Uncertainty-Aware Semi-Supervised Method Using Large Unlabeled and Limited Labeled COVID-19 Data

ROOHALAH ALIZADEHSANI, Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Australia
DANIAL SHARIFRAZI, Department of Computer Engineering, School of Technical and Engineering, Shiraz Branch, Islamic Azad University, Iran
NAVID HOSEINI IZADI, Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran
JAVAD HASSANATAJ JOLOUDARI, Department of Computer Engineering, Faculty of Engineering, University of Birjand, Iran
AFSHIN SHOEIBI, Computer Engineering Department, Ferdowsi University of Mashhad, Iran and Faculty of Electrical and Computer Engineering, Biomedical Data Acquisition Lab, K. N. Toosi University of Technology, Iran
JUAN M. GORRIZ, Department of Signal Theory, Networking and Communications, Universidad de Granada, Spain
SADIQ HUSSAIN, System Administrator, Dibrugarh University, India
JUANE ARCO, Department of Signal Theory, Networking and Communications, Universidad de Granada, Spain

This work was partly supported by the MINECO/FEDER under the RTI2018-098913-B100, CV20-45250, and A-TIC-080-UGR18 projects.

Authors’ addresses: R. Alizadehsani, F. Khozeimeh, A. Khoosravi, and S. Nahavandi, Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Geelong 3216, 75 Pigdons Rd, Victoria, Australia; emails: {r.alizadehsani, f.khozeimeh, a.khoosravi, s.nahavandi}@deakin.edu.au; D. Sharifrazi, Department of Computer Engineering, School of Technical and Engineering, Shiraz Branch, Islamic Azad University, Sadra Hwy, Shiraz, Iran; email: danial.sharifrazi@gmail.com; N. H. Izadi, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran; email: navid.hoseini1@ec.iut.ac.ir; J. H. Joloudari, Department of Computer Engineering, Faculty of Engineering, University of Birjand, Daneshgah e Sanati Hwy Birjand, Iran; email: javad.hasannataj@gmail.com; A. Shoeibi, Computer Engineering Department, Ferdowsi University of Mashhad, Vakil Abad Hwy, Mashhad, Iran; Faculty of Electrical and Computer Engineering, Biomedical Data Acquisition Lab, K. N. Toosi University of Technology, Shariati Ave., Tehran, Iran; email: afshin.shoeibi@gmail.com; J. M. Gorriz and J. E. Arco, Department of Signal Theory, Networking and Communications, Universidad de Granada, Granada, Av. del Hospicio, Spain; emails: {j.m.gorriz, j.e.arco}@ugr.es; S. Hussain, System Administrator, Dibrugarh University, Assam, Dibrugarh 786004, India; email: sadiq@dibru.ac.in; Z. A. Sani, Rajale Cardiovascular Medical and Research Center, Iran University of Medical Sciences, Valiasr Ave., Tehran, Iran; Omid Hospital, Iran University of Medical Sciences, Tehran, Iran; email: d.alizadeh.sani@gmail.com; S. M. S. Islam, Institute for Physical Activity and Nutrition, Deakin University, Melbourne, 221 Burwood Highway, Australia; Cardiovascular Division, The George Institute for Global Health, King St, Sydney, Australia; Sydney Medical School, University of Sydney, Fisher Road, Sydney, Australia; email: shariful.islam@deakin.edu.au; U. R. Acharya, Department of Electronics and Computer Engineering, Ngee Ann Polytechnic, 535 Clementi Rd, Singapore, Singapore; Department of Biomedical Engineering, School of Science and Technology, Singapore University of Social Sciences, Singapore, 463 Clementi Rd, Singapore; Department of Bioinformatics and Medical Engineering, Asia University, Liuqeng Rd Taichung, Taiwan; email: rajendra_udyavara_acharya@np.edu.sg.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.
1551-6857/2021/11-ART104 $15.00
https://doi.org/10.1145/3462635
The new coronavirus has caused more than one million deaths and continues to spread rapidly. This virus targets the lungs, causing respiratory distress which can be mild or severe. The X-ray or computed tomography (CT) images of lungs can reveal whether the patient is infected with COVID-19 or not. Many researchers are trying to improve COVID-19 detection using artificial intelligence. Our motivation is to develop an automatic method that can cope with scenarios in which preparing labeled data is time consuming or expensive. In this article, we propose a Semi-supervised Classification using Limited Labeled Data (SCLLD) relying on Sobel edge detection and Generative Adversarial Networks (GANs) to automate the COVID-19 diagnosis. The GAN discriminator output is a probabilistic value which is used for classification in this work. The proposed system is trained using 10,000 CT scans collected from Omid Hospital, whereas a public dataset is also used for validating our system. The proposed method is compared with other state-of-the-art supervised methods such as Gaussian processes. To the best of our knowledge, this is the first time a semi-supervised method for COVID-19 detection is presented. Our system is capable of learning from a mixture of limited labeled and unlabeled data where supervised learners fail due to a lack of sufficient amount of labeled data. Thus, our semi-supervised training method significantly outperforms the supervised training of Convolutional Neural Network (CNN) when labeled training data is scarce. The 95% confidence intervals for our method in terms of accuracy, sensitivity, and specificity are 99.56 ± 0.20%, 99.88 ± 0.24%, and 99.40 ± 0.18%, respectively, whereas intervals for the CNN (trained supervised) are 68.34 ± 4.11%, 91.2 ± 6.15%, and 46.40 ± 5.21%.

CCS Concepts: • Applied computing → Health care information systems;

Additional Key Words and Phrases: Semi-supervised learning, generative adversarial networks, COVID-19, supervised learning, deep learning

ACM Reference format:
Roohallah Alizadehsani, Danial Sharifrazi, Navid Hoseini Izadi, Javad Hassannataj Joloudari, Afsin Shoeibi, Juan M. Gorriz, Sadiq Hussain, Juan E. Arco, Zahra Alizadeh Sani, Fahime Khozeimeh, Abbas Khosravi, Saeid Nahavandi, Sheikh Mohammed Shariful Islam, and U. Rajendra Acharya. 2021. Uncertainty-Aware Semi-Supervised Method Using Large Unlabeled and Limited Labeled COVID-19 Data. ACM Trans. Multimedia Comput. Commun. Appl., 17, 3s, Article 104 (November 2021), 24 pages. https://doi.org/10.1145/3462635

1 INTRODUCTION
Since the beginning of 2020, the coronavirus disease 2019 (COVID-19) has been spreading all over the globe as it is contagious in an unprecedented manner [6]. In severe cases, it may lead to multiple organ failure, acute respiratory distress, heart problems, secondary infections in a comparatively high fraction of patients, and thus may cause deaths [18, 29]. On January 30th, 2020, the World
Health Organization (WHO), announced the outbreak as a "public health emergency of international concern" (PHEIC). The epicenter of the outbreak was the Huanan Seafood Wholesale Market in Wuhan City, Hubei Province, China, and on March 11th, 2020, the WHO declared the COVID-19 a pandemic. Early detection and initiation of treatment in severe cases are vital to deal with the disease and thus mitigate mortality [1, 13].

Reverse-transcription polymerase chain reaction (RT-PCR) is generally used to confirm COVID-19. However, sensitivity and specificity of RT-PCR are not robust enough for an early detection of the disease and for the subsequent election of treatment [5, 19]. A non-invasive imaging technique called computed tomography (CT) can play a crucial role in the identification of COVID-19 signs in lungs [15, 54]. A study using chest CT scans achieved 97% sensitivity (580/601 patients, 95% CI, 95–98%) for COVID-19 detection [5]. As another attempt, a deep transfer learning approach for fast detection of COVID-19 has been proposed with reported accuracy of 95% and sensitivity of 94% [40]. After benchmarking multiple existing deep learning models in [47], VGG-19 has been reported as the superior method with an accuracy of 94.52%. Hence, chest CT scans may be effectively employed in diagnosis and early detection of the disease. However, pulmonary infections due to COVID-19 can be very similar to signs associated with other types of pneumonia [11]. For this reason, identifying the cause of the pneumonia is a highly challenging task.

