

Different Deep Learning Based Classification Models for COVID-19 CT-Scans and Lesion Segmentation Through the cGAN-UNet Hybrid Method



Ivoline C. Ngong¹, Nurdan Akhan Baykan^{*1}

Department of Computer Engineering, Konya Technical University, Konya 42075, Turkey

Corresponding Author Email: nbaykan@ktun.edu.tr

<https://doi.org/10.18280/ts.400101>

ABSTRACT

Received: 25 November 2022

Accepted: 9 February 2023

Keywords:

COVID-19 segmentation, COVID-19 classification, conditional generative adversarial network (cGAN), convolutional neural network (CNN), PatchCNN, capsule neural network (CapsNet)

The new coronavirus, which emerged in early 2020, caused a major global health crisis in 7 continents. An essential step towards fighting this virus is computed tomography (CT) scans. CT scans are an effective radiological method to detecting the diagnosis in early stage, but have greatly increased the workload of radiologists. For this reason, there are systems needed that will reduce the duration of CT examinations and assist radiologists. In this study, a two-stage system has been proposed for COVID-19 detection. First, a hybrid method is proposed that can segment the infected region from CT images. The reason for this is that there is not always a reference image in the datasets used in the classification. For this purpose; UNet, UNet++, SegNet and PsPNet were used both separately and as hybrids with GAN, to automatically segment infected areas from chest CT slices. According to the segmentation results, cGAN-UNet hybrid system was selected as the most successful method. Experimental results show that the proposed method achieves a segmentation success with a dice score of 92.32% and IoU score of 86.41%. In the second stage, three classifiers which include a Convolutional Neural Network (CNN), a PatchCNN and a Capsule Neural Network (CapsNet) were used to classify the generated masks as either COVID-19 or not, using the segmented images obtained from cGAN-UNet. Success of these classifiers was 99.20%, 92.55% and 73.84%, respectively. According to these results, the highest success was achieved in the system where cGAN-UNet and CNN are used together.

1. INTRODUCTION

At the end of 2019, the novel coronavirus disease (COVID-19) caused by SARS-CoV-2 emerged in Wuhan, China [1]. World Health Organization (WHO) declared the outbreak as a Public Health Emergency of International Concern (PHEIC) on January 30, 2020 and subsequently declared it a pandemic on March 11, 2020 [2, 3]. As of December 15 2020, more than 70 million cumulative cases and 1.6 million deaths have been reported globally since the start of the pandemic [4].

Given the highly contagious nature of COVID-19, critical step towards fighting COVID-19 is early detection to limit the spread of the virus. Reverse transcriptase-polymerase chain reaction (RT-PCR) is used worldwide though some reports on the sensitivity of RT-PCR were not clear or consistent. Furthermore, the process is required to be performed in a highly controlled environment, there is a shortage of materials and also, it can take more than 4 hours to obtain the test results [5-8].

A complementary method for COVID-19 screening is the use of radiography examination on medical images like chest computed tomography (CT), Magnetic resonance imaging (MRI) scans or chest X-ray (CXR) scans. The authors of [9, 10] noted that COVID-19 can be observed in CT images of patients even before some clinical symptoms appear like ground glass opacities, presence of pulmonary nodules, bronchiectasis and round cystic changes [11]. While it has been demonstrated in literature [12, 13] that CT scan tests produce higher sensitivity than the RT-PCR tests, there are

some challenges include shortages of well-trained radiologists and increasing number of patients during a pandemic [14-16]. Furthermore, a new disease like COVID-19 requires radiologists to update their knowledge and learn new interpretation skills. This makes diagnosis even more challenging in remote areas where there is limited access to training resources. It is therefore of extreme importance to urgently develop artificial intelligence (AI) systems to assist in the diagnosis of COVID-19.

In recent years; there has been a rapid surge in the development of deep learning models for the medical image segmentation like skin lesions, lungs, tumors and medical image classification can be used to determine the presence or absence of a particular disease [17-20]. These can aid radiologists in diagnosis, monitoring disease progression, reduction in inspection time, and improvement in accuracy [21]. A systematic review from the authors of [22] has shown that the diagnostic performance of deep learning models may be equivalent to that of health-care professionals.

In this work, a deep learning based system is proposed for COVID-19 detection based on two main stages. The first stage is segmentation and the other is classification. First, some preprocessing techniques were applied to the original images and then the segmentation of the ROI is performed using a new cGAN-UNet hybrid method. After that, several binary classifiers were used to classify whether the CT scan is positive or negative for COVID-19. In this study, in addition to using various methods that increase the performance and stability of GANs; detection of the presence of coronavirus

with the dual-part mask created using cGAN-UNet was performed using a binary convolution neural network (CNN), a PatchCNN and Capsule Neural Network (CapsNet). The results obtained revealed that COVID-19 CT scans can be classified using conditional adversarial networks (cGAN), with high success.

2. RELATED WORK

In the past few years, machine learning has become one of the most popular and accurate methods for automated medical image analysis [17]. Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called “artificial neural networks” (ANNs). Convolutional Neural Networks (CNNs) are deep learning algorithms that have mostly been used recently because they can automatically learn feature representation from a given image hierarchically. They can learn low-level to high level representations of an image and can then assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. In this section, recent research that mainly focused on COVID-19 segmentation and classification in medical imaging are given.

2.1 Research on COVID-19 Segmentation

The goal of medical image segmentation is to identify ROIs such as organs or medical abnormalities by labeling each pixel in a medical image. Segmentation is a crucial step especially in COVID-19 image analysis as it can aid radiologists in diagnosis, disease progression monitoring, and improvements in speed and efficiency. However, for deep learning models to generalize well, they should be trained using large amounts of data and in certain tasks like image segmentation, the data must be well-labeled data.

The highly contagious nature of COVID-19 makes the data collection process particularly challenging as medical staff have to be well protected and equipment used in the process have to be frequently disinfected [21]. In addition, some datasets are not public. Despite these challenges, several approaches have been proposed for COVID-19 segmentation.

Due to the lack of annotated medical images, some researches involve unsupervised and semi-supervised methods. Zheng et al. [23] presented an unsupervised learning technique that generates pseudo segmentation masks from COVID-19 CT scans. By equipping the convolutional blocks with the so-called bottleneck blocks, Shan et al. [24] used a VB-Net to segment COVID-19 infection regions in CT scans. Fan et al. [25] proposed a semi-supervised approach called Inf-Net for identification of infected regions in CT slices.

The encoder-decoder architecture is commonly used for semantic segmentation [26]. In encoder-decoder architecture, the encoder learns the feature representations of the input while the decoder takes this feature representation, recovers the location information that was lost during the pooling process and produces a binary mask. One of such well-known architectures is UNet which retains the vital information from input images by adding skip connections between the encoding and decoding layers [27]. UNet and its variants have been developed for the segmentation of COVID-19. The 3D UNet [28] replaces the layers of the conventional UNet with a 3D version, thereby using the inter-slice information for

segmentation. Zhou et al. [29] proposed a UNet based segmentation network that incorporates the attention mechanism to extract useful features from the encoders. They used spatial and channel attention to re-weight the feature representation and capture rich contextual relationships. But the result dice score was only 83.1%. After, Zhou et al. [30] proposed UNet++ where they modify UNet by inserting a nested convolutional structure between the encoding and decoding path. This network is adopted in Chen et al. [31] for lesion segmentation. They introduced a deep learning algorithm for automated segmentation of multiple COVID-19 infection regions. In order to learn a robust and expressive feature representation they used aggregated residual transformations and applied soft attention mechanisms. After segmenting the lesion, a 3-consecutive slice and quadrant based post-processing methods was applied to determine positive and negative scans. The obvious drawback to UNet architectures is that learning may slow down in the middle layers of deeper models, so there is some risk of the network learning to ignore the layer.

Wu et al. [32] developed a Joint Classification and Segmentation (JCS) system that performs real-time and explainable COVID-19 diagnosis. But their model achieved only a dice of 78% on a large scale dataset of 144167 CT images which the authors created.

Another popular architecture for semantic segmentation is the Fully Convolutional Network (FCN) [33] where the fully connected layers of the CNN are removed. Milletari et al. [34] used a FCN with residual blocks as the basic convolutional block. Their proposed architecture called V-Net is made for 3D volumes and Dice loss is used to optimize the network. FCN classifies each pixel in a spliced image as spliced or authentic. But it often loses or smooths detailed structures and ignores small objects. In turn, Oktay et al. [35] brought forward an Attention UNet that uses advanced attention mechanisms to capture fine structures in COVID-19 images. The network was capable of segmenting particularly variable small size organs such as the pancreas.

Several approaches using Generative adversarial networks (GANs) also have been proposed for image segmentation. However, these approaches are based on augmenting the dataset by synthesizing COVID-19 CT scans. Zhang et al. [36] proposed CoSinGAN which can be learned from a single radiological image with the annotation mask of infected regions. The model was able to synthesize high-resolution and diverse images that match the input conditions.

2.2 Research on COVID-19 classification

Medical image classification aims to label a complete image to predefined classes and can therefore be used for diagnosis. Many approaches have been proposed in literature to classify COVID-19. Zhang et al. [37] propose a ResNet based model that performs anomaly detection and classification between COVID-19 and non-COVID-19 X-ray images. The anomaly score is used to optimize classification on the dataset that contains 10078 images. The model produces a sensitivity, specificity and AUC score of 96.0%, 70.7% and 95.2%, respectively.

Loey et al. [38] implemented a GAN with deep transfer learning for the detection of coronavirus. They used the GAN to generate more images from publicly available X-ray images. These images were then classified in the first scenario using AlexNet, ResNet18 and GoogleNet which produce accuracies

of 66.67%, 69.67% and 80.56% respectively. Apostolopoulos and Mpesiana [39] used transfer learning for the detection of COVID-19 with Vgg19 obtaining the best accuracy of 96.78%. After, Yang et al. [40] built an open-source dataset and showed that a CNN model can be used for diagnosing COVID-19. Their proposed model achieved an accuracy and F1-score of 89% and 90%, respectively. Hemdan et al. [41] proposed CovidX-Net which includes seven different transfer learning architectures for the diagnosis of COVID-19. Vgg19 and DenseNet achieved the best performance with 90% accuracy.

Sethy et al. [42] used support vector machines (SVM) to classify features obtained from several convolutional neural networks. In their study, the ResNet50 model with SVM classifier achieved best results with an accuracy of 95.33%. Similarly, Song et al. [43] utilized a Feature Pyramid Network (FPN) and attention modules to extract features from CT images and fed the extracted features into a ResNet-50 model. They obtained an accuracy of 86%. Ozturk et al. [44] presented DarkCovidNet which automatically detects COVID-19 from X-ray images with an accuracy of 98.08%. DarkCovidNet was inspired from Darknet-19 which is the backbone of a state-of-the-art real time object detection system YOLO (You Only Look Once). Covid-Net which proposed by Wang et al. [8] was trained on a large dataset of 13,975 X-ray images. The model architecture makes use of projection-expansion-projection design pattern which provides enhanced representational capacity.

