented BiGANs and CycleGAN

Table 8). For illustrative purposes, Fig. 1b reports two feature maps extracted from the implemented BiGAN and CycleGAN models.

Finally, we point out that CT scans for testing have been randomly sampled from two data sets [58, 59], which embrace: (i) 500 CT slices of COVID-19 images (different from those used for the training); and, (ii) 500 additional CT scans, which cover normal cases, pneumonia cases and three types of lung cancer (namely, adenocarcinoma, large-cell carcinoma and squamous-cell carcinoma).

2.2 The considered encoder-equipped GAN models

In order to perform classification based on the compressed versions of images (feature representations), BiGANs and CycleGANs are of interest, because they allow to efficiently extract the encoded features of the target class. In the following, we shortly present the implemented models.

2.2.1 Cross-entropy BiGANs for feature extraction

BiGANs offer a framework for weakly supervised feature learning. A BiGAN includes a GAN's generator G, and an encoder $\mathcal{E}$, which maps input data $y \in Y$ (i.e., COVID-19 images, in our framework) to feature representations $\mathcal{E} \equiv \hat{x}$ [43]. The BiGAN discriminator, D, discriminates not only in the data space (i.e., y-vs.-$G(x)$), but jointly in the latent and data spaces (i.e., $\{y, \mathcal{E}(y)\}$-vs.-$\{x, G(x)\}$) versus $(G(z); z)$, where the latent component is either an encoder output: $\mathcal{E}(y)$ or a generator input: x (see Fig. 2).

The BiGAN encoder, $\mathcal{E}$, aims to learn to invert the mapping performed by the generator G [55]. Neither module can directly communicate with the other; the encoder cannot see the generator outputs and the generator cannot see the encoder outputs.

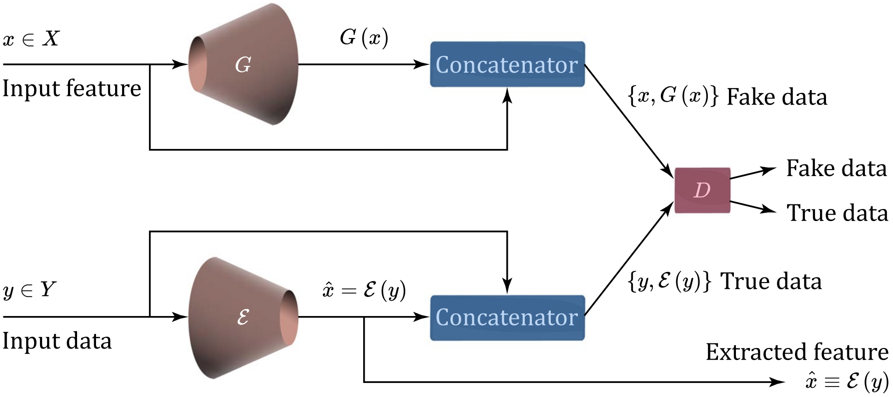


Fig. 2 Implemented training scheme of BiGAN [55] for feature extraction. x: input feature; y: input data; X: feature space; Y: data space; G: Generator; $\mathcal{E}$: Encoder; D: Discriminator; $\hat{x}$: predicted and extracted feature

The final goal of both encoder and generator is to fool the BiGAN discriminator, D [55]. For this purpose, the BiGAN encoder learns to predict features $\hat{x}$ from input data y. Since previous work on BiGAN proved that the extracted features capture semantic attributes of the input data, we argue that a trained BiGAN encoder could provide useful feature representation for related semantic tasks. Toward this end, the BiGAN negative-log-likelihood training objective is defined as follows (see [55] for major details):

$$\begin{aligned}\min_{G,\mathcal{E}} \max_{D} V(D,\mathcal{E},G) = & E_{y\sim p_y} \underbrace{E_{x\sim p_{\mathcal{E}}(.|y)}\left[\log D(x,y)\right]}_{\log D(y,\mathcal{E}(y))} \\ & + E_{x\sim p_x} \underbrace{E_{y\sim p_G(.|x)}\left[1-\log D(x,y)\right]}_{\log(1-D(G(x),x))}.\end{aligned} \tag{1}$$

While BiGANs retain many properties of GANs, they also guarantee that G and $\mathcal{E}$ are each other's inverse at the global optimum. BiGAN training is carried out by using an optimizer for training the parameters θ_D, θ_G, and $\theta_{\mathcal{E}}$ of modules D, G and $\mathcal{E}$, respectively. Training consists of performing one or more steps in the positive gradient direction to update the discriminator parameters θ_D. A step in the negative gradient direction is, then, performed, in order to update the encoder and generator parameters $\theta_{\mathcal{E}}$ and θ_G. In the following sections, we refer to the BiGAN trained according to Eq. (1) as Cross-Entropy BiGAN (CE-BiGAN). The architecture of the actually implemented BiGAN is detailed in Table 2. In our tests, the size of the extracted latent vector $\mathcal{E}(y)$ in Fig. 2 is set to 1024. All the activation functions are leaky ReLUs with slope of 0.1, barring the last layer of the generator, in which the hyperbolic tangent activation function is used (see Table 2).

2.2.2 Least-squares BiGANs for feature extraction

Least-squares generative adversarial networks (LSGANs) adopt the least squares loss function for training [60]. The authors of [60] point out two advantages of LSGANs over standard CE-GANs. First, LSGANs are capable of generating images of higher-quality than CE-GANs. Second, LSGANs also exhibit more stable performance during the learning process. In fact, since a CE-GAN discriminator typically adopts the sigmoid cross-entropy loss function, when the generator is updated, vanishing gradient may happen for samples on the correct side of the decision boundary, which are still far from the real data [60]. LSGANs attempt to bypass this problem by using the following least squares-based training loss function:

$$\begin{aligned} & \min_{G} V_{LSGAN}(G) \triangleq \frac{1}{2} E_{x\sim p_x}\left[(D(G(x),x)-c)^2\right], \\ & [2ex] \min_{D,\mathcal{E}} V_{LSGAN}(D,\mathcal{E}) \triangleq \frac{1}{2}\Big\{ E_{y\sim p_y}\left[(D(y,\mathcal{E}(y))-a)^2\right] \\ & \qquad + E_{x\sim p_x}\left[(D(G(x),x)-b)^2\right]\Big\},\end{aligned} \tag{2}$$

where a and b are the labels for true data and fakes, while c indicates the value that G wants D to believe for fake data [60]. As suggested in [60], in our test, we set

Table 2 The implemented BiGAN architecture. Conv: Convolution; ConvTr: Transposed Convolution; BN: Batch Normalization; LR: Leaky ReLU; DR: Dropout

(a) Generator		
Noise	1024×1	
Dense+BN+LR	80000×1	
Reshape	$25 \times 25 \times 128$	
ConvTr+BN+LR	$50 \times 50 \times 128$	
ConvTr+BN+LR	$50 \times 50 \times 128$	
ConvTr+BN+LR	$100 \times 100 \times 256$	
ConvTr+BN+Tanh	$100 \times 100 \times 1$	
(b) Encoder		
Conv+BN+LR	$100 \times 100 \times 1$	
Conv+BN+LR	$100 \times 100 \times 256$	
Conv+BN+LR	$50 \times 50 \times 128$	
Conv+BN+LR	$50 \times 50 \times 128$	
Conv+BN+LR	$25 \times 25 \times 128$	
Flatten	80000×1	
Dense+BN+LR	1024×1	
(c) Discriminator		
	Input1	$100 \times 100 \times 1$
	Conv+LR+DR	$100 \times 100 \times 256$
	Conv+LR+DR	$50 \times 50 \times 128$
	Conv+LR+DR	$25 \times 25 \times 128$
	Conv+LR+DR	$25 \times 25 \times 128$
	Flatten	80000×1
	Input2	1024×1
	Dense+DR	512×1

Table 2 (continued)

(c) Discriminator		
	Dense	512×1
	Output	512×1
	Concatenate	Flatten + Output
	Dense+DR	1024×1
CE-BiGAN	Dense+Sigmoid	1
W-BiGAN	Dense+Linear	1
LS-BiGAN	Dense+Linear	1

$a = c = 1$, and $b = 0$ where 0-1 is binary labeling scheme used for fake-true data. We apply the loss function of Eq. (2) together with the linear activation function in the last layer of the discriminator of Fig. 2. The architecture of the implemented BiGAN is still the one of Table 2.

2.2.3 Wasserstein BiGANs for feature extraction

Wasserstein GANs (WGANs) [61] generate loss functions with better characteristics than the cross-entropy original GANs by using the Wasserstein distance. For this purpose, the authors of [61] impose weight clipping by requiring that the discriminator (called *critic* in their paper) falls in the 1-Lipschitz space. Accordingly, the loss function of a Wasserstein BiGAN (W-BiGAN) is defined as in [61]:

$$\mathcal{L}_{\text{W-BiGAN}} = \frac{1}{M} \sum_{y \sim p_y} D(y, \mathcal{E}(y)) - \frac{1}{M} \sum_{x \sim p_x} D(G(x), x), \tag{3}$$

where $M \geq 1$ is the number of terms in each summation. Ad pointed out in [62], the r.h.s. of (3) provides, indeed, a reasonable good computable approximation of the actual Wasserstein distance. Unlike the original BiGAN where D is a 0/1 classifier estimating the *a posteriori* probability that its input is a true data, in the Wasserstein BiGAN (W-BiGAN), D is a regressor, which estimates the *trueness* score of its input. In terms of implementation, the scalar output of D in the original BiGAN uses the sigmoid nonlinearity, while that of the W-BiGAN is linear. The Wasserstein loss in Eq. (3) is the difference of the trueness scores of true and fake samples. D is trained to maximize this difference, while G is trained to minimize it. D wants that its output: $D(y, \mathcal{E}(y)$ is higher for true samples y than for the generated fake samples: $D(G(x), x)$, while G aims at the opposite. Due to the interactions between weight constraints and cost function, WGAN optimization process may result in either vanishing or exploding gradients if the clipping threshold calibration is not suitably tuned [62]. After several validation trials, we set the weight clipping value to 0.01 and normalize the norm of error gradient vector to 10. The same architecture of BiGAN of Table 2 is utilized under the training loss function in Eq. (3), with the linear activation function in the last layer of the W-BiGAN discriminator.