The access to huge datasets facilitates deep learning tools in the processing of large amounts of unstructured information, allowing high level abstractions with enhanced generalizability and allowing key feature extractions [22]. In the domain of medical imaging, deep learning has shown outstanding performance in automatic feature extraction [28, 48, 51]. Due to the noteworthy performance that deep learning methods achieved in image processing tasks, they can be extremely useful as feature learners for classification tasks [22]. Deep learning has been used to detect and differentiate between viral and bacterial pneumonia in pediatric chest CT scans [49]. Different imaging features of chest CT can also be detected using deep learning [9, 17]. Chest CT scans of positive COVID-19 patients always have a specific radiographic pattern: multifocal patchy consolidation, ground-glass opacities, and/or interstitial changes with a predominantly peripheral distribution [5, 15]. The application of deep learning methods to chest CT scans has been successfully employed for distinguishing pneumonia associated with COVID-19 from other types of viral pneumonia, highlighting its usefulness as a diagnostic tool. This, in turn, leads to an ability to control and manage the pandemic situation [26].

Chest X-ray and CT scans have the potential to detect COVID-19 and the early isolation of patients. As most hospitals are equipped with X-ray, it is the first choice of radiologists. However, chest X-ray images cannot distinguish soft tissues accurately [55]. Chest CT scan can be utilized as an alternative method. As the number of radiologists is scarce and thus not enough in pandemic situations, automatic detection of COVID-19 from chest images is highly desirable. Utilizing deep learning, Li et al. [32] designed a COVID-19 detection model called COVNet by extracting features from chest CT. Other non-pneumonia and community-acquired pneumonia (CAP) CT exams were conducted to evaluate the robustness of the model. Their model could discriminate CAP from other lung diseases, accurately. Wang et al. [57] used the deep learning strategies to derive the graphical features from the CT images of COVID-19 patients. These images show radiographical changes in infected patients. Gozes et al. [23] proposed an automated approach using CT images for quantifying, detecting, and monitoring COVID-19 patients. They used robust 3D and 2D models based on deep learning concepts and integrated them with clinical perspectives. They utilized a multi-center international dataset and generated a corona score using a 3D volume review from thoracic CT features. This score helped the system to compute the evolution of ailment over time.

Several studies have applied new approaches to detect COVID-19 cases using machine learning and deep learning approaches. Hemdan et al. [24] devised a deep learning–based framework
dubbed as COVIDX-Net to aid clinicians to detect COVID-19 from X-ray images. They partitioned the dataset into 80% for training and 20% for testing. Zhang et al. [62] presented an anomaly detection deep technique for reliable and efficient COVID-19 detection. Apostolopoulos et al. [10] analyzed the X-ray images of normal incidents, confirmed COVID-19 cases, and common bacterial pneumonia for automated detection of COVID-19 patients. They applied a convolutional neural network (CNN) with transfer learning. Their model derived biomarkers related to the COVID-19 illness. Butt et al. [59] compared multiple CNN models and devised a deep learning model based on 3D and 2D networks to classify no-infection, influenza viral pneumonia, and COVID-19 samples. Their model successfully differentiated the non-coronavirus and coronavirus cases per thoracic CT records.

Other approaches include Song et al. [53] who devised a CT diagnosis system based on deep learning technology and termed it as DeepPneumonia for the identification of COVID-19 patients. Main lesion features, including ground-glass opacity (GGO) which are the key in the diagnosis, are located by their model. Sethy et al. [46] proposed a deep learning technique based on X-ray radiographs analysis for detection of COVID-19 patients. They implemented a support vector machine classifier using the deep features to discriminate COVID-19 X-ray images from others. Their technique could assist the radiologists in the diagnosis of COVID-19 patients.

The test kits for COVID-19 are limited in hospitals owing to the exponential growth of cases. Therefore, it is crucial to search for a fast alternative to detect such cases in order to confine the spread. Narin et al. [37] presented detection of COVID-19 patients based on Inception-ResNetV2, InceptionV3, and ResNet50 utilizing chest X-ray images. Confusion matrices and ROC analyses are performed with fivefold cross-validation. Shi et al. [50] established a deep learning–oriented CT and clinical features–based prognosis model for assessing the severity of COVID-19 infection. They applied the least absolute shrinkage and selection operator (LASSO), and developed the pneumonia severity index (PSI). Severe patients had higher PSI ($p < 0.001$), percentage of infection (POICT), and mass of infection (MOICT) than non-severe ones. Their model proved its efficacy in the prediction of patients’ severity.

Pervasive demands have arisen to combat COVID-19 pandemic crisis by designing an automated and efficient diagnosis system. Maghdid et al. [36] presented an accurate deep learning tool with fast detection mechanism for COVID-19 cases. Various CT and X-ray images have been integrated to provide a comprehensive and publicly available dataset. The detection technique consists of transfer learning and deep learning. The network is trained using AlexNet and CNN models on the CT and X-rays dataset.

Deep neural networks have great representation power but, their performance heavily relies on the availability of labeled training data. In a case in which the labeled data is limited, the deep networks won’t be able to learn well. However, we can still train the deep networks well by exploiting unlabeled data. The justification behind our hypothesis is the success of transfer learning [39]. In a nutshell, in transfer learning a learner trained for a specific task is modified for another task which bears some similarity to that task. Although the tasks are not the same, the learner trained for can still benefit from the similarity between them, which accelerates the learning. This is the motivation for our two-phase semi-supervised approach. In the first phase, the discriminator is trained to detect valid CT scan images. In the second phase, using the gained expertise from the first phase, the trained discriminator can learn sick and healthy CT images faster. The learning boost is due to the fact that regardless of having COVID or being healthy, each training/test image is a valid CT scan which the discriminator has mastered using unlabeled data.

Considering the above argument, one may be tempted to use non-deep learners in an attempt to reduce the required amount of training data. However, the major drawback of such learners is that they treat the input samples as vectors. To feed images to such non-deep learners, we
are forced to reshape the images into vectors. The reshaping operation destroys the meaningful features that each pixel has to offer relative to its neighboring pixels. Therefore, the application of deep learning based models seems to be inevitable if the features present in image inputs are to be captured properly.

The main contribution of our work is twofold. First, we collected a dataset of lung CT scan images useful for training/evaluation of COVID-19 detection methods. Second, to the best of our knowledge, we are the first group to propose the semi-supervised COVID-19 detection method based on generative adversarial network (GAN) [21] to detect this disease. We have improved this approach by employing Sobel edge detection. Despite being semi-supervised, our method is competitive to its supervised counterparts. This feature is beneficial when labeled data is hard to obtain. Although we have focused on COVID-19 detection, our method is not limited to any specific dataset. The rest of the article is organized as follows: Section 2 provides prerequisites, Section 3 describes our dataset, Section 4 elaborates the proposed method, Section 5 presents the experimental results, and Section 6 concludes the article.

2 METHODS
In this section, the required mathematical concepts are briefly reviewed. First, GAN is reviewed since our method is based on it. The Gaussian Process is also reviewed since it is used during our experiments.

2.1 GANs
Originally proposed by GoodFellow, GAN is a generative model with massive applications. Compared to its predecessors, GAN is capable of generating high-quality images which are vivid and sharp. Basically, GAN is made of two neural networks, namely, Generator (G) and Discriminator (D). The job of the Generator is to produce high-quality images which are called fake samples. The Discriminator must distinguish between the real and fake samples. The two networks compete against each other in a minimax game. That is why we call them adversarial networks. The objective function based on which the two networks are trained is given as [21]

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim p_z(z)} \left[ \log (1 - D(G(z))) \right],$$

where \(p_{\text{data}}(x)\) is the real data (available dataset) distribution, \(p_z(z)\) is the Generator input noise distribution, \(D(x)\) is the Discriminator output, \(z\) is the (Gaussian) noise vector, and \(G(z)\) is the Generator output. As can be seen, the Generator is trying to minimize the objective function in Equation (1), while Discriminator is trying to maximize it. To this end, the following gradients are calculated:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left\{ \log D \left( x^{(i)} \right) + \log \left[ 1 - D \left( G \left( z^{(i)} \right) \right) \right] \right\},$$

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left\{ \log \left[ 1 - D \left( G \left( z^{(i)} \right) \right) \right] \right\},$$

where \(\theta_d\) and \(\theta_g\) are the Discriminator and Generator parameters, respectively [21]. The gradients are computed over a mini-batch of \(m\) samples.