Wang et al. [45] proposed a 2D CNN model that performs classification between COVID-19 and viral Pneumonia. Experimental results showed that the proposed model achieves an accuracy of 73.1%, sensitivity of 74.0% and specificity of 67.0%. Covid-Net, a deep CNN based model was proposed by Wang et al. [8] to detect COVID-19 in X-ray images producing a testing accuracy of 83.5%. Narin et al. [46] on the other hand, employed several transfer learning methods to detect COVID-19. Specifically they used ResNet50, InceptionV3 and an ensemble Inception-ResNetV2 on two different datasets. While ResNet50 attained the best performance with an accuracy of 98%, InceptionV3 and Inception-ResNetV2 achieved accuracies of 97% and 87%, respectively. Additionally, Ghoshal and Tucker [47] presented a Bayesian Convolutional Neural network that improves the classification accuracy of the standard VGG16 model from 85.7% to 92.8%. The authors enhanced explainability by using saliency maps to point out the specific locations that the model focuses on to make its decisions.

Li et al. [48] introduced Covid-MobileXpert which is a lightweight deep neural network based mobile app that can be used for the screening of COVID-19. The model consists of a 3-player knowledge transfer and distillation (KTD) framework that includes a pre-trained attending physician network (DenseNet-121), a resident fellow network and a medical student network. Then, Lahsaini et al. [49] presented a transferred DenseNet-201 model for classification of COVID-19 from CT images that give the classification accuracy of 98.18%. Bargshady et al. [50] used CycleGAN to augment the dataset and then classified the images with a modified Inception model with 94.2% accuracy.

Several proposed methods performed both segmentation and classification. Zheng et al. [32] employed a U-Net model for lung segmentation and the result is used as an input to a 3D CNN to predict the presence of COVID-19. A sensitivity of 90.7%, specificity of 91.1% and AUC of 95.9% was achieved with this model. Similarly, Chen et al. [28] used UNet++ to

obtain segmented lesions from CT images and, to predict COVID-19 and other forms of viral pneumonia. With the proposed model, the reading time of radiologists shortened by 65%. Xu et al. [51] used a V-Net to segment candidate infected regions and combined the resulting output with handcrafted features. This combination was then sent to a ResNet-18 network that attains a classification accuracy of 86.7%. Li et al. [52] used a combination of U-Net and ResNet50 to extract lung regions and performed classification with 96% specificity, 90% sensitivity and an AUC of 96%. Amyar et al. [53] proposed a multi-task deep learning model which jointly classifies, reconstructs and segments COVID-19 lesions from CT images. Their architecture comprised a common encoder, 2 decoders and a multi-layer perceptron. Their model produced a dice coefficient of 96% accuracy. Karakanis and Leontidis [54] proposed a light weight deep neural network that uses synthetic images generated from a GAN to detect COVID-19. An accuracy of 98.7% was obtained from the proposed model.

3. DATASET

In spite of the growing number of COVID-19 infected patients, along with their volumetric CT scans, labeled CT scans are still only available in a limited capacity. Hence, publicly accessible CT scan datasets are very limited. For this reason, we chose to use the lung CT dataset of [55] (Dataset-1) to train and evaluate our proposed network which is, to the best of our knowledge, the first publicly available data-efficient learning benchmark for medical image segmentation.

The first dataset (Dataset-1) [55] contains 20 CT scans as well as segmentations of lungs and infection masks made by experts. There are a total of 2112 slices in the Dataset-1; 1615 are slices diagnosed with COVID-19 while 497 are healthy images. Figure 1(a) shows a sample of a CT scan and its corresponding mask.

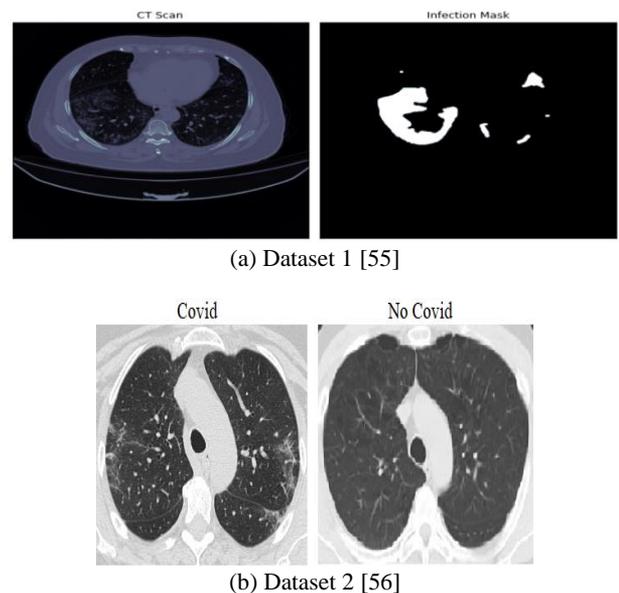


Figure 1. Sample CT images and masks from datasets

The second dataset (Dataset-2) which was proposed by Angelov and Soares [56] was collected from real patients in Sao Paulo, Brazilian hospitals. The detailed characteristic of each patient has been omitted by the hospitals due to ethical

concerns. Dataset-2 contains 2481 CT scans in total. 1252 scans correspond to 60 patients positive for COVID-19 which 32 are male and 28 are female; while 1229 CT scans corresponds to 60 negative patients which 30 are male and 30 are female. But Dataset-2 has not binary masks for image segmentation. In order to test the classification success in Dataset-2, the deep learning structures obtained from the segmentation results using the binary masks in Dataset-1 were used. Figure 1(b) shows a sample of 2 patients, infected and not infected by COVID-19.

The Dataset-1 has been used to train and evaluate the performance of the segmentation model. The resulting segmentation model has been used on the Dataset-2 to generate mask which are then used for classification. Table 1 shows the summary of both datasets.

Table 1. Summary of datasets used in study

	COVID-19	Healthy	Total	Used For
Dataset-1	1615	497	2112	Segmentation
Dataset-2	1252	1229	2481	Classification

4. PROPOSED METHOD

In medical image processing, both segmentation and classification studies can be performed. The purpose of segmentation is to identify Regions of Interest (ROI) such as organs or medical abnormalities by labeling each pixel in a medical image. On the other hand, medical image classification aims to label a complete image into predefined classes, and for this, it first divides the image into specified sub-domains. Thus, only the ROI is examined in medical

images divided into sub-areas, and classification is made according to these areas. However, ROI areas cannot always be obtained in medical images. In such a case, first ROI should be segmented and then the classification algorithm should be applied. In the COVID-19 classification dataset (Dataset-2) used in the study, there are no ROI areas as masked image areas. For this reason, first of all, a successful segmentation algorithm based on deep learning was proposed by using Dataset-1 that has ground truth masked images. After the success of the proposed segmentation algorithm has been proven on Dataset-1, the resulting segmentation model has been used on the Dataset-2 to generate mask which are then used for classification. So, ROI of lung was segmented and According to the classification results, images could be classified more accurately in a short time compared to the literature. Since both segmentation and classification were made within the scope of the study, the literature studies and results in these two fields were examined separately and comparisons were made with the results.

Our proposed method can be summarized in two main stages (Figure 2) as segmentation and classification. The data was preprocessed before feeding it into the model.

After preprocessing, the data has been fed into a deep learning model to obtain a binary mask of lesions in COVID-19 CT images in Stage-1. For this purpose; UNet, UNet++, SegNet and PsPNet were used both separately and as hybrids with GAN, to automatically segment infected areas from chest CT slices. The resulting binary mask was then fed into Stage-2, where a CNN, PatchCNN and CapsNet models were used to predict whether or not a patient has COVID-19 or not. The binary masks that contain COVID-19 have white areas on a black background while those without COVID-19 are completely black. Steps of the algorithm are given in Figure 3.

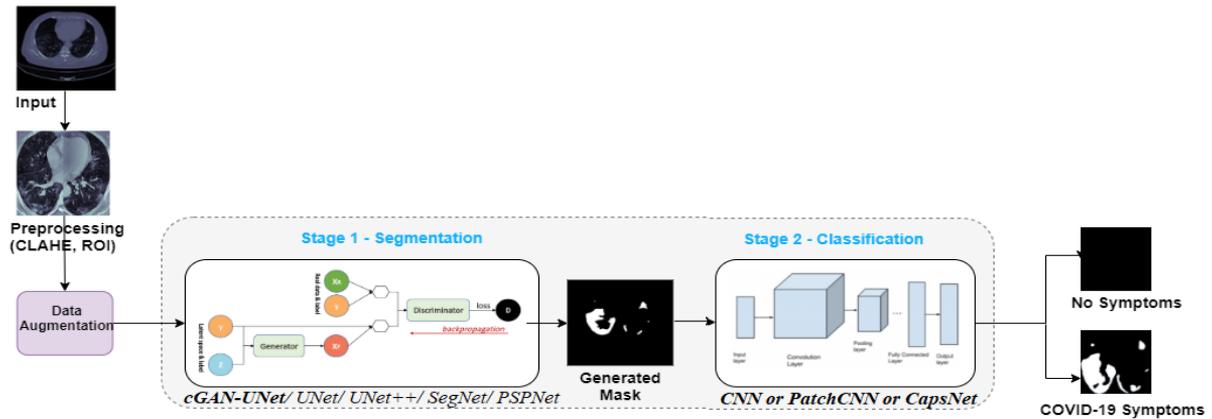


Figure 2. Steps of the proposed method

Algorithm-1

- 1) Acquiring images from Dataset-1
- 2) Extracting the image slices from original images as follows: $x \in D: \{x_i\}_{i=1}^{2112}$
- 3) Converting the images to gray-scale, resizing to 256x256 pixels and normalizing with min-max scaling. ($x_{pre} = \text{Preprocessing}(x)$)
- 4) Applying CLAHE to enhance the images ($x_{CL} = \text{CLAHE}(x_{pre})$)
- 5) Extracting ROI by drawing contours over image and then cropping out the closed boundary ($x_{ROI} = \text{ROI}(x_{CL})$)
- 6) Applying data augmentation strategies using standard methods like rotation, shearing, zooming and horizontal flipping
- 7) Splitting data into 80% for training and 20% for testing.
- 8) Constructing the proposed cGAN-UNet model, training the model using the training dataset to get segmentation masks that will be used as input to our classifiers.
- 9) Evaluating, testing and comparing cGAN-UNet segmentation model to other baseline models.
- 10) Using trained model to generate segmentation masks on Dataset-2 and converting them to binary masks
- 11) Training the CNN / PatchCNN/ CapsNet model using generated binary masks
- 13) Evaluating, testing and comparing the performance of all classifiers.

Figure 3. Steps of the algorithm

4.1 Preprocessing

Before the first stage, several preprocessing steps are applied to the input images. First, 2D slices are extracted from the provided CT scans. Then, images are converted to gray-scaled and normalized. Generally, medical image analysis is challenging due to inherent characteristics of medical images like low contrast, noise, signal dropouts and complex anatomical structures [57]. In order to solve this problem, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to the CT scans. It enhances small details, textures and the local contrast of the images [58]. CLAHE is a variant of adaptive histogram equalization where the contrast amplification is limited to reduce noise amplification. This is done by clipping the value of a histogram at each pixel value and then redistributing it equally among all histogram bins [59].