2.2.4 CycleGANs for feature extraction

An input image that is transformed by a CycleGAN [56] can retain fine details, so to closely reproduce the structure of the input image. CycleGAN explores the unpaired style transfer paradigm, in which the model attempts to learn stylistic differences between sources and targets without explicitly pairing input to output [63]. As sketched in Fig. 3, a CycleGAN has two generators, G and $\mathcal{E}$, such that $G : X \longrightarrow Y$ and $\mathcal{E} : Y \longrightarrow X$. Ideally, G and $\mathcal{E}$ should be the inverse of each other, so to implement one-to-one bijection. The authors of [56] train simultaneously both the generators G and $\mathcal{E}$ under both adversarial and cycle consistency losses, so to encourage $\mathcal{E}(G(x)) \cong x$ and $G(\mathcal{E}(y)) \cong y$. A CycleGAN is typically equipped with two discriminators D_G and $D_{\mathcal{E}}$ which are paired to the corresponding generators G and $\mathcal{E}$,

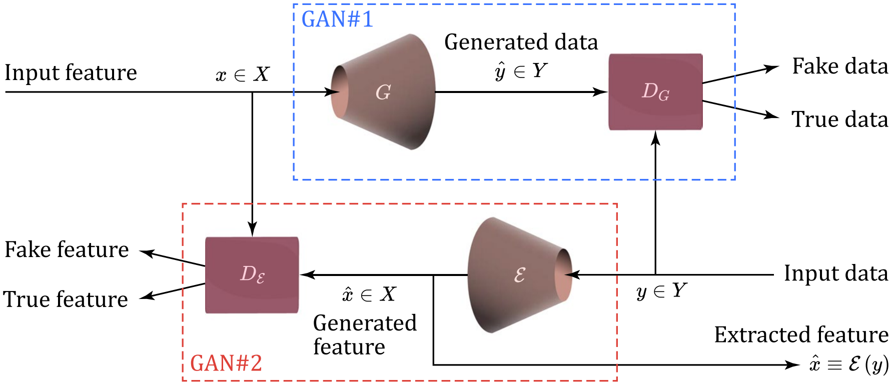


Fig. 3 Implemented scheme of CycleGAN for feature extraction. x: Input feature; y: Input data; $\hat{x}$: Generated and extracted feature; $\hat{y}$: Generated data; X: Feature space; Y: Data space; G: Generator; $\mathcal{E}$: Encoder; D_G: Generator's discriminator; $D_{\mathcal{E}}$: Encoder's discriminator

respectively. In [56], it is argued that a pair of generators/discriminators could learn the best possible translation from the source domain Y (or X) to the target domain X (or Y). The overall cycle consistency loss $\mathcal{L}_{\text{Cyc}}$ ensures that the reconstruction of the original input from the generated output is as close as possible, and it is defined as in [56]:

$$\mathcal{L}_{Cyc}(G,\mathcal{E}) = E_{x\sim p_x}\|\mathcal{E}(G(x)) - x\|_1 + E_{y\sim p_y}\|G(\mathcal{E}(y)) - y\|_1. \tag{4}$$

Afterward, the overall objective of a CycleGAN is a weighted sum of the adversarial losses: $\mathcal{L}_{\text{GAN1}}$ and $\mathcal{L}_{\text{GAN2}}$ and the cycle consistency loss $\mathcal{L}_{\text{Cyc}}$, and, then, it reads as in:

$$\mathcal{L}_{\text{CycleGAN}}\left(G,\mathcal{E},D_G,D_{\mathcal{E}}\right) \triangleq \mathcal{L}_{\text{GAN1}}\left(G,D_G\right) + \mathcal{L}_{\text{GAN2}}\left(\mathcal{E},D_{\mathcal{E}}\right) + \lambda\,\mathcal{L}_{\text{Cyc}}(G,\mathcal{E}). \tag{5}$$

In our tests, $\lambda = 0.1$ and the Wasserstein loss function is employed to implement both the adversarial losses $\mathcal{L}_{\text{GAN1}}$ and $\mathcal{L}_{\text{GAN2}}$ in Eq. (5). The implemented CycleGAN is sketched in Fig. 3. In this regard, we stress that we use it for feature extraction (see Fig. 3). The size of the extracted features is reported in the 6th column of Table 8.

2.3 The pursued KDE-based inference approach

In order to estimate the probability density function (PDF) of the extracted hidden features, generated by (previously described) GAN-based models, the first step is to choose among parametric-vs.-non-parametric methods. Due to the fact that we have no *a priori* information about the actual shape of the PDF and we want to avoid bias effects, we choose a non-parametric estimate. For this purpose, we select the kernel density estimation (KDE) method due to its efficiency and expected performance [64]. To describe the KDE, we first illustrate it for the simple case of a univariate

PDF. Hence, let us consider a set of n real numbers: x_i for $i = 1, ..., n$, drawn from a (hidden) Random Variable (RV) $\boldsymbol{X}$, which possess an unknown PDF, $f_X(x)$, to be estimated. Hence, the KDE estimate $\bar{f}_x(x)$ of $f_x(x)$ is defined as:

$$\bar{f}_X(x) \triangleq \frac{1}{\alpha} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{6}$$

The constant α is a normalization factor, which guarantees that the area under the curve $\bar{f}_X(x), x \in \mathbb{R}$, is unit valued. The kernel function, $K(.)$, is used as an interpolating function to build the PDF estimate. Although different kernels can be used, according to [64], we consider the Gaussian one, i.e., $K(x) = e^{-x^2}$. The parameter h in Eq. (6) is the kernel bandwidth, which is used to set the width of the kernel. It controls the size of the receptive field of the kernel. Since our inference method is based on the evaluation of the distances between actual and target PDFs, we have numerically ascertained that the impact of h is minor. Hence, we set the bandwidth to the unit.

The target COVID-PDF is evaluated by applying Eq. (6) to the average of all the extracted feature vectors obtained by the encoders of the considered architectures for all the training images.

2.4 Exploiting hidden features for test classification

After training on COVID-19 through the BiGANs and CycleGAN, we evaluate the proposed classification method. Using the procedure shown in Fig. 4, we classify each test image. To this end, we only deal with the encoders of BiGANs and the first generator's encoder of CycleGAN. In order to accomplish this, each COVID-19 test image is fed to the trained encoder and its corresponding hidden feature vector is extracted. After computing the PDF of the test feature vector through KDE, the distance d between the target and test PDFs is evaluated and given as input to a binary threshold detector (see the lost block of Fig. 4). This last generates the final COVID/non-COVID decision on the corresponding input image.

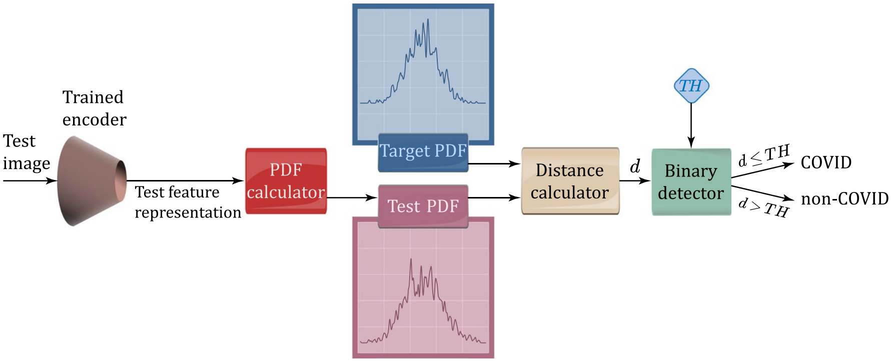


Fig. 4 Proposed inference mechanism for binary classification of test images

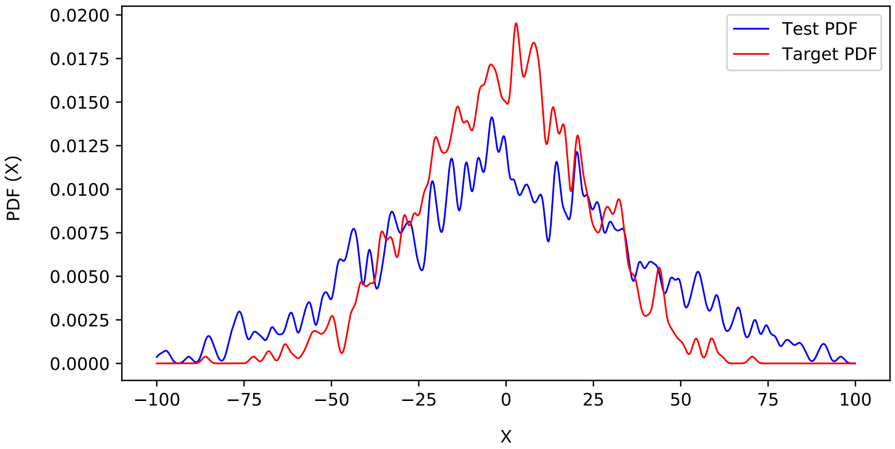


Fig. 5 Instances of target and test PDFs

Table 3 Considered inter-PDF distances

Considered metric distance	Formula
Euclidean	$\|\mathbf{P}-\mathbf{Q}\|_2$
Kullback-Leibler divergence	$\sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right)$
Correlation	$1-\frac{(\mathbf{P}-\bar{\mathbf{P}})^T(\mathbf{Q}-\bar{\mathbf{Q}})}{\|(\mathbf{P}-\bar{\mathbf{P}})\|_2\,\|(\mathbf{Q}-\bar{\mathbf{Q}})\|_2}$
Jensen-Shannon	$\sqrt{\frac{KL(\mathbf{P}\|\mathbf{M})+KL(\mathbf{Q}\|\mathbf{M})}{2}}$

Figure 5 shows two examples of attained target and test PDFs.