2.2 Gaussian Process
Gaussian Process (GP) is a non-parametric probabilistic model which can be used for regression or classification. GP can be considered as an infinite-dimensional Gaussian distribution which is defined on functions as follows [42].
\textbf{Definition:} \( f \) is a Gaussian process if for any index set \( \{ t^{(i)} \in \mathbb{R}^D, i = 1, \ldots, n \} \), vector \( f(t) = [ f(t^{(1)}), \ldots, f(t^{(n)})]^T \) has a multivariate Gaussian distribution of the form \( f(t) \sim \mathcal{N}(m(t), K(t, t)) \). Each \( f(t^{(i)}) \) is a random variable, \( m(t) \) is the mean function, and \( K(t, t) \) is the covariance matrix of the Gaussian distribution [42]. The mean function is initialized to constant zero \((m(t) = 0)\). Each element of \( K(t, t) \) is the output of a positive definite kernel function \( k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R} \) which receives \( t^{(i)} \) and \( t^{(j)} \) as input. In this article, we use a squared exponential kernel:

\[
k(t^{(i)}, t^{(j)}) = \sigma_f^2 \exp\left(\sum_{d=1}^{D} \frac{-1}{2l_d^2} (t^{(i)}_d - t^{(j)}_d)^2\right) + \sigma_n^2 \delta_{ij}, \quad \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases},
\]

where the kernel function hyper-parameters \( \sigma_f^2, \sigma_n^2 \), and \( l \) are signal variance, noise variance, and length scale, respectively. The hyper-parameters are learned based on the available training data.

2.2.1 GP Classification. In this article, we focus on binary classification of healthy and sick people. Hence, in this section, classification using GP is briefly reviewed. Following the binary classification conventions, the two class labels are denoted as \(-1, +1\). The basic idea behind binary classification using GP is as follows. A prior GP is placed over the latent function \( f(x) \). The output of GP is squashed through a logistic function \( \sigma(.) \) to obtain prior \( \pi(x) \triangleq p(y = +1|x) = \sigma(f(x)) \). Here \( f(x) \) values are not observable and we are not interested in them either. We only care about input vector \( x \) and desired class label \( y \). The sole purpose of \( f(x) \) is to make the model formulation more convenient.

Assuming that the training inputs are aggregated as column vectors in matrix \( X \), and their corresponding labels are expressed as vector \( y \), the GP inference is carried out in two steps:

(1) For a test case \( x_t \), the distribution of the latent variable \( f_t \) is computed as [42]

\[
p(f_t | X, x_t) = \int p(f_t | X, x_{-t}) p(f_{-t} | X, y) df_{-t},
\]

where

\[
p(f | X, y) = \frac{p(y | f) p(f | X)}{p(y | X)}.
\]

(2) Now \( p(f_t | X, y, x_t) \) is used to produce a probabilistic prediction as

\[
\tilde{\pi}_t \triangleq p(y_t = +1 | X, y, x_t) = \int \sigma(f_t) p(f_t | X, y, x_t) df_t.
\]

3 DATASET DESCRIPTION

In this study, 10,000 lung CT scan images were captured at Omid Hospital in Tehran. The images have been collected from February 2020 to April 2020. The inclusion criterion includes symptomatic patients who have been referred to Omid Hospital. The exclusion criterion is pregnant women. 48.2% of the collected images belong to COVID-19 patients and 51.8% represent healthy patients. The mean and standard deviation of the patients’ age were 49.5 \pm 18.5 years old with 45% of the cases being male. Each image has been screened by three radiologists for COVID-19. Hospital ethical approval was obtained to conduct this study and use these images. Typical normal and COVID-19 CT images used for this study are shown in Figure 1.

4 PROPOSED METHOD

Despite the fact that the new coronavirus is spreading quickly worldwide, the amount of available labeled data for diagnosing the disease is limited. Also, due to the rapid growth of this virus, the ability to label data quickly is desirable. Therefore, this study takes a semi-supervised approach to
train the deep neural network for diagnosing coronavirus with high accuracy where only 10% of the training dataset is labeled. The process of training data preparation is explained in Section 4.1. The prepared data are then used by the proposed method of two phases which are explained clearly in Sections 4.2 and 4.3.

4.1 Dataset Management

Our dataset contains 10,000 CT scan images which are divided into 80% training (8,000 images) and 20% testing (2,000 images). The test data are all labeled. Moreover, 10% (800 images) of the training data is labeled. For validation, 20% (160 images) of the labeled training data is used. To further improve the feature extraction in GAN, Sobel edge detection is applied to the data. Next, labeled and unlabeled data are pre-processed. To this end, images are resized to $100 \times 100$ and normalized. The normalization maps the intensity of all image pixels to interval $[0, 1]$.

4.2 First Phase: Unsupervised Training

The objective of the first phase is to learn the underlying distribution of dataset samples achieved by standard unsupervised training [20] of the GAN model using unlabeled data. The reason for choosing GAN is its ability to learn the training data distribution. In the first phase, the generator is trained such that it deceives the discriminator with high-quality fake images. The discriminator is trained in an unsupervised way by using the unlabeled data to distinguish between real and fake images. Upon completion of the first phase, the discriminator has learned to identify the valid CT scan images. The output of the first phase is the trained parameters of discriminator which are used as a starting point for the second phase.

4.3 Second Phase: Supervised Training

The objective of the second phase is to fine-tune the discriminator parameters such that it can classify CT scan images as COVID or healthy. To this end, the labeled data are used in a supervised manner [7] to train the discriminator. Although the number of labeled data is small, the discriminator is able to distinguish COVID and healthy samples since its parameters have already been initialized to sensible values in the first phase (Section 4.2). It is worth noting that the generator is not needed in the second phase. The two phases of our method are depicted in Figure 2. As can be
seen, the discriminator output in the first phase is the probability of being real given the input image while the output in the second phase is the probability of being COVID infected. Each sample is classified as COVID if \( p(COVID) > 0.5 \), otherwise it is classified as Healthy. The advantage of probabilistic output is that it can also be used as an uncertainty measure. The closer the discriminator output is to 0.5, the higher the uncertainty about the sample class (Health/COVID). Hence, looking at the discriminator output, the human expert gets an insight into how much the classifier output can be trusted. In the case in which the classification has low confidence (\( p(COVID) \sim 0.5 \)), the human expert can ask for another human/classifier to confirm the class of the test images.

4.4 Putting It All Together

The main steps of our method are summarized in Figure 3. Steps 1–4 are related to dataset management as explained in Section 4.1. The fifth step involves the creation of GAN generator and discriminator networks’ structures that are presented in Figures 4 and 5, respectively. The sixth step involves the unsupervised training of GAN as explained in Section 4.2. In the seventh step, the discriminator (with parameters initialized in step 6) is trained in a supervised manner to gain the ability to classify input images as healthy or COVID patients (Section 4.3). The discriminator last layer activation provides the probability of being COVID for each input image. Finally, the trained discriminator is used to classify the test data in step eight.

4.4.1 SCLLD Pseudo-code. The pseudo-code in Algorithm 1 demonstrates how the steps of the proposed method described in Sections 4.1, 4.2, and 4.3 interact with each other. The input to the algorithm is CT image dataset \( C \), number of training iterations \( N \), and batch size \( m \). In line 1

![Fig. 2. Schematic illustration of the proposed SCLLD model.](image)

![Fig. 3. The proposed method steps.](image)
of the pseudo-code, the dataset is partitioned to training and test sets. In lines 2 and 3, some of the data are labeled and the validation set is formed. Next, the Sobel operator is applied to the samples (line 4). The samples are then pre-processed in lines 5–7. After creation of the GAN model (line 8), standard unsupervised training of GAN is performed in lines 9–14. Finally, the trained discriminator is further trained but with labeled data \( \{(x^{(i)}, y^{(i)}) \mid i = 1, \ldots, l_{\text{supervised}}\} \) (lines 15–18). Recall that 80% of total dataset \( C \) is used for training and 10% of the training set is labeled. Therefore, \( l_{\text{supervised}} \) is equal to \( \lceil 0.1 \times 0.8 \times |C| \rceil \). Each labeled training sample \( (x^{(i)}, y^{(i)}) \) consists of a CT image \( x^{(i)} \) and label (COVID/healthy) \( y^{(i)} \). Similar to the unsupervised phase, the loss function used for supervised training of the discriminator is Binary Cross Entropy (BCE). The only difference is that in unsupervised training, the labels of true and fake samples are
ALGORITHM 1: SCLLD pseudo-code