Most of the images in the dataset contained a lot of black space and sections below the lungs which could lead to wastage of computing resources. The proposed solution to this problem was extracting the ROI by drawing contours over the image and then cropping out the rectangle with the largest closed boundary. The resulting ROI encompasses both lungs in the image. Furthermore, the ROI images are resized to 256

x 256 pixels and then normalized to pixel values between [0,1] using min-max scaling. To avoid overfitting and increase the number of training images data augmentation strategies like rotation, shearing, zooming and horizontal flipping were applied to the training data. The corresponding segmentation masks were also cropped, resized, normalized and augmented. Figure 4 shows the outputs of the preprocessing stage.

4.2 Segmentation step

Image segmentation is an image processing problem that aims to grouping the pixels according to the similarities or dissimilarities (differences). Segmentation makes the image more meaningful and provides faster further stages [60].

Conditional Generative Adversarial Networks (cGAN) was used to segment COVID-19 lesions from CT images in this study. The generative network learns to recognize our ROI and creates a binary mask that outlines its boundaries. The discriminator on the other hand, learns to differentiate between the real segmented masks and the synthetic ones.

In this section, the overall architecture of the GAN, cGAN, the generator, discriminator and how they work are presented. The proposed architecture was inspired by the Pix-2-Pix GAN [61] which is used for image to image translation.

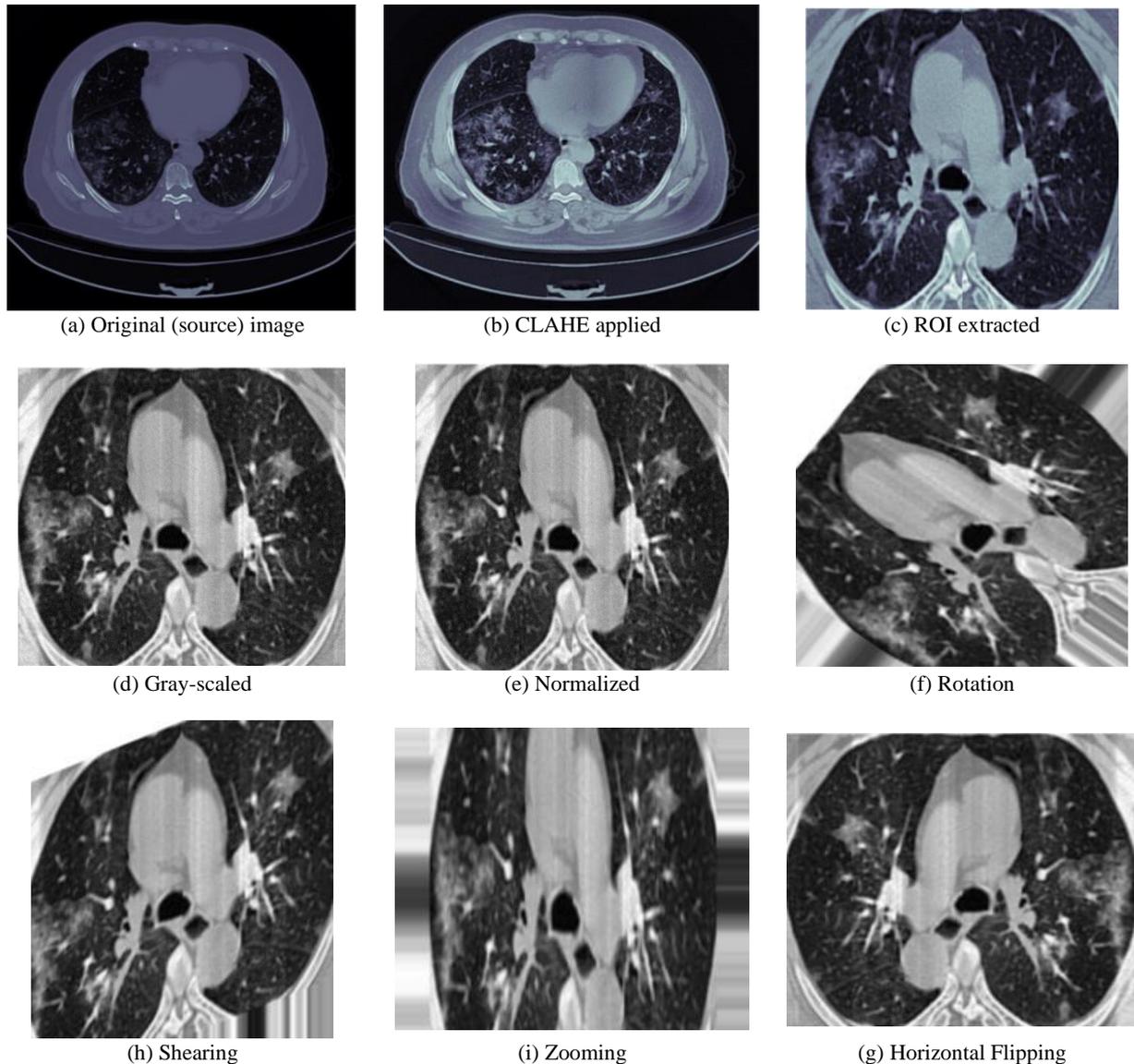


Figure 4. Steps in the order of pre-processing

4.2.1 Generative Adversarial Network (GAN) and Conditional GAN (cGAN)

A GAN is composed of a generator and a discriminator that are trained to be competitive with each other. Vanilla GAN that is basic type of the GAN can be seen in Figure 5. The generator, G ; is trained to produce realistic fake data $G(z)$ from the distribution of the real data (x) and the random noise vector z , in order to fool or confuse the discriminator. The goal of the discriminator, D , on the other hand, is to distinguish between the real data and the fake data produced by the generator. The generator is updated only through gradients back-propagated from the discriminator. By iteratively updating both the generator and discriminator, equilibrium is reached where the discriminator is no longer able to tell the difference between the real and fake data produced by the generator. In Mathematical terms, D and G play a two-player minimax game with objective function given by Eq. (1) [62, 63].

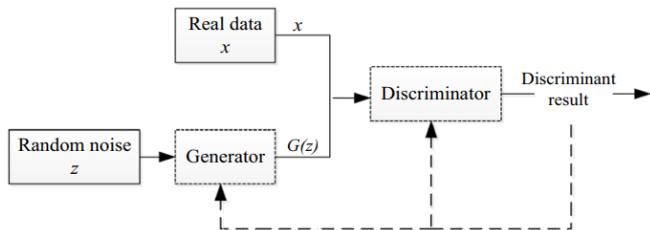


Figure 5. Vanilla GAN [64]

In Eq. (1), L is the objective function, G is generator, D is discriminator, E is the expected value, x is the input image, $P_{data}(x)$ is the distribution from which training examples are drawn and z is random noise. The discriminator D is trained to maximize the probability of correct label assignment to fake and real data while the generator G is trained to fool the discriminator by minimizing $\log(1-D(G(z)))$ (that is the log of 1 minus the discriminator output of the generated/fake data). Hence, the gradients that are back propagated through the discriminator are used to update the generator. Much stronger gradients can be obtained in earlier iterations, if the generator is optimized to maximize $\log(D(G(z)))$ instead of minimizing $\log(1-D(G(z)))$ [65]. This is an indirect way of optimizing and the advantage with this method is that it ensures that the generator doesn't memorize the images. Even though having many advantages, GANs are known that training is challenging because they are generally unstable and sometimes fail to converge [66]. In addition, GANs depict a phenomenon called model collapse where the generator generates a limited diversity of samples, or even the same sample, regardless of the input. There are numerous variations of GANs that have been tried to overcome that limitations that vanilla GANs, and cGAN is one of these variations.

One limitation of GAN is that it has no control on the actual image generation. The cGAN proposed by Mirza and Osidero tries to overcome this problem by incorporating additional information like class labels in the image generation process [67]. In the cGAN the generator has 2 inputs; a random noise z and some prior information c . Then the prior information c is fed into the discriminator together with the corresponding real or fake data. The authors depicted that this change in architectural structure, cGAN not only increased the generation of detailed features, but also enhances stability during training. Because of this, cGAN is used in this study. Eq. (2) shows the objective function of the cGAN framework. Also, the difference in the architecture between GAN and

cGAN is given in Figure 6.

$$L_{GAN}(G, D) = \mathbb{E}_{x \sim P_{data}(x)} [\log(D(x))] + \mathbb{E}_{x \sim P_{data}(x)} [1 - \log(D(G(z)))] \quad (1)$$

$$L_{cGAN}(G, D) = \mathbb{E}_{x \sim P_{data}(x)} [\log(D(x|c))] + \mathbb{E}_{x \sim P_{z(z)}} [1 - \log(D(G(z|c)))] \quad (2)$$

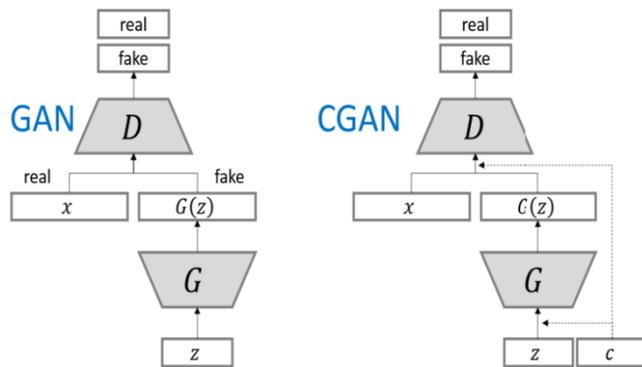


Figure 6. Difference of the architecture between GAN and cGAN [68]

4.2.2 Generator architecture

The generator of cGAN used in this study is a modified U-Net architecture hence the name (cGAN-UNet). U-Net which was introduced in [24] is an architecture that is popularly used in biomedical image segmentation. It is a "U" shaped architecture that is made up of 3 main sections as the encoder, the bottleneck and the decoder sections. The encoding section in the generator of the cGAN-UNet architecture follows typical convolutional neural network architecture and is made up of many encoder blocks. Each encoder block passes an input image through one convolutional layer, followed by batch normalization and a leaky rectified linear unit (slope 0.2). Unlike the original U-Net, all max pooling layers are removed. After each block, the number of kernels is doubled so that the architecture can learn complex structures effectively.

The bottleneck section connects the contraction and expansion layer. It is made up of one 4x4 convolutional layer with a 2x2 stride, followed by a ReLU activation layer. This is often referred to as skip connections. Just like the contraction section, the expansion section is made up of several blocks. In each decoder block the input image is passed through one deconvolutional layer, followed by batch normalization, a dropout layer, a concatenate layer and finally an activation layer. Unlike the encoder blocks, here, the number of feature maps is reduced by half in order to maintain symmetry. In the concatenate layer, the input to the corresponding contraction layer is appended. This makes sure that the features that were learnt during contraction are used to reconstruct the new image. It should be noted that the number of encoder blocks is the same as the number of decoder blocks. The result from the expansion block is then passed through another deconvolutional layer where the number of feature maps is equal to the desired number of segments or classes. All convolutional and deconvolutional layers are defined with a 4x4 kernel and a stride of 2x2 which down sample and up sample the feature maps respectively. This architecture is displayed in Figure 7.