Used distance metrics: The target COVID-PDF and the test PDF are compared by using a suitable distance in the latent space. In order to formally introduce the considered inter-PDF distances, let $\mathbf{P}$ and $\mathbf{Q}$ be two equal-size probability column vectors and let: $\mathbf{M} \triangleq (\mathbf{P}+\mathbf{Q})/2$ be the corresponding mean distribution vector. Hence, the considered Euclidean, KL, Correlation and Jensen-Shannon distances are formally defined in Table 3, where p_i (resp. q_i) indicates the i-th entry of $\mathbf{P}$ (resp. $\mathbf{Q}$) and the T superscript means vector transposition.

Setting of the decision threshold: The decision threshold for each considered distance is set by evaluating the PDFs of all training images. Then, we numerically calculate the distance between the target COVID-PDF and each training image PDF and set the threshold TH to the obtained maximum distance value. So doing, the attained value of the threshold is automatically tuned to the statistical properties of both the underlying target PDF and used distance metric. In this regard, we anticipate that, in our tests of Sect. 3, the (numerically evaluated) values of the tuned decision thresholds typically range from 0.06 to 0.6.

Table 4 Main outcomes in binary classification

Taxonomy	Description
True Positive (TP)	COVID-19 image classified as COVID-19
True Negative (TN)	Non-COVID-19 image classified as non-COVID-19
False Positive (FP)	Non-COVID-19 image classified as COVID-19
False Negative (FN)	COVID-19 image classified as non-COVID-19

Table 5 Performance metrics for binary classification [65]

Performance index	Formula
Recall	$TP/(TP + FN)$
Precision	$TP/(TP + FP)$
F-score	$2\,TP/(2\,TP + FP + FN)$
Accuracy	$(TP + TN)/(TP + FN + FP + TN)$

3 Comparative numerical results and discussion

The main goal of this section is twofold. First, after describing the experimental setup and the adopted performance indexes, we discuss the sensitivity of the training and test performance of the implemented BiGAN and CycleGAN models on the considered training loss functions and inter-PDF distance metrics. Second, we present the accuracy-vs.-test time-vs.-training time performance of the implemented BiGAN and CycleGAN models and, then, compare them against the corresponding ones of the CAE-based models recently presented in [38] and [57].

3.1 Considered performance metrics

The considered performance metrics for the carried out binary classification tasks are based on the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) assignments. The meaning of these outcomes is detailed in Table 4. They can be represented in a compact form as the four elements of the resulting confusion matrix [65].

The main performance metrics can be derived by a combination of these items [65]. In the following sections, we consider accuracy, recall, precision, F1-score, area under the receiver operating characteristic (ROC) curve (AUC) as affiliated performance indexes. Formal definitions of these indexes are given in Table 5.

3.2 Experimental setup

All the numerical tests have been carried out on a PC equipped with: (i) an AMD Ryzen 9 5900X 12-Core 3.7 GHz processor; (ii) two GeForce RTX 3080 graphics cards; and (iii) 128 GB RAM.

The iterative solving algorithm used for the training of the implemented CE-BiGAN, LS-BiGAN and LS-CycleGAN models is Adam [66], while the W-BiGAN is trained by using the RMSprop solver with clipping threshold set to 0.01 [61]. The hyper-parameters of all implemented solvers have been optimized through validation trials, and their main optimized values are reported in Table 6. Mini-batches of size of 16 have been utilized for model training under all implemented solvers.

3.3 Comparison of the simulated training loss curves

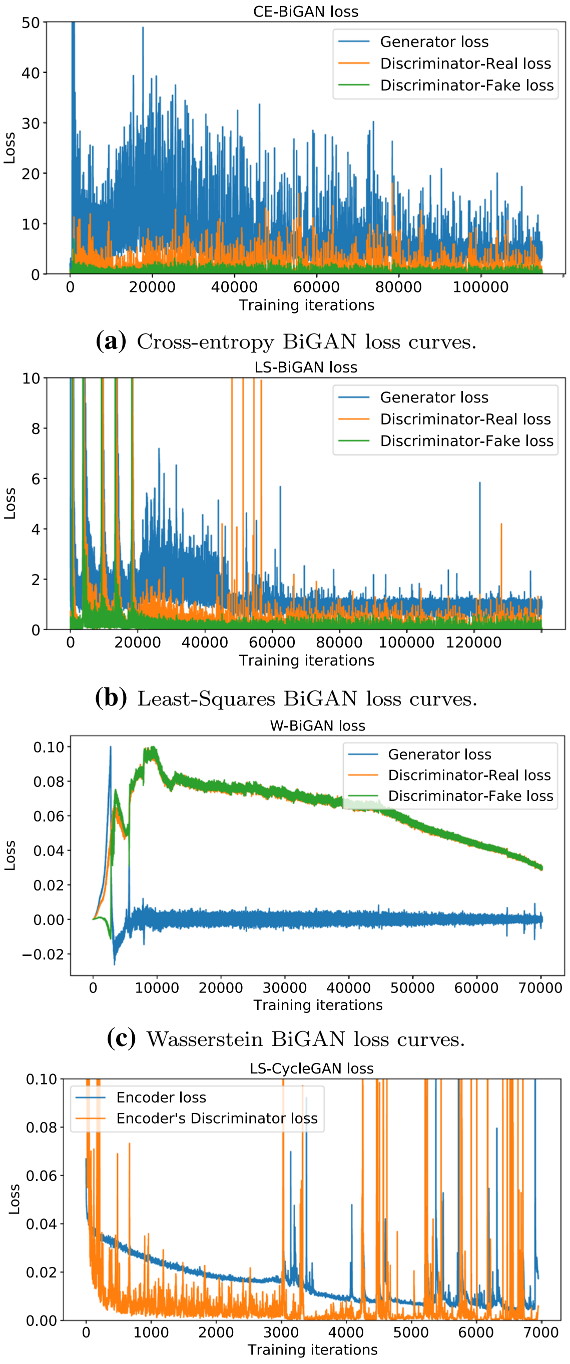
According to [55, 60, 61], a training iteration of each implemented BiGAN and CycleGAN model embraces $m \geq 1$ gradient-based steps for the optimization of the underlying discriminators, which are followed by a single step for the optimization of the corresponding generators. We have numerically ascertained that, in our framework, $m = 1$ (resp., $m = 5$) is suitable for the training of the CE-BiGAN, LS-BiGAN and LS-CycleGAN models (resp., for the training of the W-BiGAN model).

The attained loss curves are reported in Fig. 6. Regarding the BiGAN model, a comparative examination of the training curves of Figs. 6a–c points out that the Wasserstein (resp., Cross-Entropy) loss function gives rise to the most (resp., less) stable behavior during the overall training phase, with the behavior of the least-squares loss function falling somewhat in the middle. This conclusion is also supported by the following additional two remarks. First, a comparative view of the entries in the second column of Table 11 unveils that the number of training iterations needed for achieving the convergence is the highest (resp., the lowest) one for the CE-BiGAN (resp., the W-BiGAN), with the ranking of the LS-BiGAN still falling in the middle. In detail, as it could be expected, the two discriminator losses of Fig. 6c nearly overlap, so that the resulting generator loss fluctuates around zero and asymptotically vanishes. Second, the results reported in Table 7 show that, under each checked distance metric, the corresponding test accuracy of the W-BiGAN is the highest one, although the relative gaps with respect to the competing CE-BiGAN and LS-BiGAN models are not so impressive. However, we have numerically ascertained that, at least under the considered training dataset, the least-squares loss function gives rise to the most stable behavior in the training phase of the implemented CycleGAN (see the plots of Fig. 6d). So, in the following sections, we directly focus on the LS-CycleGAN model.

Table 6 Implemented solvers for the optimization and related hyper-parameter tuning. LR: Learning Rate; WclipV: Weight clip Value; GclipN: Gradient clip Norm

Model	Solver	LR	β_1	WclipV	GclipN
CE-BiGAN	Adam	10^{-4}	0.1	–	–
W-BiGAN	RMSprop	10^{-5}	–	0.01	10
LS-BiGAN	Adam	10^{-4}	0.1	–	10
LS-CycleGAN	Adam	10^{-4}	0.35	–	–

Fig. 6 Training loss curves of the considered approaches



(a) Cross-entropy BiGAN loss curves.

(b) Least-Squares BiGAN loss curves.

(c) Wasserstein BiGAN loss curves.

(d) Least-Squares CycleGAN loss curves.