Input: CT image dataset $C$, Number of iterations $N$, Batch size $m$

// Dataset management (Section 4.1)
1 Partition $C$ to 80% training set, 20% test set
2 Label test set and 10% of training set manually
3 Use 20% of labeled training set as validation set
4 Perform Sobel operator on samples in $C$

for $i = 1$ to $|C|$ do
5    Resize($x^{(i)}$, new_size=[100 × 100])
6    Normalize($x^{(i)}$, range=[0, 1])
7
8 Create GAN generator $G$ and discriminator $D$

// Gan unsupervised training (Section 4.2)
9 for $n = 1$ to $N$ do
10       // Train discriminator
11       $loss_D = \frac{1}{m} \sum_{i=1}^{m} \{ \log D(x^{(i)}) + \log [1 - D(G(z^{(i)}))] \}$ // Equation (2)
12       $\theta_D = \text{Adam}(loss_D, \nabla_{\theta_D} loss_D)$
13       // Train generator
14       $loss_G = \frac{1}{m} \sum_{i=1}^{m} \{ \log [1 - D(G(z^{(i)}))] \}$ // Equation (3)
15       $\theta_G = \text{Adam}(loss_G, \nabla_{\theta_G} loss_G)$
16       Evaluate $\{G, D\}$ on validation set
17
18 // Discriminator supervised training (Section 4.3)
19 for $n = 1$ to $N$ do
20       $loss_D = -\frac{1}{m} \sum_{i=1}^{m} \{ y^{(i)} \log D(x^{(i)}) + (1 - y^{(i)}) \log [1 - D(x^{(i)})] \}$ // BCE loss
21       $\theta_D = \text{Adam}(loss_D, \nabla_{\theta_D} loss_D)$
22       Evaluate $\{G, D\}$ on validation set
23
24 return $D$

assumed to be one and zero, whereas for supervised training the labels $\{y^{(i)}, i = 1, \ldots, I_{\text{supervised}}\}$ are obtained from training data.

4.5 GAN Architecture

The generator architecture [21] is shown in Figure 4. The input to the generator is a (Gaussian) noise vector which is fed to multiple convolutional layers. These layers convert the noise vector to synthetic CT scan images. The discriminator architecture is depicted in Figure 5. The input to discriminator is a two-dimensional image which could be a real sample or a fake one produced by the generator. During the forward pass of the discriminator, the input image is reduced to a scalar value which denotes the discriminator judgment about the received input sample. In the first phase, the ideal behavior of the discriminator is to provide a higher probability of being real ($p(\text{real})$) for real images and a lower probability for fake ones. Since the discriminator is competing against the generator (usually after the successful training), the probability provided by the discriminator is around 0.5 for both real and fake samples. This stems from the fact that the quality of the generator’s fake samples is high enough to deceive the discriminator. The desired behavior of the discriminator in the second phase is to distinguish between COVID and healthy images.
Table 1. Parameters of the Proposed Method

| Parameters                              | Values          |
|-----------------------------------------|-----------------|
| Size of the convolution kernels         | $3 \times 3$    |
| Generator input noise vector length     | 100             |
| Batch size                              | 32              |
| Number of iterations                    | 3,500, 4,000, and 4,500 |
| Optimizer method                        | Adam            |
| Adam parameters                         | $\beta_1 = 0.9, \beta_2 = 0.999$ |
| Generator and discriminator loss        | Binary Cross Entropy (BCE) |
| Generator learning rate                 | 1e-3            |
| Discriminator learning rate             | 1e-3            |
| Convolution layers activation function  | Leaky ReLU ($\alpha = 0.01$) |
| Edge detection algorithm                | Sobel with kernel size=3 |
| Dropout probability                     | 0.5             |

4.6 Implementation Details

The proposed method has been implemented in Python using the Keras library, which runs on top of TensorFlow. The experiments have been run on a PC with GFORCE GTX 950 GPU and 16 GB of RAM. Hyper-parameters of the method are explained in Table 1.

5 RESULTS

In this section, the experimental results of the GAN and SCLLD methods for detection of COVID-19 patients based on CT scan data are presented. For the SCLLD method, we provided three loss plots at iterations 3,500, 4,000, and 4,500 averaged on five number of runs with 95% confidence intervals. These plots are shown in parts (ac) of Figures 6–8. Part (d) of these figures show ROC plots for the SCLLD method.

Moreover, we have compared the methods based on criteria such as accuracy, precision, sensitivity, recall, specificity, F1-score, and AUC. These criteria are computed according to the following equations [52]:

\[
\text{Accuracy} = \frac{TP + TN}{FP + FN + TP + TN},
\]

\[
\text{Sensitivity} = \frac{TP}{TP + FN}.
\]

\[
\text{Precision} = \frac{TP}{TP + FP}.
\]

\[
\text{F1-Score} = \frac{2TP}{2TP + FP + FN}.
\]

\[
\text{Specificity} = \frac{TN}{TN + FP}.
\]

The experimental results for the SCLLD method are presented in Section 5.1. We have also summarized the results in Table 2.

5.1 SCLLD Results on Our Dataset

The results of experiments at iterations 3,500, 4,000, and 4,500 of the SCLLD method are presented in Figures 6–8. The proposed method has two training phases: unsupervised and supervised. The loss plots in parts (a) and (b) of Figure 6 belong to the unsupervised training phase while the loss
Table 2. Comparison Results between Methods of GAN and SCLLD Through Different Evaluation Metrics

| Methods | Number of Iterations | Accuracy (%) | Precision (%) | Recall (%) | Specificity (%) | F1-score (%) | Loss | AUC (%) |
|---------|----------------------|--------------|---------------|------------|----------------|--------------|------|---------|
| GAN     | 3,500                | 98.42        | 99.39         | 97.39      | 99.42          | 98.38        | 0.2567 | 98.4061 |
|         | 4,000                | 98.37        | 100           | 96.69      | 100            | 98.32        | 0.2856 | 98.3466 |
|         | 4,500                | 97.09        | 100           | 94.09      | 100            | 96.95        | 0.2205 | 97.0440 |
| SCLLD   | 3,500                | 99.61        | 99.8          | 99.4       | 99.81          | 99.6         | 0.2548 | 99.6023 |
|         | 4,000                | 98.77        | 100           | 97.49      | 100            | 98.73        | 0.1570 | 98.7474 |
|         | 4,500                | 98.96        | 97.94         | 100        | 97.96          | 98.96        | 0.3853 | 98.9805 |

Fig. 6. SCLLD method results at iteration 3,500: (a) loss plot on real data during unsupervised training, (b) loss plot on fake data (Generator output) during unsupervised training, (c) loss plot for labeled data during supervised training, and (d) ROC plot for classification of labeled data.

The motivation behind plotting the results at multiple points during training is the investigation to the increasing number of iterations. This investigation is important since at some point during the training, the performance of SCLLD may degrade. Therefore, it is common practice to diagnose the
Uncertainty-Aware Semi-Supervised Method Using Large Unlabeled

Fig. 7. SCLLD method results at iteration 4,000: (a) loss plot on real data during unsupervised training, (b) loss plot on fake data (Generator output) during unsupervised training, (c) loss plot for labeled data during supervised training, and (d) ROC plot for classification of labeled data.

SCLLD performance and stop the training when the performance of the model does not improve any more.

It can be noted that, parts (a) and (b) of Figures 6–8 reveal that using the Sobel operator has accelerated the training pace. Therefore, the best performance is achieved at iteration 3,500. Beyond that iteration, the model has started to degrade quickly, leading to considerable loss value at iteration 4,500 (part (b) of Figure 8).

The ROC plots at different iterations of the SCLLD method match the accuracy results reported in Table 2. Considering that the best performance is achieved at iteration 3,500, it makes sense that ROC plots at this iteration reach a value of 1.0 faster compared to their counterparts at iterations 4,000 and 4,500. For SCLLD, the accuracy decreases at iteration 4,000 but increases slightly at iteration 4,500. That is why ROC at iteration 4,500 increases faster as compared to ROC at iteration 4,000. An example of the CT images, results of applying the Sobel filter on them, and the final results of GRAD-CAM are shown in Figure 9. Grad-CAM is a generalization of the Class Activation Mapping. It does not need re-training and can be applied broadly to any CNN-based architectures [45].