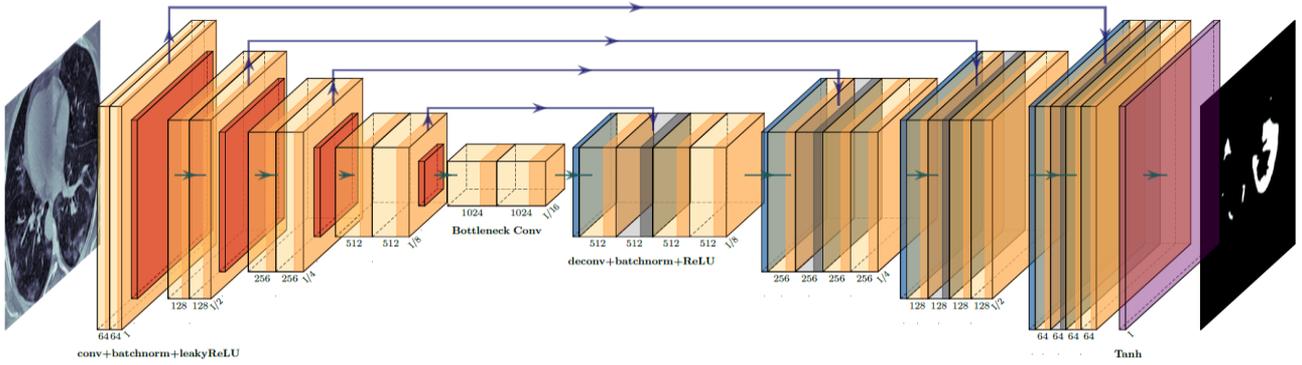


Figure 7. Generator architecture in the study

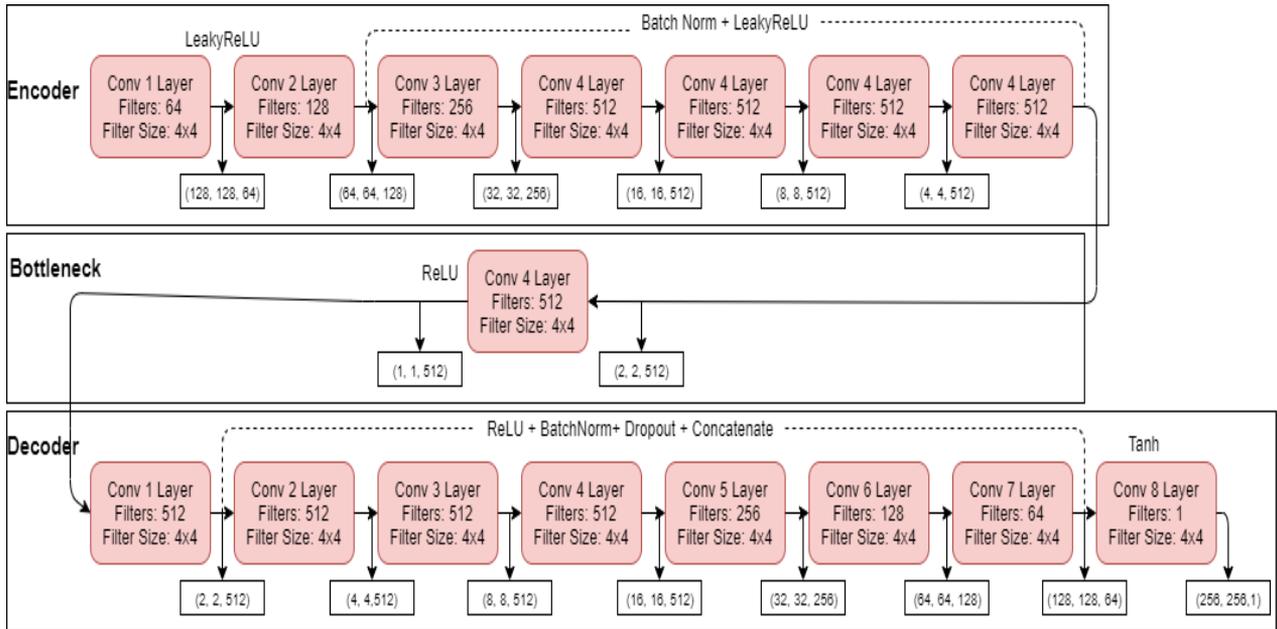


Figure 8. Size of feature vectors in input / output layers of the generator model

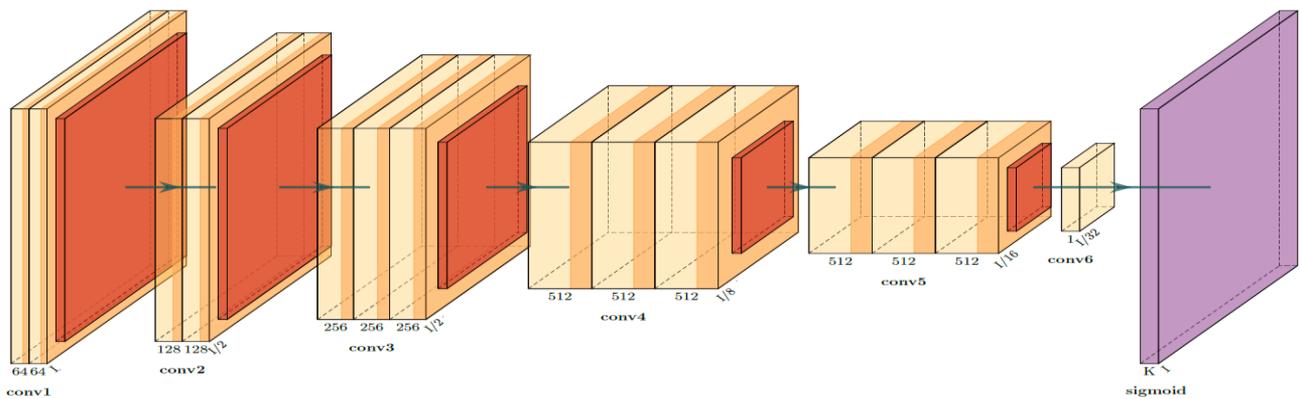


Figure 9. Discriminator architecture in the study

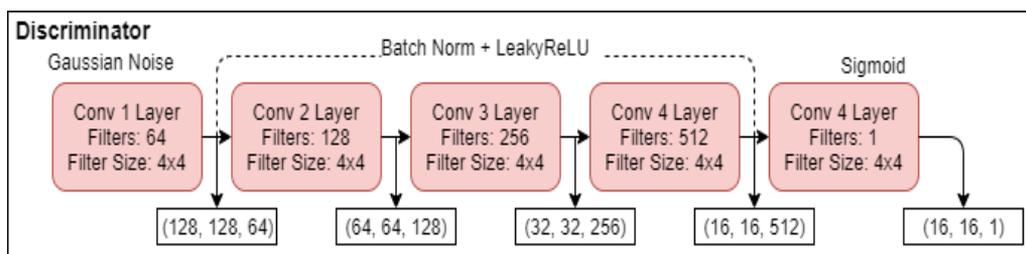


Figure 10. Size of feature vectors in input / output layers of the discriminator model

To improve the performance of cGAN-UNet, several strategies from [66-69] were applied in this study. These strategies were included as batch normalization, using leakyReLU, random Gaussian weight initialization with a standard deviation of 0.002 in both the discriminator and generator. In addition, L2 regularization was added to all layers of the discriminator. Furthermore, label smoothing was applied on all class labels. In label smoothing, rather than use hard labels like 1 and 0 to represent real and fake labels, soft labels were used by generating random values between 0.9 and 1.0 to represent real images and values between 0 and 0.1 to represent fake images. After implementing these changes, there was a noticeable change in how fast the model converged. Sizes of the all feature vectors both input and output layers of the generator is given in Figure 8.

4.2.3 Discriminator architecture

The discriminator determines if a given binary mask is real or fake. The discriminator used in this study is called a ‘‘patch discriminator’’ [61]. The patch discriminator divides the input image into a set of patches and maps each patch to a single scalar output. Unlike a normal image discriminator which performs prediction on an entire image, a patch discriminator makes a prediction for each patch and the final prediction is the average of all patch predictions. In addition, the patch discriminator requires fewer parameters, it works well with very large and blurry images and the computational time is lower. In the study, the patch discriminator uses a patch size of 70x70 pixels. The authors of [61] showed that a 70x70 patch size is a good choice as it can accommodate global features. The implemented model takes in 2 input images that are concatenated together. To prevent overfitting, Gaussian noise (standard deviation of 0.2) is added to this concatenated input before passing into the first hidden layer. The model consists of 3 hidden layers, each followed by batch normalization and leaky ReLU. Each convolutional layer has 4x4 filters and a stride of 2. In addition, the discriminator is regularized by constraining the norms of its gradients using L2 regularization. Therefore, an L2 regularizer is added to all convolutional layers. The discriminator architecture is given in Figure 9. Sizes of the all feature vectors both input and output layers of the discriminator is given in Figure 10.

The discriminator is optimized using a combination of the weighted binary cross entropy (bce) loss (Eq. (3)) and the dice loss (Eq. (4)). The discriminator loss is calculated using (Eq. (5)) and this loss back propagated through the generator. The cross-entropy loss is the most widely used loss for CNN classification [70].

The generator is updated using the discriminator loss and the L_1 loss (Eq. (6)) with a weighting of 1 to 100 in favor of the L_1 loss as recommended by the authors of [61]. The final objective function for the generator (with $\lambda=100$) is shown in Eq. (7).

$$L_{BCE} = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (3)$$

$$L_{DICE} = 1 - \frac{2x(y \cap p(y))}{y + p(y)} \quad (4)$$

$$L_{DIS} = L_{BCE} + L_{DICE} \quad (5)$$

$$L_{L1}(G) = \mathbb{E}_{x,y,z} \|y - G(x, z)\| \quad (6)$$

$$L_{GEN} = \arg \min_G \max_D L_{CGAN}(G, D) + \lambda L_{L1}(G) \quad (7)$$

where, y is the ground truth, $p(y)$ is the predicted mask for all values of N that is the number of images in the dataset and L_{DIS} is the discriminator loss, λ is the constant used to increase the weight of L_1 loss, L_{L1} is the L_1 loss, L_{GEN} is the generator loss and L_{CGAN} is the adversarial loss as shown in Eq. (2).

The generator and discriminator are optimized simultaneously where the generator learns to create a reasonable binary mask and the discriminator learns how to differentiate between synthetic and real segmentation. Besides introducing the weighted bce dice loss, Gaussian noise and L2 regularization, different learning rates and loss functions were used. Adam is found to be the best optimizer with a learning rate of 0.0001 and momentum of β_1 of 0.5 and β_2 of 0.999. However, the model was trained for 130800 epochs to achieve best results.

It should be noted that after generating the segmentation masks, a simple thresholding was performed to generate binary masks. 54.4 was selected as the best threshold value at which the 92.23 F1-score value was obtained as shown in Figure 11.