3.4 Performance robustness with respect to the distance metrics

The impact of the considered distance metrics on the performance indexes of Table 5 in the test phase may be evaluated through a comparative view of the entries of Table 7. In this regard, three main conclusions may be drawn. First, the test performance of all models is quite robust with respect to the choice of the distance metric used for implementing the classifier of Fig. 4. Specifically, the resulting accuracy gaps over the full spectrum of checked model-vs.-distance settings are, indeed, limited up to 5.7%. Second, the accuracies of the CE-BiGAN and LS-BiGAN (resp., W-BiGAN and LS-CycleGAN) models attain their corresponding maxima under the correlation (resp., Jensen-Shannon) distance metric. Third, the highest test accuracy is obtained by the LS-CycleGAN model combined with the Jensen-Shannon distance.

The numerically evaluated distance spectra between the test and target COVID PDFs of the checked models under their corresponding best distance metrics are drawn in Figs. 7, 8, 9 and 10, while the associated confusion matrices are reported in Fig. 11.

The reported distance spectra corroborate the conclusion that the gaps between the accuracy performance of the best-checked models are, indeed, limited, with a slight superiority of the LS-CycleGAN model combined with the Jensen-Shannon distance (see Fig. 11).

This conclusion is further supported by the ROC curves of Fig. 12 and the associated AUC values (see the legend of Fig. 12). These curves confirm, indeed, that the LS-CycleGAN model combined with the Jensen-Shannon distance metric (resp., the LS-BiGAN model combined with the Correlation distance metric) attains the highest (resp., lowest) AUC value of 0.992 (resp., 0.977).

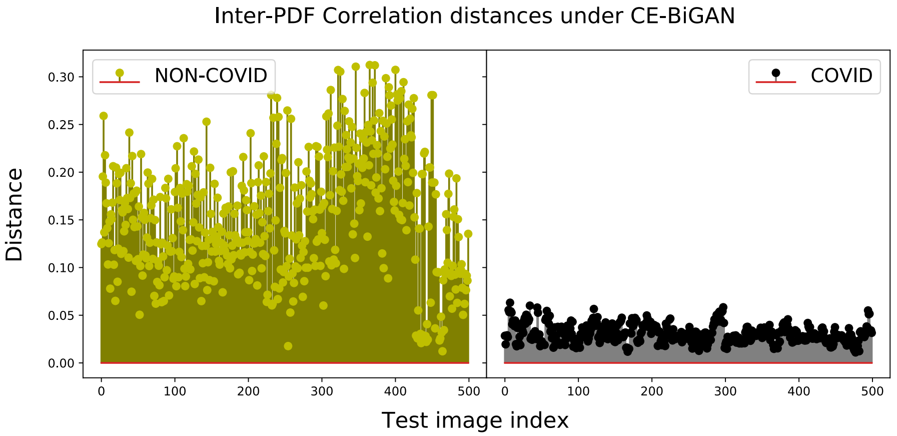


Fig. 7 Correlation distances between test-PDFs and COVID-PDF of CE-BiGAN; Threshold: $TH = 0.06$

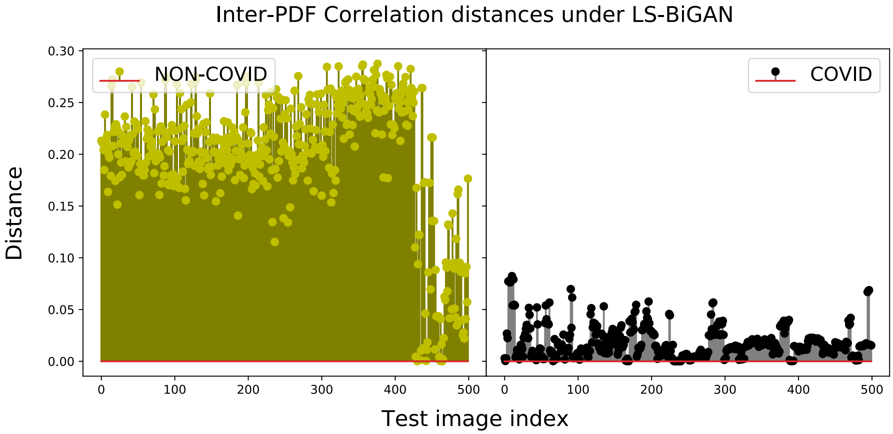


Fig. 8 Correlation distances between test-PDFs and COVID-PDF under the LS-BiGAN model; Threshold: $TH = 0.08$

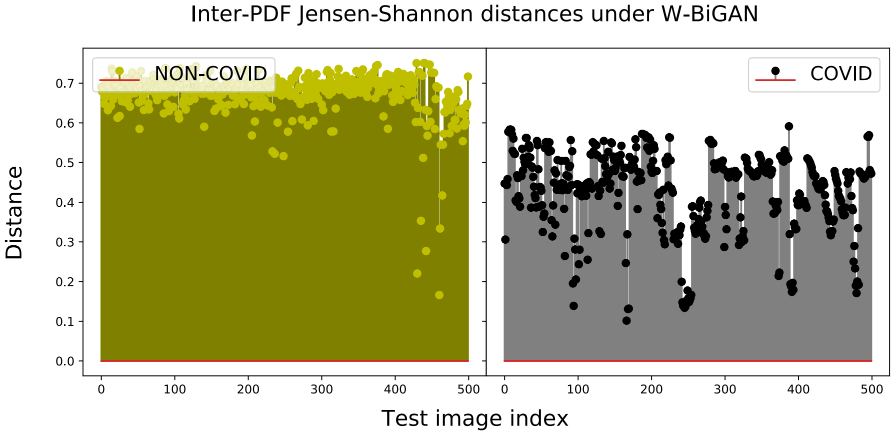


Fig. 9 Jensen-Shannon distances between test-PDFs and COVID-PDF under the W-BiGAN model; Threshold: $TH = 0.585$

3.5 Unsupervised-vs.-weakly supervised models: comparative performance

By design, all the considered BiGAN and CycleGAN models require weakly supervised (WS) training (see Sect. 2.1). Hence, it could be of interest to compare their implementation complexity-vs.-training time-vs.- test time-vs.-test accuracy trade-offs against the corresponding ones of the companion models in [38, 57], which have been recently developed in the literature for COVID-19 detection/classification. Like the considered BiGANs and CycleGANs, even the models developed in [38, 57] rely on suitably extracted hidden features for performing distance-based classification. However, unlike the here considered GAN-based models, both the

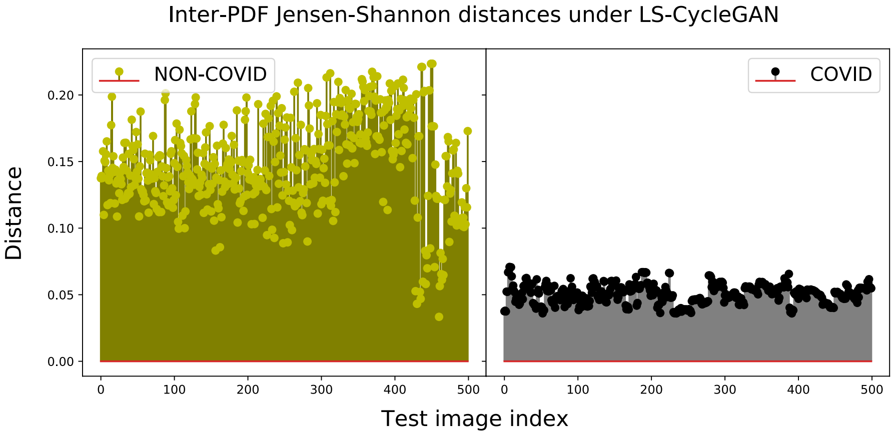
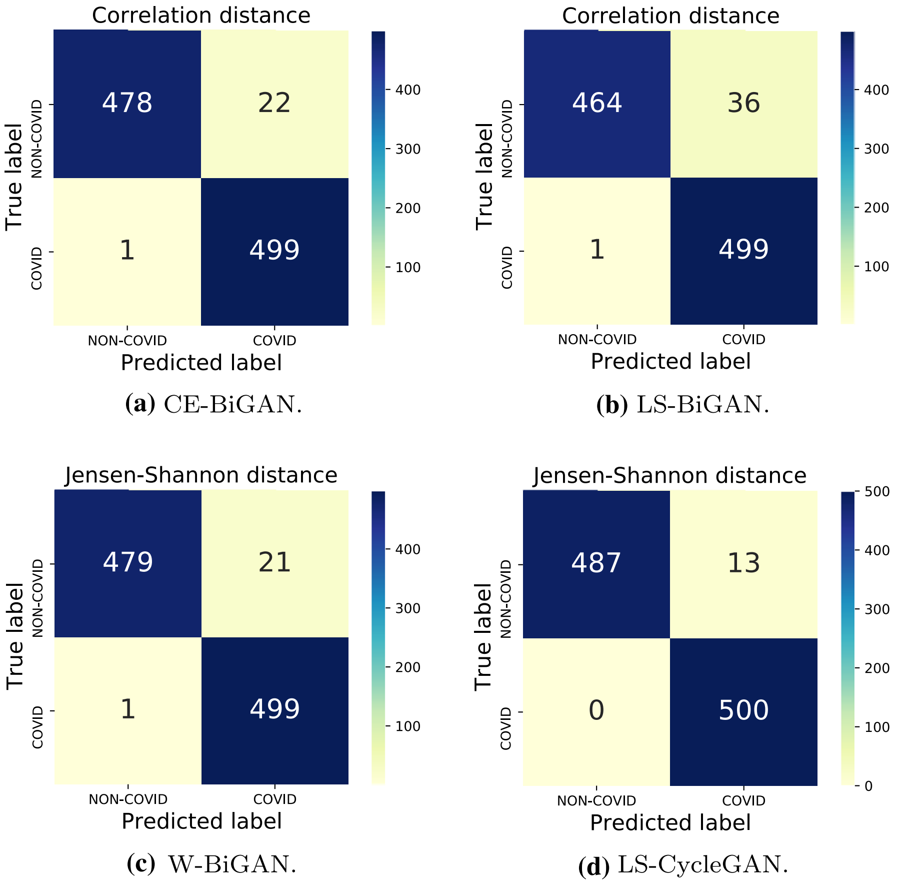


Fig. 10 Jensen-Shannon distances between test-PDFs and COVID-PDF under the LS-CycleGAN model; Threshold: $TH = 0.07$



(a) CE-BiGAN.