5.1.1 Sensitivity Analysis Regarding Labeled Training Data Size. Based on the existing literature, GAN is good at capturing the underlying distribution of the training dataset [42]. However, the
limited number of available labeled data might hurt the classification performance. This is the motivation for our semi-supervised approach SCLLD. In this section, we evaluated the performance of our method for different numbers of labeled data. This experiment reveals how much our semi-supervised approach can tolerate limited labeled data. The results for increasing sample size of training labeled data (1% to 10%) are presented in Table 3. It is clear that even in a semi-supervised setting, the amount of labeled data cannot be less than a certain threshold, otherwise the performance drops dramatically. This scenario is observed if the labeled data falls less than 3%. For labeled data size above 6%, the performance metrics are similar and the best results are achieved when 8% of the data are labeled. Theoretically, increasing the training data should lead to better performance. However, it is customary to train deep neural networks using mini-batch sizes smaller than the whole dataset. The benefit of mini-batch training is twofold [34]. First, smaller batch size reduces the memory demand for one step of training. Second, smaller batch size results in noisy gradients which have regularization effect preventing over-fitting. The noisy gradients lead to stochasticity during the learning process, which is why the accuracy values in Table 3 do not necessarily increase with larger labeled data size.

As can be seen in Table 3, the total training time is not a function of labeled data size. This is due to the fact that regardless of being labeled or not, all samples are eventually used during the

---

**Fig. 8. SCLLD method results at iteration 4,500:** (a) Loss plot on real data during unsupervised training, (b) Loss plot on fake data (Generator output) during unsupervised training, (c) Loss plot for labeled data during supervised training, and (d) ROC plot for classification of labeled data.
Fig. 9. Examples of the CT images, results of applying the Sobel filter on them, and the final results of GRAD-CAM are shown in the first, second, and third rows, respectively. The first and second columns are the images of normal cases while the third and fourth columns are the images of sick cases.

Table 3. Effect of Labeled Training Data Size on SCLLD Method Performance within 4,000 Iterations:
In Total Training Time Column, $m$ and $s$ Represent Minutes and Seconds, Respectively

| Amount of labeled training data (%) | Accuracy (%) | Precision (%) | Recall (%) | Specificity (%) | F1-score (%) | Loss | AUC (%) | Total training time (m:s) |
|-----------------------------------|-------------|---------------|------------|----------------|--------------|------|---------|--------------------------|
| 1                                 | 52.37       | 100           | 3.21       | 100            | 6.22         | 2.0755 | 51.6     | 26:52                    |
| 2                                 | 64.6        | 83.18         | 35.17      | 93.11          | 49.44        | 1.0012 | 64.14    | 26:52                    |
| 3                                 | 93.54       | 89.52         | 98.4       | 88.83          | 93.75        | 1.2660 | 93.62    | 26:57                    |
| 4                                 | 96.15       | 98.52         | 93.59      | 98.64          | 95.99        | 0.2459 | 96.11    | 27:06                    |
| 5                                 | 95.36       | 91.7          | 99.6       | 91.26          | 95.49        | 1.0138 | 95.43    | 26:55                    |
| 6                                 | 99.26       | 98.91         | 99.6       | 98.93          | 99.25        | 0.2071 | 99.27    | 26:56                    |
| 7                                 | 99.56       | 99.8          | 99.3       | 99.81          | 99.55        | 0.0963 | 99.55    | 26:47                    |
| 8                                 | 99.95       | 100           | 99.9       | 100            | 99.95        | 0.1823 | 99.95    | 26:57                    |
| 9                                 | 99.41       | 99.7          | 99.1       | 99.71          | 99.4         | 0.1091 | 99.4     | 26:56                    |
| 10                                | 99.36       | 98.81         | 99.9       | 98.83          | 99.35        | 0.2391 | 99.37    | 26:54                    |

supervised or semi-supervised training. Hence, the total cost of training is determined by the total number of labeled/unlabeled samples.

5.2 Evaluation of Convolution Neural Network on Our Dataset
To highlight the role of using unlabeled data in our method, we have also trained a CNN on labeled data in a purely supervised manner. The structure of CNN is the same as our GAN discriminator.
According to Figure 10, it is possible to compare the loss of CNN and our proposed method during the training and validation phases. According to this figure, it is clear that although the loss function of our proposed method is worse than CNN during the training phase, it shows better performance during the validation phase.

The accuracy on training data and validation data during the training of CNN is shown in Figure 11. Looking at the accuracy plot, it is evident that the training has been stopped early by the Keras library. The reason for early stopping is the fact that the CNN performance has started to degrade after iteration 11.

The performance of the trained CNN on test data is presented in Table 4. CNN is one of the most powerful classifiers on image data but it usually needs lots of training samples. Therefore, in the absence of a sufficient amount of labeled data, CNNs will not be able to reach their expected performance. The setup in this section reflects such a scenario where labeled samples are limited but we have plenty of unlabeled ones. Thus, it should come as no surprise that CNN will suffer a limited labeled training set and exhibit low performance. On the contrary, our method is able to learn useful features from unlabeled samples in an unsupervised manner. That is why our method can perform well even when labeled data is limited.

### 5.3 Evaluation of Gaussian Process on Our Dataset

To gain a better understanding of our results, we have also compared our proposed method with a GP approach which is trained in a supervised manner with our dataset. The advantage of GP is
that it can represent any dataset. The drawback is its high computational complexity, which is of the order $O(N^3)$ \cite{17} for $N$ training samples. Considering that the amount of labeled data in our dataset is small, GP is an ideal choice since (i) the computation overhead is limited as the amount of available labeled data is limited. (ii) GP provides an estimate of the uncertainty in portions of the sample space where it has not seen enough data so the user knows where the GP output can be trusted.

In our experiment, we used a squared exponential kernel function (see Section 2.2) for GP. The initial value of the kernel function hyper-parameters ($\sigma_f^2, \sigma_n^2, l$) are given in Table 5. Moreover, the mean function of the GP is initially set to constant zero.

The performance of GP using our dataset is presented in Table 6. For each of the performance metrics (accuracy, precision, etc.), a 95% confidence interval is reported over four runs. As can be seen in Table 6, despite tight confidence intervals on performance metrics, GP has poor average performance. This is due to a lack of sufficient labeled data. We can observe that even robust supervised methods cannot deal with limited labeled data. This clearly shows the importance of semi-supervised learning methods. The ROC plot of GP experiment on our dataset is also presented in part (a) of Figure 12. To ease the comparison between GP and our method, part (d) of Figure 8 (SCLLD ROC plot) is repeated as part (b) of Figure 12.

### 5.4 Evaluation of Our Method Using a Public Dataset

To evaluate the generalization of our method, we have evaluated the performance of our model using the publicly available Kaggle dataset \cite{2}. The dataset belongs to China National Centre for Bio-information with 8,535 positive and 9,430 negative CT scan samples. The average and standard deviation of samples age are 48.005 and 17.272, respectively. The percentages of male and female

| Accuracy (%) | Precision (%) | Recall (%) | Specificity (%) | F1-score (%) | AUC (%) | Total training time (m:s) |
|--------------|---------------|------------|-----------------|--------------|---------|--------------------------|
| 68.34 ± 4.11 | 62.6 ± 5.06   | 91.2 ± 6.15| 46.40 ± 5.21    | 74.2 ± 1.69  | 68.70 ± 3.98 | 00:09                    |

| Parameters | Values                  |
|------------|-------------------------|
| Initial mean function | $m(t) = 0$               |
| Kernel type | Squared exponential     |
| Signal variance ($\sigma_f^2$) | 1.0                     |
| Noise variance ($\sigma_n^2$) | 0.0                     |
| Length scale ($l$) | 1.0                     |

| Accuracy (%) | Precision (%) | Recall (%) | Specificity (%) | F1-score (%) | AUC (%) | Total training time (m:s) |
|--------------|---------------|------------|-----------------|--------------|---------|--------------------------|
| 54.12 ± 0.42 | 52 ± 0.27     | 100 ± 0.43 | 10.38 ± 0.81    | 68 ± 0.39    | 54.83 ± 0.41 | 16:52                    |
samples are 55.1% and 44.9%, respectively. The comparison results of GAN and SCLLD using the public dataset are presented in Table 7. All the performance metrics (accuracy, precision, etc.) reported in the table have obtained more than 95% confidence interval over five runs with different random seeds. As it is evident from Table 7, confidence intervals of performance metrics of GAN are wider than their counterparts for the SCLLD method. This stems from the fact that GAN does not have the opportunity of learning from a variety of unlabeled samples. GAN can only learn the underlying distribution of limited labeled samples, which is not a good representative of the various samples the system may encounter during the testing phase. Such scenario leads to an overall lower performance of GAN compared to our semi-supervised methods which can exploit a mix of labeled and unlabeled samples.