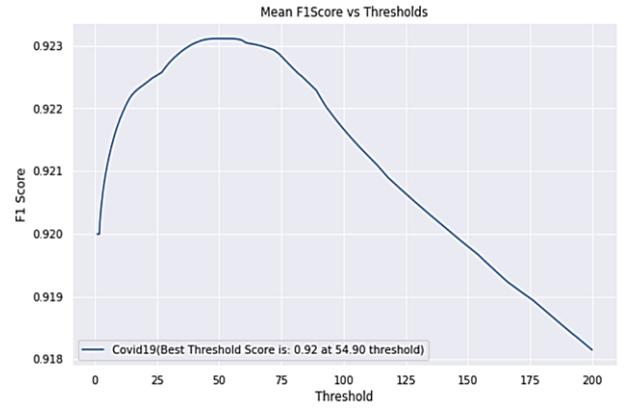


Figure 11. Threshold to F1-score graph

Training parameters of the cGAN-UNet used in the study also given in Table 2.

4.3 Classification step

After generating segmentation maps, they are used as input to a several classifiers to predict whether the masks have COVID-19 or not. Before feeding the synthetic masks into the classifiers, simple preprocessing and data augmentation methods like rotation, shearing, zooming and Gaussian blurring was applied as shown in Figure 12. Three different classification methods are implemented and their performance compared with experimental results. The architectures of these methods are presented the following section.

Table 2. Training parameters for cGAN-UNet

Number of epochs	130800
Optimizer	Adam
Learning rate	0.0001
Momentum of β_1 and β_2	0.5 and 0.999
Loss function	Binary cross entropy loss, dice loss and L_1 Loss

4.3.1 Convolutional Neural Network (CNN)

CNN's are popularly used for image classification as they accumulate sets of features (such as edges, corners, shapes etc) at each layer. As shown in Figure 13, the CNN architecture is made up of 5 convolutional layers and 3 dense layers. While each convolutional layer is followed by batch normalization and a max pooling layer, the dense layers are followed by dropout layers. The ReLU activation function is applied on all layers except the final layer that uses sigmoid as an activation function. The convolutional layers use 3x3 kernels and have a stride of 1. The network is trained for 150 epochs. Batch

normalization and dropout layers are added to avoid overfitting. The feature vectors of the CNN are given in Figure 14 and training parameters in Table 3.

Table 3. Training parameters for CNN and PatchCNN

Number of epochs	50
Optimizer	Adam
Learning rate	0.0005
Momentum of β_1 and β_2	0.5 and 0.999
Loss function	Binary cross entropy loss

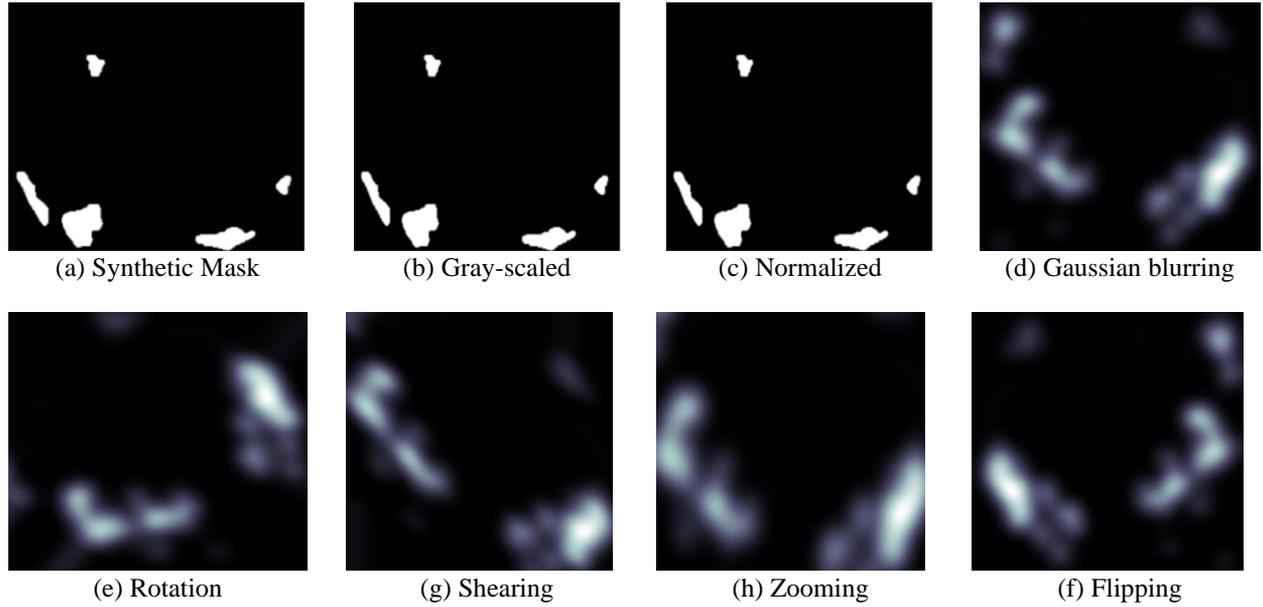


Figure 12. Preprocessing of synthetic mask

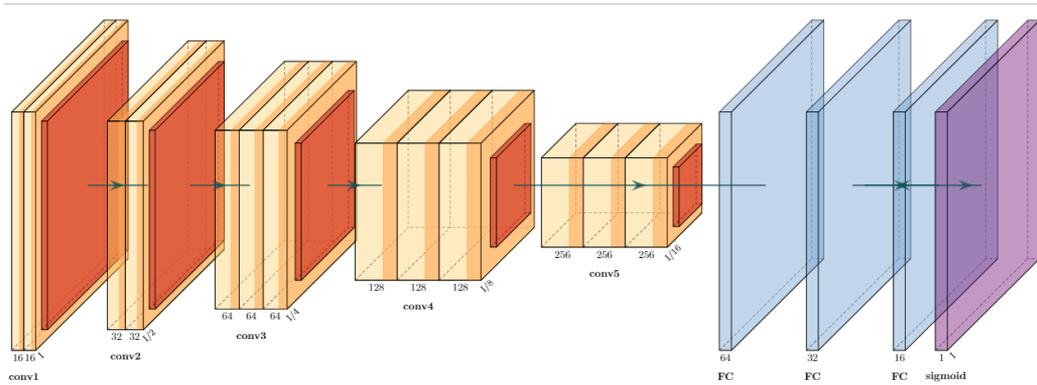


Figure 13. Convolutional neural networks

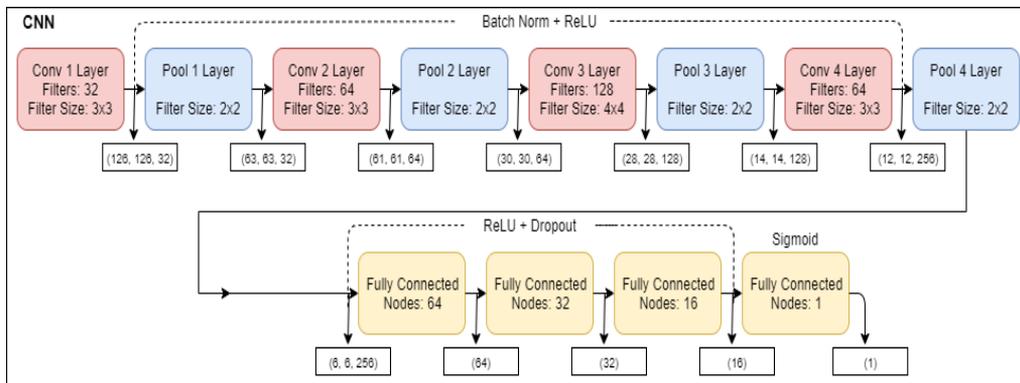


Figure 14. Size of feature vectors in input / output layers of the CNN and PatchCNN model

4.3.2 Patch Convolutional Neural Networks (PatchCNN)

The PatchCNN architecture is very similar to the CNN architecture. However, unlike like CNN where the entire image is sent into the network, several patches are extracted from an image then each patch is sent into the network. In the proposed method, the class label for each patch is predicted by the model, then a majority voting scheme is applied.

The class that receives the most votes is the final class of the image. Several patch sizes were tried on the synthetic images generated from the cGAN-UNet model whose size is 256 x 256 pixels. The patches are created by dividing the synthetic image into smaller images of a specific patch size. As can be seen in Figure 15, the patch sizes we experimented with are 16 x 16, 32 x 32, 64 x 64 and 128 x 128 pixels.

As shown in Table 4, the bigger the patch size, the better the model performance. This indicates that smaller patch sizes do not carry sufficient diagnostic information while larger patch sizes carry sufficient discriminating features. According to the patch size success, all PatchCNN models were trained with a patch size of 128. Hence, model input was 128 x 128 pixels.

The training parameters for this model are the same as the CNN as shown in Table 3 while the feature vectors can be seen in Figure 15. Figure 16 shows the model architecture and a synthetic image after it has been split into different patch sizes.

Table 4. Comparative analysis of different patch sizes on cGAN-UNet Synthetic Masks

Patch Size (pixels)	Accuracy
16 x 16	50.59%
32 x 32	68.41%
64 x 64	89.93%
128 x 128	92.56%

4.3.3 Capsule Neural Network (CapsNet)

CNNs have several pooling layers which tend to lose information such as the objects precise position and pose. CapsNet was introduced by Sabour [71] in 2017 to overcome these drawbacks in CNNs using the idea of inverse graphics. Unlike CNNs, CapsNet preserves detailed information about an object's position and pose throughout the network. A capsule network is made up of capsules; a capsule is a group of neurons that learn to detect an object or parts of an object in an image. Its output is a vector whose length represents the presence of an object in an image and whose orientation could represent the position, rotation, size or scale of the object in the image. This is unlike CNNs that ignore the spatial relationships between features.