(b) LS-BiGAN.

(c) W-BiGAN.

(d) LS-CycleGAN.

Fig. 11 Confusion matrices for the best setting of model/distance metric

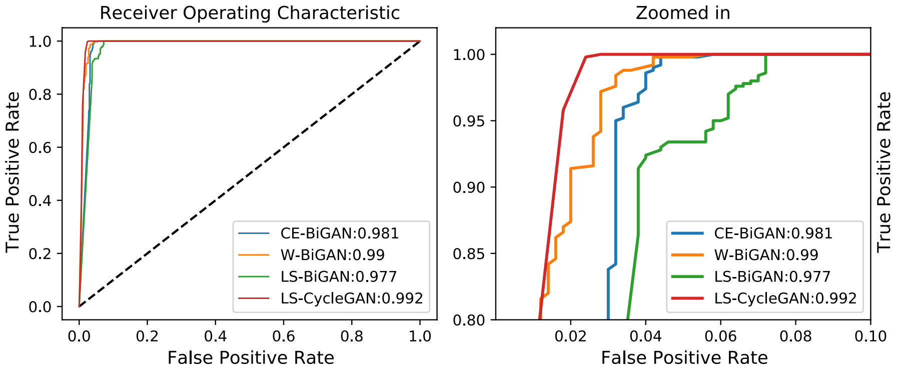


Fig. 12 ROCs of the tested models under the corresponding best distance metrics

Table 7 Model performance under different distance metrics

Model	Distance	Accuracy	Precision	Recall	F1-score	Test CTs
CE-BiGAN						
	Euclidean	0.9760	0.9760	0.9765	0.9760	1000
	Correlation	**0.9770**	0.9770	0.9778	0.9770	1000
	KL divergence	0.9740	0.9740	0.9746	0.9740	1000
	Jensen-Shannon	0.9720	0.9720	0.9728	0.9720	1000
LS-BiGAN						
	Euclidean	0.9490	0.9490	0.9492	0.9490	1000
	Correlation	**0.9640**	0.9630	0.9653	0.9630	1000
	KL divergence	0.9560	0.9560	0.9567	0.9560	1000
	Jensen-Shannon	0.9300	0.9300	0.9314	0.9299	1000
W-BiGAN						
	Euclidean	0.9770	0.9780	0.9781	0.9780	1000
	Correlation	0.9770	0.9770	0.0772	0.9770	1000
	KL divergence	0.9770	0.9770	0.9772	0.9770	1000
	Jensen-Shannon	**0.9780**	0.9780	0.9788	0.9780	1000
LS-CycleGAN						
	Euclidean	0.9760	0.9760	0.9766	0.9760	1000
	Correlation	0.9700	0.9700	0.9711	0.9700	1000
	KL divergence	0.9860	0.9860	0.9863	0.9870	1000
	Jensen-Shannon	**0.9870**	0.9870	0.9873	0.9870	1000

Bold indicates the best result

models developed in [38, 57] exploit the encoders of UnSupervised (US)-trained Denoising CAEs (DCAEs) to extract suitable hidden features from COVID-19 input images. Shortly, the extracted hidden features are utilized in [38] for building up suitable target and test histograms, while they are used in [57] for estimating the underlying test and target PDFs. Hereinafter, we refer to the model in [38] (resp.,

Table 8 Implemented unsupervised and weakly supervised models for COVID-19 detection. HB-DCAE: Histogram-Based DCAE [38]; PB-CAE: PDF-Based CAE [57]; US: Un-Supervised; WS: Weakly Supervised; TRIM: Number of TRaining IMages; TSIM: Number of TeSt IMages

Model	Model training	Size of the input images	#TRIM	#TSIM	Size of the extract features	Resulting best test accuracy
HB-DCAE	US	200 × 300	1000	1000	50 × 75 × 64	0.8270
PB-CAE	US	200 × 300	1000	1000	128 × 1	0.7610
CE-BiGAN	WS	100 × 100	1000	1000	1024 × 1	0.9770
W-BiGAN	WS	100 × 100	1000	1000	1024 × 1	0.9780
LS-BiGAN	WS	100 × 100	1000	1000	1024 × 1	0.9640
LS-CycleGAN	WS	100 × 100	1000	1000	25 × 25 × 256	**0.9870**

Bold indicates the best result

[57]) as the Histogram-Based DCAE (HB-DCAE) (resp., Probability density-Based CAE (PB-CAE)).

The middle columns of Table 8 allow us to compare the main operating settings of the considered WS/US models in terms of sizes of the used input images, numbers of utilized training and test images and sizes of the extracted feature maps. A comparative description of their interior architectures and numbers of trainable parameters (i.e., model sizes) is presented in Table 10, where the ×2 factors account for the fact that a CycleGAN is composed, by design, of two generators and two discriminator nets (see Fig. 2).

The corresponding performance of the tested models is measured through numerical evaluation of the resulting test accuracies (see the last column of Table 8), together with the number of required training iterations and associated training and test times (see Table 11). In order to guarantee fair accuracy comparisons, the same number (i.e., 1000) of training and test images is utilized in all tests (see the 4th and 5th columns of Table 8). Furthermore, in order to carry out fair comparisons among the evaluated training times, the following exit condition has been applied in all performed training simulations: The training phase of a model is stopped when the best training accuracy over a window of 30 consecutive iterations improves less

Table 9 Performance evaluation metrics related to some comparisons with other state-of-the-art approaches using the same dataset

Reference	Model	Accuracy	Precision	Recall	F1-score
[34]	AlexNet	0.9060	0.9060	0.9209	0.9052
[34]	VGG16	0.9140	0.9140	0.9266	0.9134
[17]	ResNet50	0.9245	0.9245	0.9359	0.9302
[58]	CovidNet-ct	0.9790	0.9790	0.9714	0.9752
[52, 53]	MERSGAN+	0.9765	0.9820	0.9840	0.9830
[33]	Random forest	0.9004	0.8917	0.9025	0.8971
[18]	AI-system	0.9656	0.9020	0.9710	0.9352

than 0.1% compared to the corresponding best training accuracy attained over the previous iteration window.

Finally, Table 9 shows some comparisons with other state-of-the-art approaches in the case of supervised COVID/Non-COVID classification, using the same dataset. Specifically, we provide comparisons with famous CNN models such as AlexNet [34], VGG16 [34], ResNet50 [17], and CovidNet-CT [58]. In addition, we also consider the MERSGAN+ proposed in [52, 53], which combines a modified enhanced super-resolution GAN with a Siamese capsule network, the random forest approach proposed in [33] for large-scale screening, and the AI-based system exploiting U-Net architectures introduced in [18].

An examination of Table 9 shows that the proposed BiGAN approaches generally outperform the most common supervised classification methods, although the CovidNet-CT [58] and MERSGAN+ [52] ones obtain similar results. On the other hand, the proposed CycleGAN always outperforms all the state-of-the-art approaches.

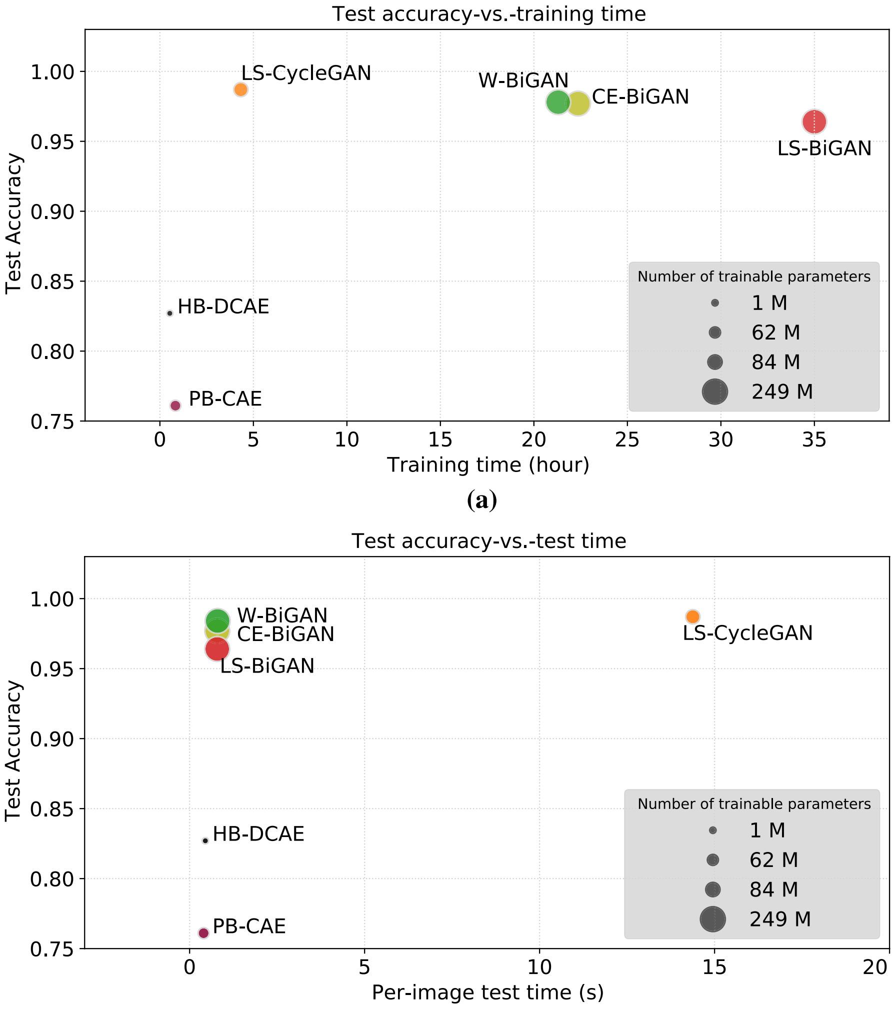
Overall, the results shown in Table 9 compared to those of Table 8 demonstrate the effectiveness of the proposed methods, since, although these are weakly supervised approaches, they are able to perform the same or better than the supervised ones.