### 6 DISCUSSION AND CONCLUSION

The COVID-19 pandemic has led to more than one million deaths and is spreading faster than expected [3]. Currently, early detection of virus is of the highest importance, which is a challenging task [61]. Moreover, the virus detection test is expensive and not robust [4]. Hence, chest X-ray and CT scans may be a useful tool to detect COVID-19 at an early stage [16]. In addition to these detection techniques, researchers are utilizing artificial intelligence methods to accelerate the detection and treatment of COVID-19 infected patients [56, 60]. This is the motivation behind our proposed method based on deep learning using a semi-supervised method.

Our objective is to classify healthy and COVID-19 infected patients accurately using chest CT scan images. We have used the GAN method to deal with limited labeled data. To this end, GAN
Table 8. Summary of Comparison with Other Automated DL Methods Developed to Detect COVID-19 with CT Images

| Authors                  | DL method              | Training approach | Number of images | Performance                      |
|--------------------------|------------------------|-------------------|-----------------|-----------------------------------|
| Ardakani et al. [12]     | ResNet-101             | Supervised        | 1,020 CT        | AUC: 0.994                        |
|                          |                        |                   |                 | Sensitivity: 100%                 |
|                          |                        |                   |                 | Specificity: 99.02%               |
|                          |                        |                   |                 | Accuracy: 99.51%                  |
| Li et al. [32]           | ResNet50               | Supervised        | 4,356 CT        | Specificity: 92%                  |
|                          |                        |                   |                 | Sensitivity: 87%                  |
|                          |                        |                   |                 | AUC: 0.95                         |
| Wang et al. [57]         | CNN                    | Supervised        | 453 CT          | Sensitivity: 74%                  |
|                          |                        |                   |                 | Specificity: 67%                  |
|                          |                        |                   |                 | Accuracy: 73.1%                   |
| Gozes et al. [23]        | ResNet-50              | Supervised        | 6,150 CT        | Specificity: 92.2%                |
|                          |                        |                   |                 | Sensitivity: 98.2%                |
|                          |                        |                   |                 | AUC: 0.996                        |
| Hemdan et al. [24]       | Google MobileNet       | Supervised        | 50 X-ray        | F1-score: 91%                     |
| and modified VGG19       |                        |                   |                 |                                   |
| Apostolopoulos and       | CNN with               | Supervised        | 1,427 X-ray     | Specificity: 96.46%               |
| Mpesiana [10]            | Transfer Learning      |                   |                 | Sensitivity: 98.66%               |
|                          |                        |                   |                 | Accuracy: 96.78%                  |
| Xu et al. [59]           | CNN                    | Supervised        | 618 CT          | Specificity: 92.2%                |
|                          |                        |                   |                 | Sensitivity: 98.2%                |
|                          |                        |                   |                 | AUC: 0.996                        |
| Song et al. [53]         | ResNet50               | Supervised        | 1,485 CT        | Sensitivity: 93%                  |
|                          |                        |                   |                 | AUC: 0.99                         |
| Sethy et al. [46]        | SVM plus Resnet50      | Supervised        | 50 X-ray        | Kappa: 90.76%                     |
|                          |                        |                   |                 | F1-score: 91.41%                  |
|                          |                        |                   |                 | FPR: 95.52%                       |
|                          |                        |                   |                 | Accuracy: 95.38%                  |
| Narin et al. [37]        | Inception-ResNetV2,    | Supervised        | 3,141 X-ray     | Accuracy: 98%                     |
|                          | InceptionV3, and       |                   |                 | AUC: 0.89                         |
|                          | ResNet50               |                   |                 | Accuracy: 72.77%                  |
|                          |                        |                   |                 | Specificity: 73.83%               |
|                          |                        |                   |                 | Sensitivity: 71.70%               |
|                          |                        |                   |                 | AUC: 83.61%                       |
| Shi et al. [50]          | VB-Net                 | Supervised        | 196 CT          | Accuracy: 96.25%                  |
|                          |                        |                   |                 | Specificity: 96.21%               |
|                          |                        |                   |                 | Sensitivity: 96.29%               |
|                          |                        |                   |                 | F1-score: 96.29%                  |
| Zhang et al. [62]        | CNN                    | Supervised        | 5,977 X-ray     | Accuracy: 94.1%                   |
|                          |                        |                   |                 | Specificity: 100%                 |
|                          |                        |                   |                 | Sensitivity: 90%                  |
| Jaiswal et al. [26]      | DenseNet201            | Supervised        | 2,492 CT        | Accuracy: 83%                     |
|                          |                        |                   |                 | Specificity: 73.83%               |
|                          |                        |                   |                 | Sensitivity: 71.70%               |
|                          |                        |                   |                 | AUC: 83.61%                       |
| Maghdid et al. [36]      | AlexNet                | Supervised        | 361 CT          | Accuracy: 96.25%                  |
|                          |                        |                   |                 | Specificity: 96.21%               |
|                          |                        |                   |                 | Sensitivity: 96.29%               |
|                          |                        |                   |                 | F1-score: 96.29%                  |

Continued
Table 8. Continued

| Authors         | DL method      | Training approach | Number of images | Performance          |
|-----------------|----------------|-------------------|------------------|----------------------|
| Zhao et al. [64] | ResNet-50      | Supervised        | 349 CT           | Accuracy: 89%
|                 |                |                   |                  | F1-score: 90%
|                 |                |                   |                  | AUC: 98%       |
| Ko et al. [30]  | ResNet-50      | Supervised        | 3,993 CT         | Accuracy: 96.97%    |
| Kumar et al. [31]| Capsule Network| Supervised        | 3,406 CT         | Accuracy: 98.68%
|                 |                |                   |                  | Sensitivity: 98%|
| Ni et al. [38]  | 3D U-Net       | Supervised        | 1,443 CT         | Accuracy: 82%
|                 |                |                   |                  | Specificity: 63%
|                 |                |                   |                  | Sensitivity: 96%
|                 |                |                   |                  | F1-score: 86%  |
| Alom et al. [8] | RNN with       | Supervised        | 420 CT           | Accuracy: 98.78%    |
|                 | Transfer Learning|                  |                  |                     |
| Javaheri et al. [27] | 3D CNN      | Supervised        | 89,145 CT        | Accuracy: 90%
|                 |                |                   |                  | Specificity: 94%
|                 |                |                   |                  | Sensitivity: 87.5% |
| Saeedi et al. [43] | DenseNet121  | Supervised        | 349 CT           | Accuracy: 90.61%
|                 |                |                   |                  | Sensitivity: 90.80%
|                 |                |                   |                  | AUC: 95.05%    |
| Zhang et al. [63] | CNN            | Supervised        | 640 CT           | Accuracy: 98.06%
|                 |                |                   |                  | Specificity: 99.38%
|                 |                |                   |                  | Sensitivity: 97.24%
|                 |                |                   |                  | AUC: 98.31%
|                 |                |                   |                  | F1-score: 98.39% |
| Wang et al. [58] | GCN and CNN    | Supervised        | 640 CT           | Accuracy: 96.64%
|                 |                |                   |                  | Specificity: 97.03%
|                 |                |                   |                  | Sensitivity: 96.25% |
| Proposed method | SCLLD          | Semi-supervised   | 10,000 CT        | Accuracy: 99.56 ± 0.20%
|                 |                |                   |                  | Specificity: 99.40 ± 0.18%
|                 |                |                   |                  | Sensitivity: 99.88 ± 0.24%
|                 |                |                   |                  | AUC: 99.56 ± 0.20% |

Discriminator network parameters are initialized using unlabeled data. Next, the network parameters are fine-tuned with a small number of labeled data. The proposed approach is further improved by pre-processing the input images using Sobel edge detection. After the training, the GAN discriminator is used to classify the test data. Using the probabilistic discriminator output, our system provides the human experts with an uncertainty measure about its decision. The experimental results reveal the superiority of the SCLLD method as compared to the GAN method alone (Table 3).