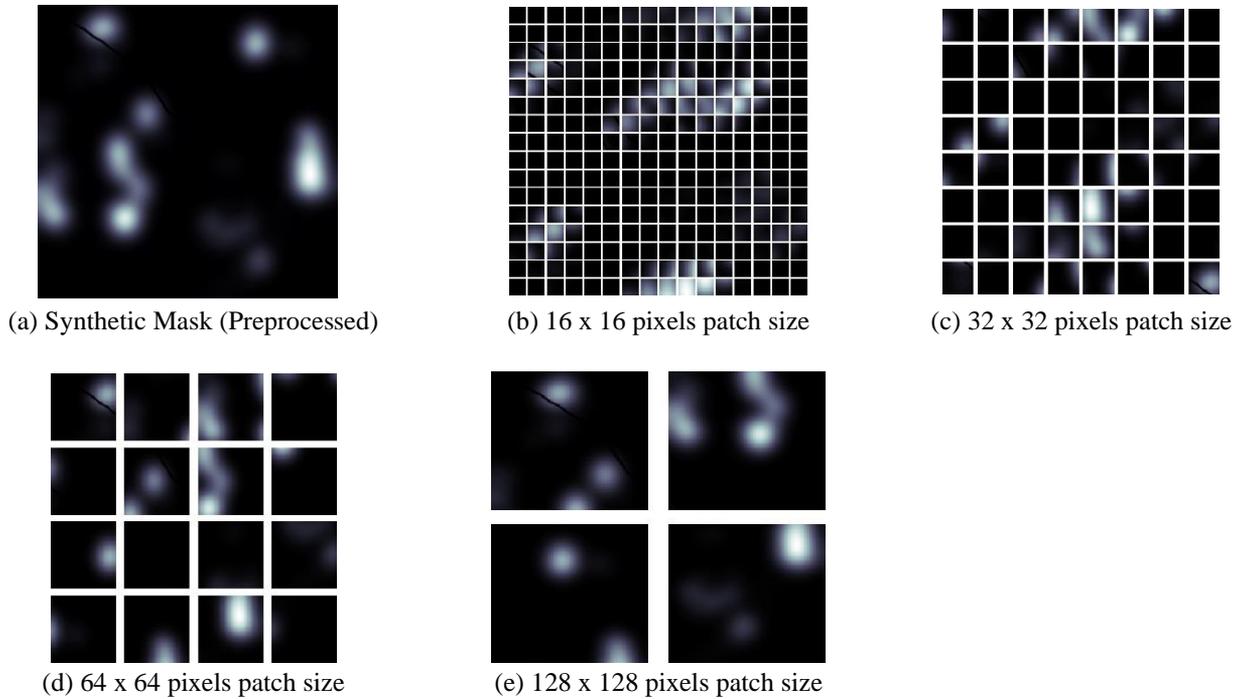


Figure 15. Input preprocessed image and the different patches extracted from it

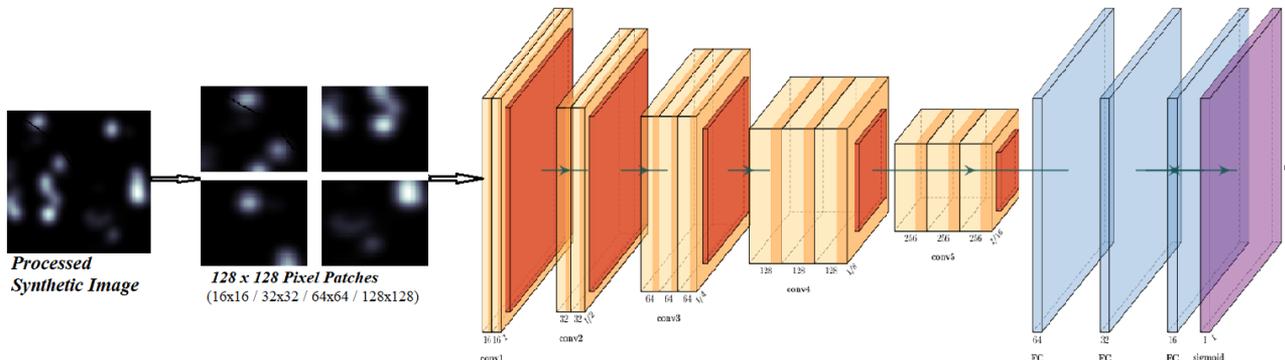


Figure 16. PatchCNN Architecture with a 128 x 128 pixel Patch size

The CapsNet architecture is made up of an encoder and a decoder. The encoder contains 3 layers which include Convolutional Layer, Primary Capsule Layer and ClassCaps Layer. The Convolutional Layer has 256 kernels with size $9 \times 9 \times 1$ and stride of 1, followed by the ReLU activation function. In this layer, the image pixel intensities are converted to local feature detectors. The output of this layer is then fed to the Primary Capsule Layer which is also a convolutional layer that contains 32 primary capsules with each capsule having a dimension of 6×6 . The lowest level entities in the image are captured in this layer. The output of this layer is then sent to the ClassCaps layer through a dynamic routing algorithm. Dynamic routing algorithm computes a coupling coefficient to quantify the connection between the Primary Capsule and the ClassCaps layers while facilitating the routing of capsules to the appropriate next layers. The CapsNet model learns through the coefficient value as it is updated during the training process and only 3 routing iterations are used to optimize the loss faster and avoid overfitting. The ClassCaps layer has 2 class capsules, one for each class. An 8×16 weight matrix is applied to each input vector here that maps the input space to the 16 dimensional capsule output space.

The purpose of the decoder network in the CapsNet is to reconstruct the vectors from the ClassCaps layer representation. It does this by minimizing the Euclidean distance between the input image and decoder's output. The decoder is made up 3 fully-connected layers; the first layer has a total of 512 neurons, the second layer 1024 neurons and the third contains 784 neurons. The CapsNet parameters are optimized using a margin loss for each output class. In addition to the margin loss, an L2 reconstruction loss (mean squared error) is used as a regularization method to boost each capsule to encode parameters of the input image. The CapsNet architecture can be seen in Figure 17. A summary of CapsNet architecture and the parameters, used in this study are shown in Figure 18 and Table 5, respectively.

Table 5. Training Parameters for CapsNet

Number of epochs	100
Optimizer	Adam
Learning rate	0.001
Momentum of β_1 and β_2	0.9 and 0.999
Loss function	Margin + L2 Loss

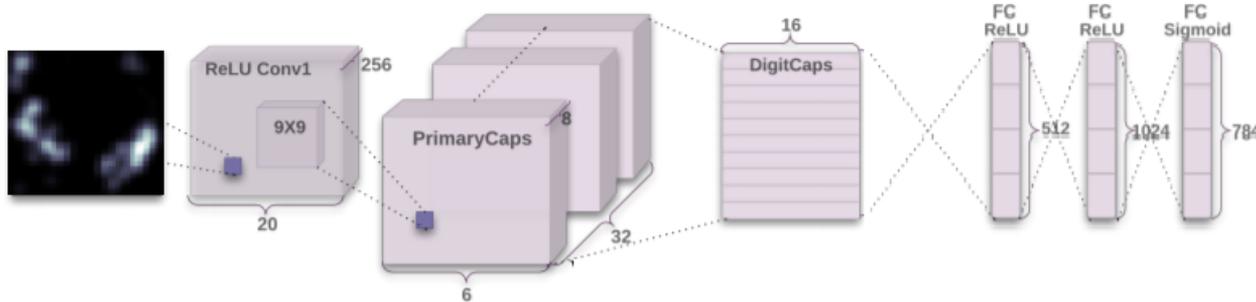


Figure 17. CapsNet Architecture [69]

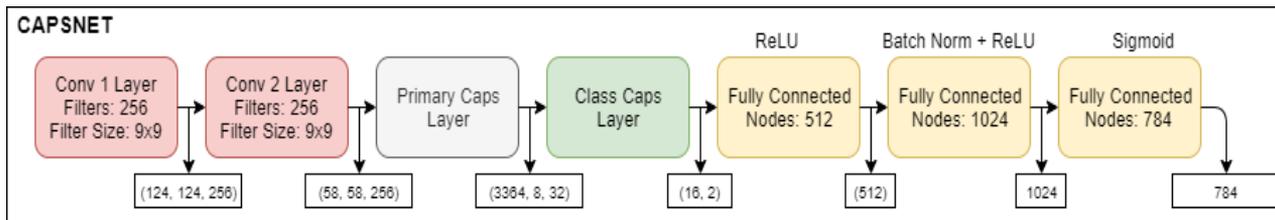


Figure 18. Size of feature vectors in input / output layers of the CapsNet model

5. EXPERIMENTAL RESULTS

In this study, all experiments were conducted on an Intel® Core™ i7-4720HQ CPU @ 2.60GHz × 8 processor, NVIDIA GeForce GTX 960M / PCIe / SSE2 graphics card, supporting NVIDIA GPU Computing Toolkit 10.1 GPU. The application was implemented on Ubuntu 18.04 platform and Google Colab was the chosen integrated development environment (IDE) as it provides a free GPU and a jupyter notebook environment with a Tesla T4 GPU, a RAM of approximately 12.6 GB and disk space of 33 GB. Python is the programming language used because it is cross-platform, vastly used, has an active community of developers and provides easy to use built-in libraries [72]. In addition, the open source machine learning libraries Keras 2.3.1 and Tensorflow 2.0 are implemented in Python. Keras 2.3.1 provides the possibility of easy model

building with APIs like Keras with eager execution which makes debugging easy. Tensorflow 2.0 provides robust machine learning production and finally a powerful experimentation environment for research [73]. In this section, developing platforms, developing context and results for each experiment are presented.

In this study, two separate experiments were conducted to evaluate segmentation and classification models as given in Figure 2. Firstly, UNet, UNet ++, SegNet, PspNet, that are 4 deep learning models used in the literature were used both separately and as hybrids with GAN and to segment the CT images in the Dataset-1. At this stage, the accuracy of the system has been demonstrated by analyzing the segmentation results using the binary masks in the Dataset-1. In the study, the experiments were carried out by dividing the data in the dataset-1 into 80% training and 20% testing. The data on

training and test numbers are given in Table 6. Also, for comparison, results were obtained by applying 5-fold cross validation on the data (Table 7).

Table 6. Numbers of training and testing images for the segmentation of the Dataset-1

	COVID-19	Healthy	Total
Training (80%)	1292	398	1690
Testing (20%)	323	99	422
Dataset-1(Total)	1615	497	2112

The segmentation models were evaluated quantitatively and qualitatively. Dice similarity score (Dice), Intersection Over Union (IoU), precision and recall, and F1-score were used for segmentation success.

The Dice and IoU metrics are well-known and accepted as evaluation methods for medical image segmentation [74]. Higher scores for both metrics depict better segmentation performance. The Dice similarity coefficient, like F1-score is an overlap measure that is used for pairwise comparison of the target mask and segmentation masks and is defined as in Eq.8. The IoU score is a popular metric for measuring the percentage of overlap between the ground truth mask and the predicted mask. It is also called the Jaccard index and works by measuring the number of similar pixels between the target and predicted masks divided by the total number of pixels present in both masks as defined in Eq. (9).

$$Dice = \frac{2 * (A \cap B)}{A + B} \quad (8)$$

$$IoU = \frac{A \cap B}{|A \cup B|} * 100 \quad (9)$$

In Eqns. (8) and (9), where A and B denote ground truth and segmentation masks, respectively.

The precision and recall metrics reflect the similarities of the generated masks and the target. Their measures are based on the number of misclassified and correctly classified pixels and can be expressed in terms of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). While TP defines the outcome where the model correctly predicts the positive class as positive while FP defines the correctly predicts the negative class as negative. On the other hand, FN and TN define the false outcomes for the classes.

To get the value of precision (Eq. (10)) the total number of correctly classified positive examples (TP) is divided by the total number of elements labelled as belonging to the positive class (TP+FP). High Precision indicates an example labeled as positive is indeed positive. However, under-segmentation errors are not reflected in the precision. The recall (Eq. (11)), also known as the sensitivity, can be defined as the ratio of the

total number of correctly classified positive examples (TP) to the total number of elements that actually belong to the positive class (TP+FN). On the other hand, specificity (Eq. (12)) is as the ratio of the total number of correctly classified negative examples (TN) to the total number of elements that actually belong to the negative class (TN+FP). Even though recall does not account for over-segmentation errors in image segmentation [73], a high recall shows that there is a low number of false negatives which is desired in cases like COVID-19 because the disease is highly contagious and spreads easily. The F1-score is the harmonic mean of precision and recall and becomes high only when both precision and recall are high (Eq. (13)).