3.6 Performance-vs.-computational complexity tradeoff

Figure 13 provides a compact synoptic view of the implementation complexity-vs.-training time-vs.-test time-vs.-accuracy tradeoffs attained by the tested US/WS models. Specifically, in Fig. 13, the diameters of the disk-shaped markers are proportional to the corresponding model sizes (i.e., the number of trainable parameters reported in Table 10).

An examination of Fig. 13 leads to the following insights about the relative merits of the compared models. In terms of test accuracy, the GAN-based models, although present a not negligible training time, outperform the CAE-based ones, with the accuracy of the most performing GAN model (e.g., the LS-CycleGAN one) that is larger than the accuracy of the most performing CAE model (e.g., the HB-DCAE one) of about 16.1% (see also the last column of Table 8). Furthermore, due to their larger learning capability, the GAN-based models are capable to operate on input images whose sizes are smaller than the ones required by the CAE-based models (see the 3rd column of Table 8). We have numerically ascertained that these results are mainly dictated by the US-vs.-WS nature of the tested DL models.

However, in terms of training times, opposite conclusions take place. In fact, as a direct consequence of the major sizes of the GAN-based models compared to the corresponding ones of the CAE-based models, both the number of training iterations and the resulting training times of the implemented BiGAN and CycleGAN models are larger than the corresponding ones of the HB-DCAE and PB-CAE ones. Specifically, the training time of the ‘fastest-to-train’ GAN-based model (e.g., the LS-CycleGAN) is larger than the one of the ‘fastest-to-train’ CAE model (e.g., the HB-DCAE) of about 18.5 times.



(a)

(b)

Fig. 13 **a** Per-model training time versus test accuracy. **b** Per-model test time versus test accuracy

Finally, a similar conclusion holds for the corresponding test times. Specifically, the per-image test times of the 'fastest-to-test' GAN-based models (e.g., the Bi-GAN models) are larger than the one of the 'fastest-to-test' CAE model (e.g., the PB-CAE) of about 80 times (see the last column of Table 11). In this regard, we have numerically ascertained that the achieved test times are mainly dictated by the sizes of the extracted features. This is also the reason why the test time of the LS-CycleGAN is larger than the ones of the BiGAN models (see Fig. 13b).

Table 10 Details and number of trainable parameters of the considered model architectures. BN: Batch-Normalization; Conv: Convolution; ConvTr: Transposed Convolution

	BiGANs	CycleGAN	HB-DCAE[38]	PB-CAE[57]
Generator				
Conv+BN	–	$21 + 24 \quad (\times 2)$	–	–
ConvTr+BN	4+4	$3 + 3 \quad (\times 2)$	–	–
Dense	1	$3 \quad (\times 2)$	–	–
Trainable parameters	82,753,665	$35,264,003 \quad (\times 2)$	–	–
Discriminator				
Conv + BN	4+0	$6 + 6 \quad (\times 2)$	–	–
Dense	4	–	–	–
Trainable parameters	83,826,561	$6,960,321 \quad (\times 2)$	–	–
Encoder				
Conv+BN	4+4	3+3	3+2	3+2
Dense	1	–	–	1
Trainable parameters	82,515,072	374,016	376,640	31,096,768

Table 11 Training/test times of the considered models

Model	Number oftraining iterations	Per-iteration time (s)	Per-image test time (s)
CE-BiGAN	115,000	0.7	0.79
W-BiGAN	70,000	1.08	0.80
LS-BiGAN	140,000	0.9	0.79
LS-CycleGAN	7000	2.23	14.38
HB-DCAE	3125	0.27	0.05
PB-CAE	6250	0.306	0.01

Overall, by considering the complexity-vs.-training time-vs.-test time-vs.-accuracy tradeoff, we can argue that the proposed method can represent a promising approach for the robust classification of the COVID-19 disease from unlabeled CT scans.

4 Conclusion and hints for future research

In this paper, we developed a KDE-based inference method, which leverages the hidden features extracted by BiGANs and CycleGANs for estimating, in the training phase, the (*a priori* unknown) PDF of the CT scans of COVID-19 patients (that is, the target COVID-PDF). Afterward, in the test phase, the distance (in the latent space) between the PDF of each test CT scan and the target COVID-PDF is evaluated, and, then, a tunable binary detector is implemented for generating the COVID/Non-COVID final decisions. We have numerically checked the implementation complexity-vs.-performance trade-offs attained by the designed BiGAN and CycleGAN models under several settings of training loss functions and distance metrics for test classification. In order to better corroborate the obtained numerical results, we have also checked the corresponding implementation complexity-vs.-performance trade-offs of some state-of-the-art competing models, which utilize the encoders of unsupervised-trained CAEs as feature extractors. The comparative analysis of the obtained numerical results supports the final conclusions that: i) the test accuracies of the proposed CycleGAN-based (resp., BiGAN-based) models outperform the corresponding ones of the benchmark CAE-based models of about 16% (resp., 14%); while, ii) the average training times of the tested CAE-based models are lower than the ones of the developed Cycle/BiGAN-based models of about 18–19 times.

The presented results open, indeed, the doors to five main research directions regarding the utilization of Cycle/Bi-GAN-based engines for image classification.

First, recovery of hyperspectral images (i.e., images composed by a number of inter-depending multispectral spatial slices) is an ill-posed (typically, nonconvex) constrained inverse problem, in which high-resolution multiband images must be recovered from their low-resolution (i.e., mixed and/or noise-affected) counterparts [67]. Recently, in [67, 68], supervised-trained CNN-based methods have been developed for unmixing and classification of hyperspectral images. Hence, developing effective Cycle/BiGAN-based models for the weakly supervised recovery/classification of hyperspectral images may be a first research topic of potential interest.

Second, in [68], supervised-trained graph convolutional networks (GCNs) (i.e., CNNs capable to operate on input data described by assigned adjacency graphs) have been designed for hyperspectral image classification. Motivated by the good performance reported in [68], we believe that an interesting research topic could concern the design of BiGAN and CycleGAN models that are capable to operate over graph-structured input data, in which long-range spatial dependence is captured by suitable adjacency matrices.

The recent contribution [69] proposes a CNN-based architecture for the joint extraction and fusion of features from multi-modal input data (i.e., heterogeneous input data that refer to a same object/scene to be classified). The design of novel BiGAN and CycleGAN architectures for multi-modal learning could be a third research line of potential interest.

A further hint for future research arises from the consideration that hyperspectral images are typically represented as data cubes with spatial-spectral information, in which non-negligible inter-data correlation is typically present along the spectral axis. To suitably exploit this correlation, the recent contribution in [70] proposes a new supervised-trained transformer-based DNN model (referred to as *SpectralFormer*) for the reliable classification of hyperspectral images. Hence, an interesting topic could concern the exploitation of BiGANs and CyCleGANs for the design of transformer-based DNN architectures that rely on weakly supervised training for image classification.

Finally, a potential drawback of the developed BiGAN and CycleGAN models is that their training times are quite long (i.e., more than 18 times larger than the corresponding ones of the tested CAE-based models). Hence, how to exploit Cloud/Fog-based [71, 72] virtualized [73] and (possibly) multi-antenna [74, 75] computing architectures for the parallel and distributed training of heavy BiGAN/CycleGAN models in interference-affected broadband wireless domains [76, 77] could be a final research topic of potential interest.

Acknowledgements This work has been supported by the projects: "SoFT: Fog of Social IoT" funded by Sapienza University of Rome Bando 2018 and 2019; "End-to-End Learning for 3D Acoustic Scene Analysis (ELeSA)" funded by Sapienza University of Rome Bando Acquisizione di medie e grandi attrezzature scientifiche 2018; and, "DeepFog – Optimized distributed implementation of Deep Learning models over networked multitier Fog platforms for IoT stream applications" funded by Sapienza University of Rome Bando 2020 and 2021.

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Data availability The datasets generated during and/or analyzed during the current study are available in the COVID-Net repository (https://alexswong.github.io/COVID-Net/), and the Chest CT-Scan images Dataset (https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images).