To evaluate our SCLLD against other supervised methods, we have compared SCLLD performance with our dataset and other methods using their datasets (Table 8). Testing on different datasets is necessary since our dataset has limited labeled data which can hurt the performance of supervised methods [35]. This comparison reveals that while supervised learners are crippled in the absence of enough labeled data, our method is capable of dealing with limited labeled data and learning from unlabeled data reaching performance which is on par with supervised methods. This is clearly the advantage of our method, which is due to its semi-supervised nature.
The drawback of our method is its computation overhead [25]. In this work, we have used two training phases. In the first phase, standard GAN training is executed, which involves training of a generator and discriminator together. The first source of overhead is generator training since once training is done, it is no longer needed. In the second phase of training, the discriminator is trained using labeled data. Hence, the second source of overhead is training of the discriminator twice (unsupervised and supervised). We may note that the final output of our method is the trained discriminator.

In future work, we investigate the effect of training the generator using feature matching [44]. GAN models are usually hard to train and may suffer instability during training [33, 41]. Therefore, it is interesting to explore the effect of using soft labels [44] during the unsupervised training phase of GAN. To this end, hard labels (zero and one) corresponding to fake and real samples are replaced with a random number between [0, 0.3] and [0.7, 1], respectively. Soft labels usually lead to stable training.

REFERENCES

[1] 2021. Retrieved May 1, 2021 from https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200513-covid-19-sitrep-114.pdf?sfvrsn=17ebbbe_4.
[2] 2021. Retrieved May 1, 2021 from https://www.kaggle.com/bayazjafarli/covid19-covidpneumanianormal-cases.
[3] 2021. Retrieved May 1, 2021 from https://coronavirus.jhu.edu/map.html.
[4] 2021. Retrieved May 1, 2021 from https://utsmed.com/medblog/covid19-testing-methods/.
[5] Tao Ai, Zhenlu Yang, Hongyan Hou, Chenao Zhan, Wenzhi Lv, Qian Tao, Ziyong Sun, and Liming Xia. 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 (2020), E32–E40.
[6] Roohallah Alizadehsani, Zahra Alizadeh Sani, Mohaddeseh Behjati, Zahra Roshanzamir, Sadiq Hussain, Niloofar Abedini, Fereshteh Hasanzadeh, Abbas Khosravi, Afshin Shoebi, Mohamad Roshanzamir, et al. 2020. Risk factors prediction, clinical outcomes, and mortality in COVID-19 patients. Journal of Medical Virology 93 (2020), 2307–2320.
[7] Roohallah Alizadehsani, Abbas Khosravi, Mohamad Roshanzamir, Moloud Abdar, Nizal Sarrazfadegan, Davood Shafie, Fahime Khoeimeh, Afshin Shoebi, Saeid Nahavandi, Maryam Panahiazar, et al. 2020. Coronavirus artery disease detection using artificial intelligence techniques: A survey of trends, geographical differences and diagnostic features 1991–2020. Computers in Biology and Medicine 128 (2020), 104095.
[8] Md Zabangir Alom, M. M. Rahman, Mst Shamima Nasrin, Tarek M. Taha, and Vijayan K. Asari. 2020. COVID_MTNet: Covid-19 detection with multi-task deep learning approaches. arXiv:2004.03747.
[9] Marios Anthimopoulos, Stergios Christodoulidis, Lukas Ebner, Andreas Christe, and Stavroula Mougiakakou. 2016. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. IEEE Transactions on Medical Imaging 35, 5 (2016), 1207–1216.
[10] Ioannis D. Apostolopoulos and Tzani A. Mpesiana. 2020. Covid-19: Automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine 43, 2 (2020), 635–640.
[11] Ali Abbasian Ardakani, U. Rajendra Acharya, Sina Habibollahi, and Afshin Mohammadi. 2021. COVIDIag: A clinical CAD system to diagnose COVID-19 pneumonia based on CT findings. European Radiology 31, 1 (2021), 121–130.
[12] Ali Abbasian Ardakani, Alireza Rajabzadeh Kanafi, U. Rajendra Acharya, Nazanin Khadem, and Afshin Mohammadi. 2020. Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. Computers in Biology and Medicine 121 (2020), 103795.
[13] Hamzeh Asgharnejad, Afshar Shamsi, Roohallah Alizadehsani, Abbas Krosvari, Saeid Nahavandi, Zahra Alizadeh Sani, and Dipti Srinivasan. 2020. Objective evaluation of deep uncertainty predictions for COVID-19 detection. arXiv:2012.11840.
[14] Mikhail Belyaev, Evgeny Burnaev, and Yermek Kapushev. 2014. Exact inference for Gaussian process regression in case of big data with the Cartesian product structure. arXiv:1403.6573.
[15] Nanshan Chen, Min Zhou, Xuan Dong, Jieming Qu, Fengyun Gong, Yang Han, Yang Qiu, Jingli Wang, Ying Liu, Yuan Wei, et al. 2020. Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: A descriptive study. The Lancet 395, 10223 (2020), 507–513.
[16] Debabrata Dansana, Raghvendra Kumar, Aishik Bhattacharjee, D. Jude Hemanth, Deepak Gupta, Ashish Khanna, and Oscar Castillo. 2020. Early diagnosis of COVID-19-affected patients based on X-ray and computed tomography images using deep learning algorithm. Soft Computing (2020), 1–9.
[17] Adrien Depeursinge, Anne S. Chin, Ann N. Leung, Donato Terrone, Michael Bristow, Glenn Rosen, and Daniel L. Rubin. 2015. Automated classification of usual interstitial pneumonia using regional volumetric texture analysis in high-resolution CT. Investigative Radiology 50, 4 (2015), 261.

[18] Rahimeh Eskandarian, Zahra Alizadeh Sani, Mohaddeseh Behjati, Mehrdad Zahmatkesh, Azadeh Haddadi, Kourosh Kakhki, Mohammad Roshanzamir, Afshin Shoebi, Rooollah Alizadehsani, Sadiq Hussain, et al. 2021. Identification of clinical features associated with mortality in COVID-19 patients. medRxiv. https://doi.org/10.1101/2021.04.19.21255715

[19] Yicheng Fang, Huangqi Zhang, Jicheng Xie, Minjie Lin, Lingjun Ying, Peipei Pang, and Wenbin Ji. 2020. Sensitivity of chest CT for COVID-19: Comparison to RT-PCR. Radiology 296, 2 (2020), E115–E117.

[20] Zoubin Ghahramani. 2003. Unsupervised learning. In Summer School on Machine Learning. Springer, 72–112.

[21] Juan M. Górriz, Javier Ramírez, Andrés Ortiz, Francisco J. Martínez-Murcia, Fermín Segovia, John Suckling, Matthew Leming, Yu-Dong Zhang, Jose Ramón Álvarez-Sánchez, Guido Bologna, et al. 2020. Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications. Neurocomputing 410 (2020), 237–270.

[22] Ezz El-Din Hemdan, Marwa A. Shouman, and Mohamed Esmail Karar. 2020. COVIDX-Net: A framework of deep learning classifiers to diagnose COVID-19 in x-ray images. arXiv:2003.11055.

[23] Taocheng Hu and Jinhui Yu. 2015. Generalized entropy based semi-supervised learning. In 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS’15). IEEE, 259–263.

[24] Aayush Jaiswal, Neha Gianchandani, Dilbag Singh, Vijay Kumar, and Manjit Kaur. 2020. Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. Journal of Biomolecular Structure and Dynamics 39 (2020), 5682–5689.

[25] Tahereh Javaheri, Morteza Homaeyounfar, Zohreh Amoozgar, Reza Reiazi, Fatemeh Homayounieng, Engy Abbas, Azadeh Laali, Amir Reza Radmard, Mohammad Hadi Gharib, Seyed Ali Javad Mousavi, et al. 2021. CovidCTNet: An open-source deep learning approach to identify covid-19 using small cohort of CT images. npj Digit. Med. 4, 29 (2021). https://doi.org/10.1038/s41746-021-00399-3

[26] Rajesh Kumar, Abdullah Aman Khan, Simrin Zhang, WenYong Wang, Yousif Abudridis, Waqas Amin, and Jay Kumar. 2020. Blockchain-federated-learning and deep learning models for covid-19 detection using CT imaging. IEEE Sensors Journal 21, 14 (2021), 16301–16314. DOI: 10.1109/JSEN.2021.3076767

[27] Lin Li, Lixin Qin, Zeguo Xu, Youbing Yin, Xin Wang, Bin Kong, Junjie Bai, Yi Lu, Zhenghan Fang, Qi Song, et al. 2020. Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. Radiology 296 (2020), E65–E71.