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN} \quad (11)$$

$$Specificity = \frac{TN}{TN + FP} \quad (12)$$

$$F1\ Score = \frac{2TP}{2TP + FP + FN} \quad (13)$$

Most of the current COVID-19 CT segmentation methods proposed in literature were trained on different datasets, which are mostly private datasets. They were trained in different settings and were evaluated using different metrics [75]. This makes performance comparison across proposed methods challenging. However, for quantitative analysis our results are compared for both segmentation and classification with other methods in literature. Comparisons between proposed model (cGAN-UNet) both with and without cross validation and baseline segmentation models (UNet, UNet ++, SegNet, PspNet) for the segmentation success on the Dataset-1 is given in Table 7. The combination of cGAN with the other baseline models; as cGAN-UNet++, cGAN-SegNet and cGAN-PspNet used in the study, did not learn during the training process and produced completely black images. Therefore, the results of these hybrid methods are not given in Table 7 and they were not used in the next classification step.

According to the results the proposed method outperforms the others in all metrics. Our model is almost 6% above the second best model U-Net which is closely followed by UNet++ and then SegNet. The model with the worst performance metric is the PspNet with only a 51.20% dice score and 37.66% IOU score. The use of a cGAN model with UNet therefore provides a higher accuracy to detect infection regions compared to using other baseline methods alone. Figure 19 shows the discriminator and generator loss of the proposed method during training.

Table 7. Comparisons of the segmentation results on the Dataset-1 (%)

Model	Dice	IOU	Precision	Recall	F1-Score
UNet [24]	86.90	77.74	86.64	86.65	86.29
UNet++ [25]	85.12	75.20	84.11	86.39	84.50
SegNet [76]	77.37	65.08	82.43	74.87	77.37
PspNet [77]	51.20	37.66	61.55	53.96	51.37
cGAN-UNet (proposed method without 5-fold cross validation)	92.31	86.41	92.2	92.70	92.31
cGAN-UNet (proposed method with 5-fold cross validation)	92.21	86.20	92.4	92.03	92.21

The high metric scores obtained from the experiment show that adding adversarial training to existing models like U-Net improves the accuracy of segmentation. Results of the segmented binary masks generated by each model are given in Figure 20. The cGAN-UNet model clearly outputs results that are closest to the ground truth unlike other models whose results differ from the ground truth significantly.

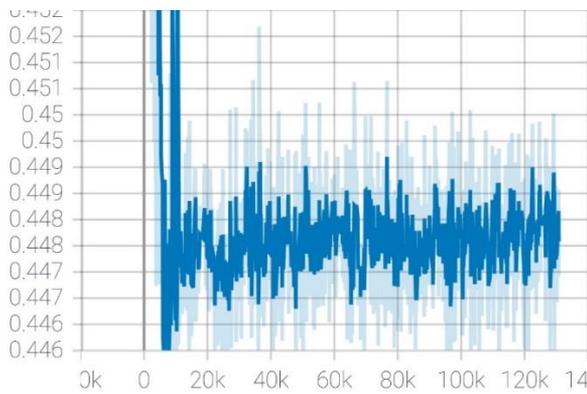
In the classification stage (given in Figure 2), classification was performed on Dataset-2 which contains CT scans of healthy and COVID-19 patients. It is important to note that Dataset-2 does not have the ground truth binary masks for image segmentation. Therefore, the obtained binary masks getting from segmentation stage were used for classification on the Dataset-2 using 3 different deep learning classifiers named CNN, PatchCNN and CapsNet explained in the previous section. Like in segmentation, the dataset is divided as 80% for training and 20% testing as shown in Table 8. Also,

like in segmentation, results were obtained by applying 5-fold cross validation on the data (Table 9).

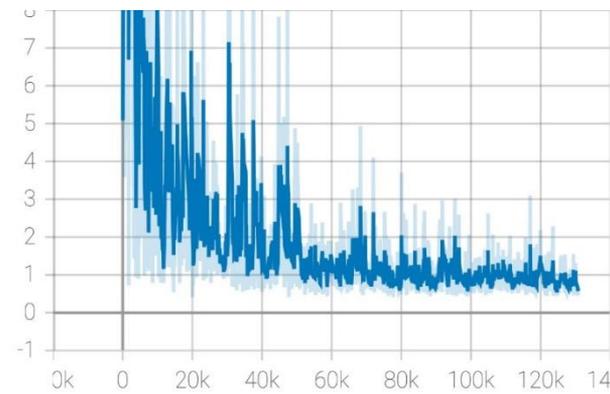
Table 8. Numbers of training and testing images for the classification of the Dataset-2

	COVID-19	Healthy	Total
Training (80%)	1001	983	1984
Testing (20%)	251	246	497
Dataset-2 (Total)	1252	1229	2481

To evaluate the performance of classification models; accuracy was used to also test the success of the systems; in addition to precision, recall, specificity, and F1-score, as in segmentation. Also, the ROC curve was used for the classification success.



(a) Discriminator Loss



(b) Generator Loss

Figure 19. Loss of cGAN-UNet against number of epochs

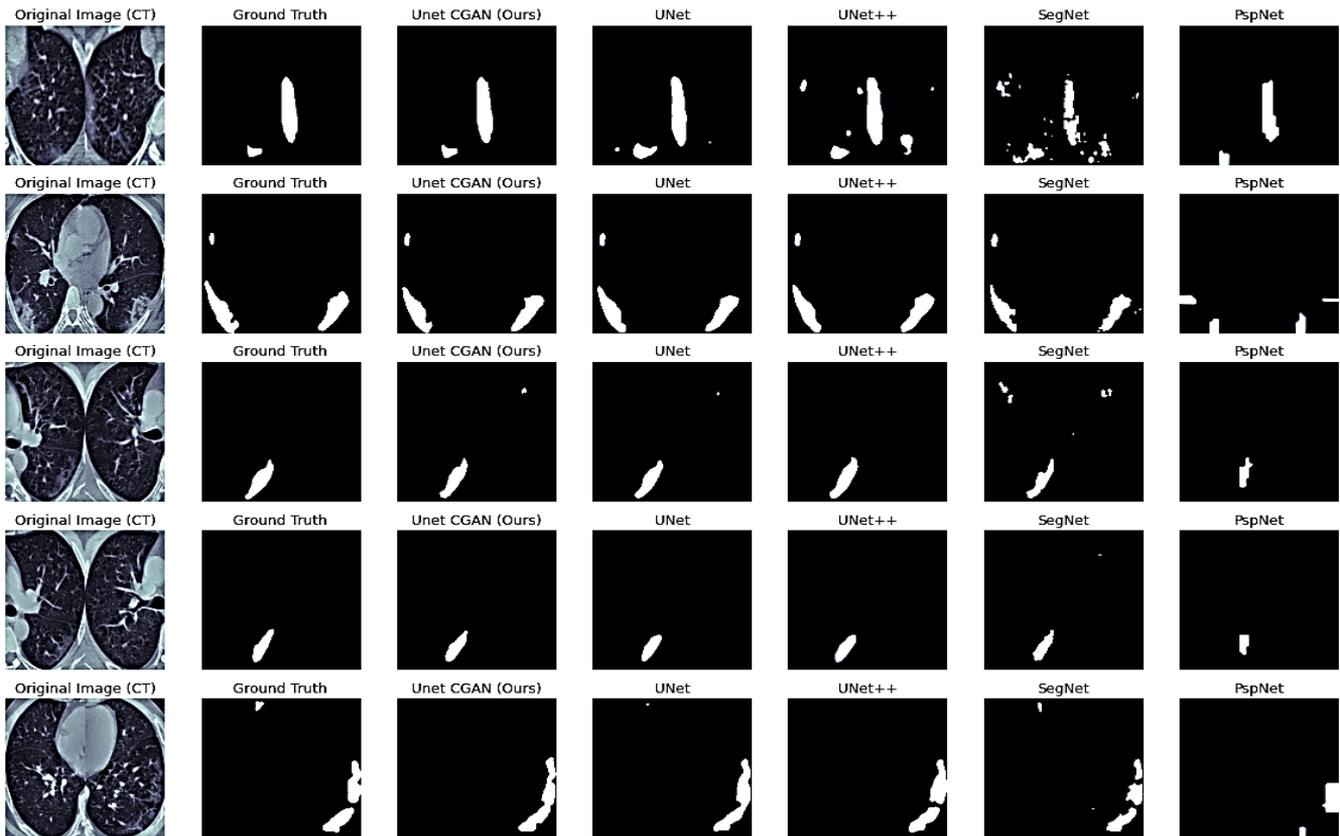


Figure 20. Results of the segmented binary masks generated by each model

The results for the classification using CNN, PatchCNN and CapsNet are shown in Table 9. Comparisons both with and without cross validation for the classification are also given. The binary masks for each segmentation model shown in Table 7 were generated and gave to all classifiers. Additionally, the original CT images were also fed into each classifier to compare with the others.

In Table 9, the cGAN-UNet segmentation achieved the best results with the CNN classifier achieving the best results with an accuracy and F1-score of 99.2%. The PspNet segmentation model produced the worst results with all classifiers. The confusion matrices for the 3 best models for each classifier are shown in Figure 21(a), Figure 21(b) and Figure 21(c), respectively. Figure 22 shows the receiver operator characteristic (ROC) curve of all classification models. The results showed that segmenting lesions from CT scans could distinguish between patients with COVID-19 and healthy.

The results show that binary mask generated from our cGAN-UNet segmentation model produce the best performance regardless of the classifier that is used. Table 4 shows that the performance of the PatchCNN model improved

as the patch size increased. This demonstrates that unlike large patches, sufficient diagnostic information is not carried in smaller patch sizes. However, although the results of the PatchCNN classifier were not bad, the CNN classifier performed much better.

The results obtained from PatchCNN with cGAN-UNet and UNet++ are very close, approximately 92% accuracy. The results from the CapsNet classifier indicate that the model is not well suited for training on binary masks as the results are significantly lower than the other classifiers. With CapsNet, the Original CT model achieved the best accuracy of 78.29%, followed by UNet++ with an accuracy of 75.77% and then by cGAN-UNet with an accuracy of 73.84%. Furthermore, Table 9 shows the difference between training the classifiers on the original CT scans and on binary masks. We see that applying the classifiers on binary mask generated from cGAN-UNet, UNet and UNet++ outperform the models trained on the original CT. On the other hand, results obtained from SegNet and PspNet binary mask, are consistently lower than the original CT images.