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Axiaq A, Almohtadi A, Massias SA, Ngemoh D, Harky A (2021) The role of computed tomography scan in the diagnosis of COVID-19 pneumonia. Curr Opin Pulm Med 27(3):163–168. https://doi.org/10.1097/MCP.0000000000000765
2. Islam N, Ebrahimzadeh S, Salameh J-P et al (2021) Thoracic imaging tests for the diagnosis of COVID-19. Cochrane Database Syst Rev 3(CD013639):1–147. https://doi.org/10.1002/14651858.CD013639.pub4
3. Fang Y, Zhang H, Xie J, Lin M, Ying L, Pang P, Ji W (2020) Sensitivity of chest CT for COVID-19: comparison to RT-PCR. Radiology 296(2):E115–E117. https://doi.org/10.1148/radiol.2020200432
4. Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Tao Q, Sun Z, Xia L (2020) Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. Radiology 296(2):E32–E40. https://doi.org/10.1148/radiol.2020200642
5. Xie X, Zhong Z, Zhao W, Zheng C, Wang F, Liu J (2020) Chest CT for typical coronavirus disease 2019 (COVID-19) pneumonia: relationship to negative RT-PCR testing. Radiology 296(2):E41–E45. https://doi.org/10.1148/radiol.2020200343
6. Öztürk Ş, Özkaya U, Barstugan M (2020) Coronavirus (covid-19) classification using deep features fusion and ranking technique. In: Big data analytics and artificial intelligence against COVID-19: innovation vision and approach. Springer International Publishing, Cham, pp 281–295. https://doi.org/10.1007/978-3-030-55258-9_17

7. Jin C, Chen W, Cao Y et al. (2020) Development and evaluation of an AI system for COVID-19 diagnosis. medRxiv, https://doi.org/10.1101/2020.03.20.20039834
8. Sethy PK, Behera SK, Ratha PK, Biswas P (2020) Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. Int J Math Eng Manag Sci 5(4):643–651. https://doi.org/10.33889/IJMEMS.2020.5.4.052
9. Sarv Ahrabi S, Scarpiniti M, Baccarelli E, Momenzadeh A (2021) An accuracy vs. complexity comparison of deep learning architectures for the detection of COVID-19 disease. Computation. https://doi.org/10.3390/computation9010003
10. Toraman S, Alakus TB, Turkoglu I (2020) Convolutional capsnet: a novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. Chaos Solitons Fractals. https://doi.org/10.1016/j.chaos.2020.110122
11. Tan W, Liu P, Li X, Liu Y, Zhou Q, Chen C, Gong Z, Yin X, Zhang Y (2021) Classification of COVID-19 pneumonia from chest CT images based on reconstructed super-resolution images and VGG neural network. Health Inf Sci Syst 9(1):1–12. https://doi.org/10.1007/s13755-021-00140-0
12. Mahmud T, Rahman MA, Fattah SA (2020) CovXNet: a multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. Comput Biol Med. https://doi.org/10.1016/j.compbiomed.2020.103869
13. Heidarian S, Afshar P, Enshaei N et al (2021) COVID-FACT: a fully-automated capsule network-based framework for identification of COVID-19 cases from chest CT scans. Front Artif Intell. https://doi.org/10.3389/frai.2021.598932
14. Song Y, Zheng S, Li L et al (2021) Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. IEEE/ACM Trans Comput Biol Bioinf. https://doi.org/10.1109/TCBB.2021.3065361
15. Yang S, Jiang L, Cao Z, Wang L, Cao J, Feng R, Zhang Z, Xue X, Shi Y, Shan F (2020) Deep learning for detecting corona virus disease 2019 (COVID-19) on high-resolution computed tomography: A pilot study. Ann Transl Med 8(7):450. https://doi.org/10.21037/atm.2020.03.132
16. Loddo A, Pili F, Di Ruberto C (2021) Deep learning for COVID-19 diagnosis from CT images. Appl Sci 11(17). https://doi.org/10.3390/app11178227https://www.mdpi.com/2076-3417/11/17/8227
17. Nneji GU, Cai J, Deng J, Monday HN, James EC, Ukwuoma CC (2022) Multi-channel based image processing scheme for pneumonia identification. Diagnostics. https://doi.org/10.3390/diagnostics12020325
18. Shuo J, Wang B, Xu H, Luo C, Wei L, Zhao W, Hou X, Ma W, Xu Z, Zheng Z et al (2021) AI-assisted CT imaging analysis for COVID-19 screening: Building and deploying a medical AI system in four weeks. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106897
19. Song Y, Zheng S, Li L, Zhang X, Zhang X, Huang Z, Chen J, Wang R, Zhao H, Chong Y, Shen J, Zha Y, Yang Y (2021) Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. IEEE/ACM Trans Comput Biol Bioinf 18(6):2775–2780. https://doi.org/10.1109/TCBB.2021.3065361
20. Rehman N-U, Zia MS, Meraj T, Rauf HT, Damaševičius R, El-Sherbeeny AM, El-Meligy MA (2021) A self-activated cnn approach for multi-class chest-related covid-19 detection. Appl Sci. https://doi.org/10.3390/app11199023
21. Wang S, Kang B, Ma J et al (2021) A deep learning algorithm using CT images to screen for Corona Virus disease (COVID-19). Eur Radiol 31:1–9. https://doi.org/10.1007/s00330-021-07715-1
22. Khan MA, Hussain N, Majid A, Alhaisoni M, Bukhari SA, Kadry S, Nam Y, Zhang Y-D (2021) Classification of positive COVID-19 CT scans using deep learning. Comput Mater Contin 66(3):2923–2938. https://doi.org/10.32604/cmc.2021.013191
23. Rahimzadeh M, Attar A, Sakhaei SM (2021) A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset. Biomed Signal Process Control 68:102588. https://doi.org/10.1016/j.bspc.2021.102588
24. Zhu Z, Xingming Z, Tao G et al (2021) Classification of COVID-19 by compressed chest CT image through deep learning on a large patients cohort. Interdiscip Sci Comput Life Sci 13(1):73–82. https://doi.org/10.1007/s12539-020-00408-1
25. Alom MZ, Rahman MM, Nasrin MS, Taha TM, Asari VK (2020) COVID-MTNet: COVID-19 detection with multi-task deep learning approaches, https://arxiv.org/abs/2004.03747arXiv platform
26. Zhou T, Lu H, Yang Z, Qiu S, Huo B, Dong Y (2021) The ensemble deep learning model for novel COVID-19 on CT images. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106885

27. Rehman A, Naz S, Khan A, Zaib A, Razzak I (2020) Improving coronavirus (COVID-19) diagnosis using deep transfer learning. medRxiv,https://doi.org/10.1101/2020.04.11.20054643.https://www.medrxiv.org/content/10.1101/2020.04.11.20054643v1
28. Hussain E, Hasan M, Rahman MA, Lee I, Tamanna T, Parvez MZ (2021) CoroDet: a deep learning based classification for COVID-19 detection using chest X-ray images. Chaos Solitons Fractals 142(110495):1–12. https://doi.org/10.1016/j.chaos.2020.110495
29. Maghdid HS, Asaad AT, Ghafoor KZ, Sadiq AS, Mirjalili S, Khan MK (2021) Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms. In Multimodal image exploitation and learning 2021, vol 11734 of SPIE, pp 99–110, Florida, United States, SPIE. https://doi.org/10.1117/12.2588672
30. Li C, Yang Y, Liang H, Wu B (2021) Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets. Knowl-based Syst 218(106849):1–9. https://doi.org/10.1016/j.knosys.2021.106849
31. Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi A (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks. Comput Biol Med 121(103795):1–9. https://doi.org/10.1016/j.compbiomed.2020.103795
32. Scarpiniti M, Sarv Ahrabi S, Baccarelli E, Piazzo L, Momenzadeh A (2021) A histogram-based low-complexity approach for the effective detection of COVID-19 disease from CT and X-ray images. Appl Sci 11(19):8867. https://doi.org/10.3390/app11198867
33. Shi F, Xia L, Shan F, Song B, Wu D, Wei Y, Yuan H, Jiang H, He Y, Gao Y, Sui H, Shen D (2021) Large-scale screening to distinguish between COVID-19 and community-acquired pneumonia using infection size-aware classification. Phys Med Biol 66(6):065031. https://doi.org/10.1088/1361-6560/abe838
34. Khan MA, Alhaisoni M, Tariq U, Hussain N, Majid A, Damaševičius R, Maskeliūnas R (2021) Covid-19 case recognition from chest ct images by deep learning, entropy-controlled firefly optimization, and parallel feature fusion. Sensors. https://doi.org/10.3390/s21217286
35. Li D, Zhangjie F, Jun X (2021) Stacked-autoencoder-based model for COVID-19 diagnosis on CT images. Appl Intell 51(5):2805–2817. https://doi.org/10.1007/s10489-020-02002-w
36. Layode OF, Rahman M (2020) A chest X-ray image retrieval system for COVID-19 detection using deep transfer learning and denoising auto encoder. In 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, IEEE. pp 1635–1640. https://doi.org/10.1109/CSCI51800.2020.00301
37. Agarwal C, Khobahi S, Schonfeld D, Soltanalian M (2021) CoroNet: a deep network architecture for enhanced identification of COVID-19 from chest X-ray images. In proceedings of medical imaging 2021: computer-aided diagnosis, vol 11597 of SPIE Medical Imaging, SPIE, pp 484–490. https://doi.org/10.1117/12.2580738
38. Scarpiniti M, Ahrabi SS, Baccarelli E, Piazzo L, Momenzadeh A (2022) A novel unsupervised approach based on the hidden features of deep denoising autoencoders for COVID-19 disease detection. Expert Syst Appl 192:116366. https://doi.org/10.1016/j.eswa.2021.116366. https://www.sciencedirect.com/science/article/pii/S0957417421016614
39. Mansour RF, Escorcia-Gutierrez J, Gamarra M, Gupta D, Castillo O, Kumar S (2021) Unsupervised deep learning based variational autoencoder model for COVID-19 diagnosis and classification. Pattern Recogn Lett 151:267–274. https://doi.org/10.1016/j.patrec.2021.08.018
40. Miao R, Dong X, Xie S-L, Liang Y, Lo S-L (2021) UMLF-COVID: an unsupervised meta-learning model specifically designed to identify X-ray images of COVID-19 patients. BMC Med Imaging 21(1):1–16. https://doi.org/10.1186/s12880-021-00704-2
41. Bashir SM, Wang Y, Khan M, Niu Y (2021) A comprehensive review of deep learning-based single image super-resolution. Peer J Comput Sci 7:e621. https://doi.org/10.7717/peerj-cs.621
42. De S, Bermudez-Edo M, Xu H, Cai Z (2022) Deep generative models in the industrial internet of things: a survey. IEEE Trans Ind Inform. https://doi.org/10.1109/TII.2022.3155656
43. Ian G, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In Advances in neural information processing systems (NIPS 2014), vol 27
44. Sabuhi M, Zhou M, Bezemer C-P, Musilek P (2021) Applications of generative adversarial networks in anomaly detection: a systematic literature review. IEEE Access 9:161003–161029. https://doi.org/10.1109/ACCESS.2021.3131949
45. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR (2020) CovidGAN: Data augmentation using auxiliary classifier GAN for improved Covid-19 detection. IEEE Access 8:91916–91923. https://doi.org/10.1109/ACCESS.2020.2994762