[28] Longze Li and Aleksandar Vakanski. 2018. Generative adversarial networks for generation and classification of physical rehabilitation movement episodes. International Journal of Machine Learning and Computing 8, 5 (2018), 428.

[29] Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J. Smola. 2014. Efficient mini-batch training for stochastic optimization. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 661–670.

[30] Xiuming Liu, Dave Zachariah, Johan Wågberg, and Thomas B. Schön. 2018. Reliable semi-supervised learning when labels are missing at random. arXiv:1811.10947. https://arxiv.org/abs/1811.10947.

[31] Halgurd S. Maghdid, Aras T. Asaad, Kayhan Zrar Ghafoor, Ali Safaa Sadiq, Seyyed Mirjalili, and Muhammad Khurram Khan. 2021. Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning. Journal of Medical Internet Research 22, 6 (2021).

ACM Trans. Multimedia Comput. Commun. Appl., Vol. 17, No. 3s, Article 104. Publication date: November 2021.
learning algorithms. In *Multimodal Image Exploitation and Learning* 2021, Vol. 11734. International Society for Optics and Photonics, 117340E.

[37] Ali Narin, Ceren Kaya, and Ziynet Pamuk. 2021. Automatic detection of coronavirus disease (COVID-19) using x-ray images and deep convolutional neural networks. *Pattern Anal Applic* 24 (2021), 1207–1220. https://doi.org/10.1007/s10244-021-09984-y

[38] Qianqian Ni, Zhi Yuan Sun, Li Qi, Wen Chen, Yi Yang, Li Wang, Xinyuan Zhang, Liu Yang, Yi Fang, Zijian Xing, et al. 2020. A deep learning approach to characterize 2019 coronavirus disease (COVID-19) pneumonia in chest CT images. *European Radiology* 30, 12 (2020), 6517–6527.

[39] Emilio Soria Olivas, Jos David Mart Guerrero, Marcelino Martinez-Sober, Jose Rafael Magdalena-Benedito, L Serrano, et al. 2009. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI Global.

[40] Harsh Panwar, P. K. Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Prakhar Bhardwaj, and Vaishnavi Singh. 2020. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. *Chaos, Solitons & Fractals* 140 (2020), 110190.

[41] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv*:1511.06434. https://arxiv.org/abs/1511.06434.

[42] Carl Edward Rasmussen and C. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press. Cambridge, MA.

[43] Abdolkarim Saeedi, Mohammad Bagherzadeh, Seyed Vahid Delavar, and Arash Maghsoudi. 2020. A novel and reliable deep learning web-based tool to detect COVID-19 infection form chest CT-scan. *arXiv*:2006.14419. https://arxiv.org/abs/2006.14419.

[44] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. *Advances in Neural Information Processing Systems* 29 (2016), 2234–2242.

[45] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.

[46] Prabira Kumar Sethy, Santi Kumar Behera, Pradyumna Kumar Rath, and Preesat Biswas. 2020. Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. DOI: 10.33889/IJMEMS.2020.5.4.052

[47] Vruddhi Shah, Rinkal Keniya, Akanksha Shridharani, Manav Punjabi, Jainam Shah, and Ninad Mehendale. 2021. Diagnosis of COVID-19 using CT scan images and deep learning techniques. *Emergency Radiology* (2021), 1–9. DOI: 10.1007/s10140-020-01886-y

[48] Danial Sharifrazii, Roohallah Alizadehsani, Javad Hassannataj Joloudari, Shahab Shamshirband, Sadiq Hussain, Zahra Alizadeh Sani, Fereshteh Hasanzehe, Afshin Shoaiabi, Abdollah Dehzangi, and Hamid Alinejad-Rokny. 2020. CNN-KCL: Automatic myocarditis diagnosis using convolutional neural network combined with K-means clustering. DOI: https://doi.org/10.20944/preprints202007.0650.v1

[49] Danial Sharifrazii, Roohallah Alizadehsani, Mohammad Roshanzamir, Javad Hassannataj Joloudari, Afshin Shoeibi, Mahboobeh Jafari, Sadiq Hussain, Zahra Alizadeh Sani, Fereshteh Hasanzehe, Fahime Khozeimeh, et al. 2021. Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19 patients using X-ray images. *Biomedical Signal Processing and Control* 68 (2021), 102622.

[50] Weiya Shi, Xueqing Peng, Tiefu Liu, Zenghui Cheng, Hongzhou Lu, Shuui Yang, Jiulong Zhang, Mei Wang, Xiaodong Zhang, Ziwang Huang, Jianwen Chen, Ruixuan Wang, Danial Sharifrazi, Roohallah Alizadehsani, Javad Hassannataj Joloudari, Shahab Shamshirband, Sadiq Hussain, Afshin Shoeibi, Huiying Zhao, Yunfei Zha, et al. 2021. Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) pneumonia in chest x-ray images. *Emergency Radiology* 28 (2021), 9, 3 (2021).

[51] Afshin Shoeibi, Navid Ghassemi, Marjane Khodatars, Mahboobeh Jafari, Sadiq Hussain, Rooahallah Alizadehsani, Parisa Moridian, Abbas Khosravi, Hossein Hosseini-Nejad, Modjtaba Rouhani, and A. Khoosravi. 2021. Epileptic seizure detection using deep learning techniques: A review. *International Journal of Environmental Research and Public Health* 18, 11 (2021), 5780.

[52] Afshin Shoeibi, Marjane Khodatars, Roohallah Alizadehsani, Navid Ghassemi, Mahboobeh Jafari, Parisa Moridian, Ali Khadem, Delaram Sadeghi, Sadiq Hussain, Assef Zare, et al. 2020. Automated detection and forecasting of COVID-19 using deep learning techniques: A review. *arXiv*:2007.10785. https://arxiv.org/abs/2007.10785.

[53] Ying Song, Shuangjia Zheng, Liang Li, Xiang Zhang, Xiaodong Zhang, Ziwang Huang, Jianwen Chen, Ruixuan Wang, Huiying Zhao, Yunfei Zha, et al. 2021. Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2021).

[54] Siija Tian, Nan Hu, Jing Lou, Kun Chen, Xuqin Kang, Zhenjun Xiang, Hui Chen, Dali Wang, Ning Liu, Dong Liu, et al. 2020. Characteristics of COVID-19 infection in Beijing. *Journal of Infection* 80, 4 (2020), 401–406.

[55] Yu Tingting, Wang Junqian, Wu Lintai, and Xu Yong. 2019. Three-stage network for age estimation. *CAAI Transactions on Intelligence Technology* 4, 2 (2019), 122–126.

[56] Linda Wang, Zhong Qi Lin, and Alexander Wong. 2020. OVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest x-ray images. *Scientific Reports* 10, 1 (2020), 1–12.
[57] Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, et al. 2021. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). European Radiology (2021), 1–9.

[58] Shui-Hua Wang, Vishnu Varthan Govindaraj, Juan Manuel Górriz, Xin Zhang, and Yu-Dong Zhang. 2021. COVID-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. Information Fusion 67 (2021), 208–229.

[59] Xiaowei Xu, Xiangao Jiang, Chunlian Ma, Peng Du, Xukun Li, Shuangzhi Lv, Liang Yu, Qin Ni, Yanfei Chen, Junwei Su, et al. 2020. A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering 6, 10 (2020), 1122–1129.

[60] Rabab Yasin and Walaa Gouda. 2020. Chest X-ray findings monitoring COVID-19 disease course and severity. Egyptian Journal of Radiology and Nuclear Medicine 51, 1 (2020), 1–18.

[61] Haider Hashim Zalzala. 2020. Diagnosis of COVID-19: Facts and challenges. New Microbes and New Infections (2020), 100761.

[62] Jianpeng Zhang, Yutong Xie, Yi Li, Chunhua Shen, and Yong Xia. 2020. COVID-19 screening on chest x-ray images using deep learning based anomaly detection. https://europepmc.org/article/ppr/ppr346108.

[63] Yu-Dong Zhang, Suresh Chandra Satapathy, Li-Yao Zhu, Juan Manuel Górriz, and Shui-Hua Wang. 2020. A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling. IEEE Sensors Journal (2020).

[64] Jinyu Zhao, Yichen Zhang, Xuehai He, and Pengtao Xie. 2020. CVID-CT-Dataset: A CT scan dataset about COVID-19. arXiv:2003.13865. https://arxiv.org/abs/2003.13865.

Received December 2020; revised March 2021; accepted April 2021