46. Zhang J, Yu L, Chen D, Pan W, Shi C, Niu Y, Yao X, Xiaobin X, Cheng Y (2021) Dense GAN and multi-layer attention based lesion segmentation method for COVID-19 CT images. Biomed Signal Process Control 69:102901. https://doi.org/10.1016/j.bspc.2021.102901
47. Loey M, Smarandache F, Khalifa NEM (2020) Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning. Symmetry. https://doi.org/10.3390/sym12040651
48. Sachdev JS, Bhatnagar N, Bhatnagar R (2021) Deep learning models using auxiliary classifier GAN for Covid-19 detection – a comparative study. In: Hassanien AE, Haqiq A, Tonellato PJ, Bellatreche L, Goundar S, Azar AT, Sabir E, and Bouzidi D (eds), Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2021), Springer International Publishing, Cham, pp 12–23. https://doi.org/10.1007/978-3-030-76346-6_2
49. Menon S, Galita J, Chapman D, Gangopadhyay A, Mangalagiri J, Nguyen P, Yesha Y, Yesha Y, Saboury B (2020) Michael Morris. Generating realistic COVID-19 X-rays with a mean teacher + transfer learning GAN. In 2020 IEEE International Conference on Big Data (Big Data), pp 1216–1225. https://doi.org/10.1109/BigData50022.2020.9377878
50. Motamed S, Rogalla P, Khalvati F (2021) Data augmentation using generative adversarial networks (GANs) for GAN-based detection of pneumonia and COVID-19 in chest X-ray images. Inf Med Unlocked 27:100779. https://doi.org/10.1016/j.imu.2021.100779
51. Asghar U, Arif M, Ejaz K, Vicoveanu D, Izdrui D, Geman O (2022) An improved COVID-19 detection using GAN-based data augmentation and novel QuNet-based classification. Biomed Res Int. https://doi.org/10.1155/2022/8925930
52. Nneji GU, Deng J, Monday HN, Hossin MA, Obiora S, Nahar S, Cai J (2022) Covid-19 identification from low-quality computed tomography using a modified enhanced super-resolution generative adversarial network plus and siamese capsule network. Healthcare. https://doi.org/10.3390/healthcare10020403
53. Nneji GU, Cai J, Monday HN, Hossin MA, Nahar S, Mgbejime GT, Deng J (2022) Fine-tuned siamese network with modified enhanced super-resolution gan plus based on low-quality chest x-ray images for covid-19 identification. Diagnostics. https://doi.org/10.3390/diagnostics12030717
54. Shah PM, Ullah H, Ullah R, Shah D, Wang Y, Islam UL, Gani A, Rodrigues JJ (2022) DC-GAN-based synthetic X-ray images augmentation for increasing the performance of EfficientNet for COVID-19 detection. Expert Syst 39(3):e12823. https://doi.org/10.1111/exsy.12823
55. Donahue J, Krähenbühl P, Darrell T (2017) Adversarial feature learning. In: Preceeding of the 5th International Conference on Learning Representations (ICLR)
56. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/ICCV.2017.244
57. Ahrabi Sima S, Piazzo L, Momenzadeh A, Scarpiniti M, Baccarelli E (2022) Exploiting probability density function of deep convolutional autoencoders' latent space for reliable COVID-19 detection on CT scans. J Supercomput. https://doi.org/10.1007/s11227-022-04349-y
58. Gunraj H, Wang L, Wong A (2020) COVIDNet-CT: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest CT images. Front Med 7:608525. https://doi.org/10.3389/fmed.2020.608525
59. Chest CT-Scan images Dataset, 1st Edition, 2020. https://www.kaggle.com/mohamedhanyyy/chest-ctscan-imagesKaggle Datasets
60. Mao X, Li Q, Xie H, Lau RYK, Wang Z, Paul Smolley S (2017) Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/ICCV.2017.304
61. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning, volume PMLR 70, pp 214–223
62. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of Wasserstein GANs. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), pp 5769–5779
63. Babcock J, Bali R (2021) Generative AI with Python and TensorFlow 2: Create images, text, and music with VAEs, GANs, LSTMs. Transformer models. Packt Publishing Ltd, Birmingham
64. Härdle W, Müller M, Sperlich S, Werwatz A (2004) Nonparametric and semiparametric models, 1st edn. Springer-Verlag, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-17146-8
65. Alpaydin E (2020) Introduction to machine learning. 4th edition, MIT Press, Cambridge US, London UK. https://mitpress.mit.edu/books/introduction-machine-learning-fourth-editionMIT Press

66. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR 2015)
67. Hong D, Yokoya N, Chanussot J, Zhu XX (2019) An augmented linear mixing model to address spectral variability for hyperspectral unmixing. IEEE Trans Image Process 28(4):1923–1938. https://doi.org/10.1109/TIP.2018.2878958
68. Hong D, Gao L, Yao J, Zhang B, Plaza A, Chanussot J (2021) Graph convolutional networks for hyperspectral image classification. IEEE Trans Geosci Remote Sens 59(7):5966–5978. https://doi.org/10.1109/TGRS.2020.3015157
69. Hong D, Gao L, Yokoya N, Yao J, Chanussot J, Qian D, Zhang B (2021) More diverse means better: multimodal deep learning meets remote-sensing imagery classification. IEEE Trans Geosci Remote Sens 59(5):4340–4354. https://doi.org/10.1109/TGRS.2020.3016820
70. Hong D, Han Z, Yao J, Gao L, Zhang B, Plaza A, Chanussot J (2022) Spectralformer: rethinking hyperspectral image classification with transformers. IEEE Trans Geosci Remote Sens 60:1–15. https://doi.org/10.1109/TGRS.2021.3130716
71. Baccarelli E, Naranjo PGV, Shojafar M, Scarpiniti M (2017) Q*: energy and delay-efficient dynamic queue management in TCP/IP virtualized data centers. Comput Commun 102:89–106, https://doi.org/10.1016/j.comcom.2016.12.010. https://www.sciencedirect.com/science/article/pii/S0140366416306892
72. Baccarelli E, Scarpiniti M, Momenzadeh A, Ahrabi SS (2021) Learning-in-the-Fog (LiFo): deep learning meets fog computing for the minimum-energy distributed early-exit of inference in delay-critical IoT realms. IEEE Access, 9:25716–25757, https://doi.org/10.1109/ACCESS.2021.3058021. https://ieeexplore.ieee.org/document/9350277
73. Amendola D, Cordeschi N, Baccarelli E (2016) Bandwidth management vms live migration in wireless fog computing for 5g networks. In: 2016 5th IEEE International Conference on Cloud Networking (Cloudnet), pp 21–26 https://doi.org/10.1109/CloudNet.2016.36
74. Baccarelli E, Biagi M, Pelizzoni C (2005) On the information throughput and optimized power allocation for MIMO wireless systems with imperfect channel estimation. IEEE Trans Signal Process 53(7):2335–2347. https://doi.org/10.1109/TSP.2005.849165
75. Baccarelli E, Biagi M (2004) Power-allocation policy and optimized design of multiple-antenna systems with imperfect channel estimation. IEEE Trans Veh Technol 53(1):136–145. https://doi.org/10.1109/TVT.2003.822025
76. Baccarelli E, Cordeschi N, Polli V (2013) Optimal self-adaptive qos resource management in interference-affected multicast wireless networks. IEEE/ACM Trans Netw 21(6):1750–1759. https://doi.org/10.1109/TNET.2012.2237411
77. Baccarelli E, Biagi M, Bruno R, Conti M, Gregori E (2005) Broadband wireless access networks: a roadmap on emerging trends and standards. In Broadband Services: Business Models and Technologies for Community Networks, Wiley Online Library, chapter 14, pp 215–240. https://doi.org/10.1002/0470022515.ch14

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Sima Sarv Ahrabi[1] · Alireza Momenzadeh[1] · Enzo Baccarelli[1] · Michele Scarpiniti[1] · Lorenzo Piazzo[1]

✉ Michele Scarpiniti
michele.scarpiniti@uniroma1.it

Sima Sarv Ahrabi
sima.sarvahrabi@uniroma1.it

Alireza Momenzadeh
alireza.momenzadeh@uniroma1.it

Enzo Baccarelli
enzo.baccarelli@uniroma1.it

Lorenzo Piazzo
lorenzo.piazzo@uniroma1.it

[1] Department of Information Engineering, Electronics and Telecommunications, Sapienza University or Rome, Via Eudossiana, 18, 00184 Roma, Italy