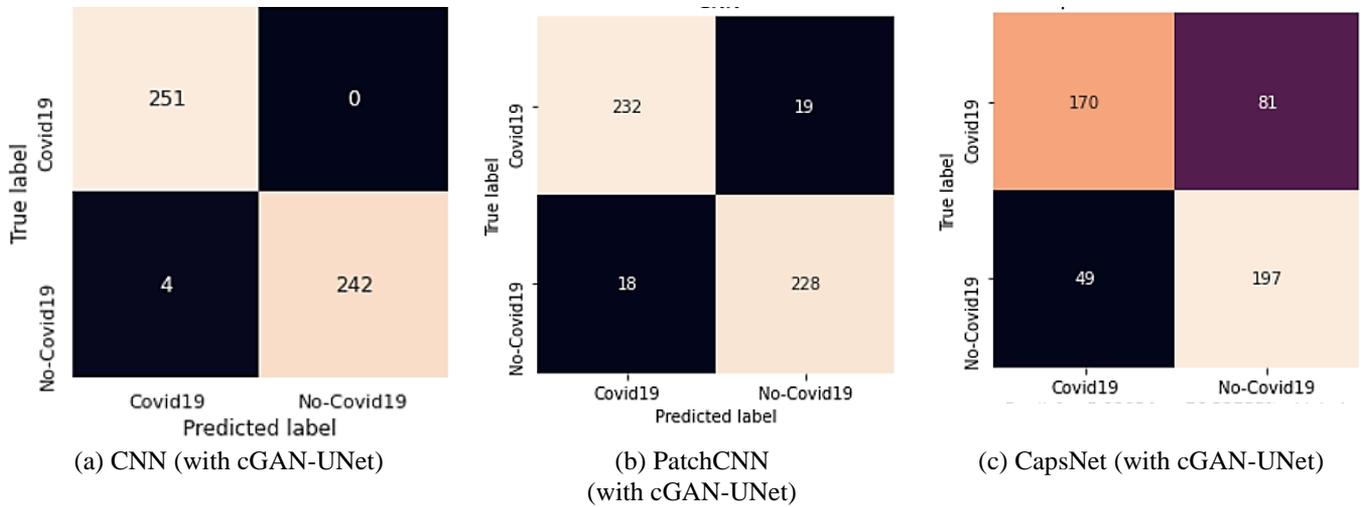


Figure 21. Confusion matrix of classifiers with cGAN-UNet segmentation

Table 9. Comparison of different classifiers on original images and binary masks generated from different segmentation models

Classifier	Method	Accuracy	Precision	Recall	Specificity	F1-Score
CNN	Original CT	90.34	96.34	84.06	96.7	89.79
	UNet	92.76	98.20	87.25	98.37	92.40
	UNet++	97.38	100	94.82	100	97.34
	SegNet	71.43	85.16	52.59	90.65	65.02
	PspNet	50.90	88.88	0.03	99.59	0.06
	cGAN-UNet (without 5- fold cross)	99.20	98.43	100	98.37	99.20
	cGAN-UNet (with 5- fold cross)	98.95	98.99	98.56	98.95	98.76
PatchCNN	Original CT	79.88	72.12	96.75	63.55	82.64
	UNet	90.54	100	80.89	100	89.44
	UNet++	92.26	92.53	92.23	92.40	92.46
	SegNet	58.35	54.52	95.52	21.91	69.42
	PspNet	50.50	26.0	51.02	50.1	34.01
	cGAN-UNet (without 5- fold cross)	92.55	92.80	92.43	92.68	92.61
	cGAN-UNet (with 5- fold cross)	90.95	91.2	90.84	91.06	91.02
CapsNet	Original CT	78.29	76.87	79.54	77.09	78.18
	UNet	64.79	73.75	47.01	82.93	57.42
	UNet++	75.77	77.77	70.87	80.49	74.17
	SegNet	61.77	82.79	30.68	93.50	44.76
	PspNet	50.50	50.50	1.0	0.0	67.11
	cGAN-UNet (without 5- fold cross)	73.84	77.62	67.73	80.08	72.34
	cGAN-UNet (with 5- fold cross)	69.11	75.33	57.62	78.89	62.60

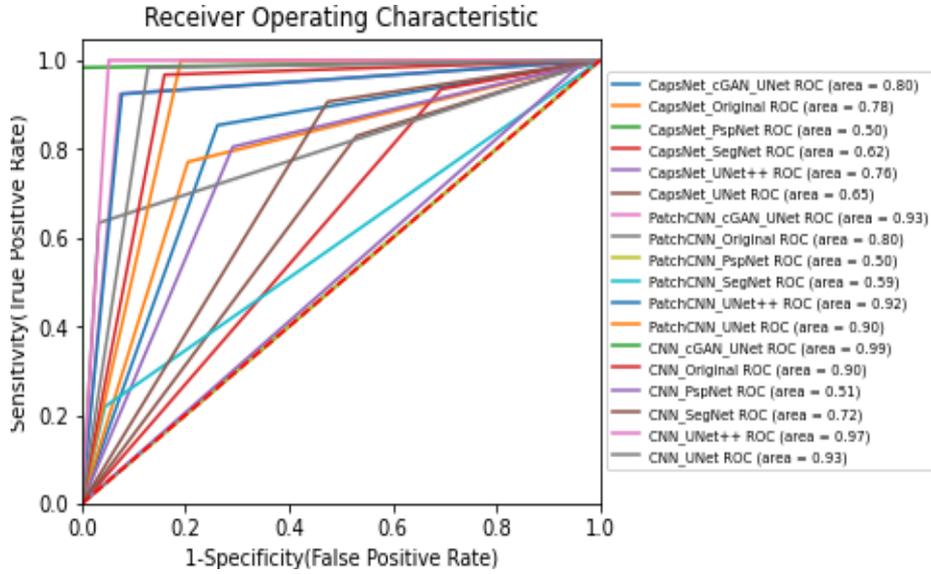


Figure 22. ROC Curve of all models

Finally, we compared our models with the state of the art methods for COVID-19 segmentation and classification. The existing results for segmentation vary from a dice score of 68.2% in [33] to 91.6% in [30] for CT images according to the number (#) of images as shown in Table 10. Table 11, on the other hand shows the state of the art classification models whose results vary from an accuracy of 66.67% in [38] to an accuracy of 98.7% in [54]. This shows that both our segmentation and classification models outperformed state of the art methods.

Table 10. A quantitative comparisons between proposed model and state of the art for the segmentation tasks

Method & Study	# of images	Dice score
Inf-Net [33]	100	68.2%
RC-CoSinGAN [36]	100	71.3 ± 19%
Semi-Inf-Net [33]	100	73.9%
JCS [35] (without image properties)	3855	77.5%
JCS [35] (with image properties)	3855	78.3%
Encoder-Decoder [53]	100	88.0%
VB-Net [30]	300	91.6 ± 10%
cGAN-Unet (proposed)	2112	92.31%

These results therefore suggest that applying the right segmentation model on CT images can significantly improve the model’s performance. In addition, it is understood that adding adversarial training to existing models like UNet improves the accuracy of segmentation.

6. CONCLUSION

With the emergence of COVID-19 at the end of 2019, the world faced a massive health crisis. In this study, different hybrid deep learning methods for the segmentation and classification of COVID-19 were proposed [78]. First of all, a conditional GAN (cGAN) was used to create binary masks that could assist radiologists in diagnosing, monitoring disease progression, reduction in examination time, and improvement in accuracy. After that, to further ease the diagnosis process, several classification models were used as

Convolutional Neural Network (CNN), a PatchCNN and a Capsule Neural Network (CapsNet) for COVID-19 detection. These classifiers take in the generated binary mask as input and predict the presence or absence of COVID-19.

Our proposed methods show very promising results as they outperform baseline and state of the art models. The proposed segmentation method achieved a dice score of 92.31% and an IOU score of 86.41%. For the classification results, the best model attained accuracy and an F1-score of 99.20% on an independent dataset that has no ground truth binary masks. Therefore, the proposed method has the potential to aid front line health care providers and can be deployed in areas with limited access to health care facilities.

Even though proposed segmentation model achieved a better performance than all other baseline models, the amount of time it took to train the model was long (about 72hrs). However, inference time for all models is approximately the same, approximately 2 seconds for one image. With higher computing power proposed method may produce faster results.

To prevent overfitting, several measures were used like batch normalization, dropout, stratified cross validation. In the future, a larger dataset can be acquired from hospitals to further validate results. Hence, the model’s extent to which the model generalizes in different locations can be evaluated. Additionally, a CapsNet model with many more layers may produce better results than the one used in the study.

Furthermore, applying better post-processing methods that can remove noisy small patches from generated masks may improve our results. Finally, the basic unit of the all models used in this study is the neural network which is a black-box model and its explainability is still a work in progress. Hence, if the model structures can be explained more clearly, different studies can be done in the future.

ACKNOWLEDGMENT

This study was carried out in the scope of the Master Thesis of Ivoline C. Ngong.

Table 11. A quantitative comparisons between our model and state of the art for the classification tasks

Study	Method	Modality	Number (#) of cases	# of images	Accuracy	Precision	Recall	Specificity	F1-score
[38]	Alexnet	X-ray	60 COVID-19, 70 Normal, 70Pneumonia_bac, 70 Pneumonia viral	270	66.67	--	66.67	--	--
[38]	Resnet18	X-ray	60 COVID-19, 70 Normal, 70 Pneumoni_bac, 70 Pneumonia viral	270	69.47	--	66.67	--	-
[38]	Googlenet	X-ray	60 COVID-19, 70 Normal, 70 Pneumoni_bac, 70 Pneumonia viral	270	80.56	--	80.56	--	--
[42]	CNN+SVM	X-ray	127 COVID-19, 127 Pneumonia, 127 Healthy	381	95.33	--	95.33	--	95.34
[40]	CNN	CT	349 COVID-19, 483 non-COVID-19	812	89	--	-	--	90
[43]	DeepPneumonia	CT	88 COVID-19, 101 Pneumonia, 86 Healthy	275	86	79	96	--	87
[44]	DarkCovidNet	X-ray	125 COVID-19, 500 Normal, 500 Pneumonia	1125	87.02	89.96	85.35	92.18	87.37
[44]	DarkCovidNet	X-ray	125 COVID-19, 500 Normal	625	98.08	98.03	95.13	95.3	96.51
[39]	VGG-19	X-ray	224 COVID-19, 700 Pneumonia, 504 Healthy	1428	96.78	--	98.66	96.46	--
[8]	Covid-Net	X-ray	53 COVID-19, 8066 Healthy, 5526 non-COVID-19	13975	92.4	--	--	--	--
[48]	Covid-MobileXpert	X-ray	179 COVID-19, 37 Healthy, 179 Pneumonia	395	88.9	--	--	--	--
[41]	Covidx-net	X-ray	25 COVID-19, 25 Normal	50	90	91.5	90	-	91
[32]	UNet + 3D DNN	CT	313 COVID-19, 229 non-COVID-19	542	90.8	--	90.7	91.11	--
[53]	Encoder-Decoder	CT	449 COVID-19, 98 Lung, 425 Normal, 2738 non-COVID-19	3710	94.67	--	96	92	--
[26]	UNet++ + CNN	CT	20886 COVID-19, 14469 NonCovid19	35355	92.59	--	100	81.82	--
[37]	UNet + DeCoVNet	CT	70 COVID-19, 1008 Pneumonia, 1431 non-COVID-19	2509	90.8	--	96.0	70.7	--
[47]	ResNet20	X-ray	68 COVID-19, 2786Pneumonia_b, 1583 Normal, 1504 Pneumonia_v	5941	92.8	--	--	--	--
[45]	Inception	CT	325 COVID-19, 740 Pneumonia	1065	79.3	--	83	67.0	
[46]	ResNet50	X-ray	50 COVID-19, 50 Normal	100	96.1	76.5	91.8	96.6	83.5
[51]	ResNet + Location Attention	CT	219 COVID-19, 175 Healthy, 224 Viral Pneumonia	618	86.7	86.9	86.7	--	86.7
[52]	CovNet	CT	468 COVID-19, 1551 Pneumonia, 1173 non-Pneumonia	3192	95	--	87	92	--
[35]	JCS	CT	64711 COVID-19, 68041 non-COVID-19	132752		--	94.5	93.5	--
[54]	CNN	CT	275 COVID-19, 275 Normal	550	98.7	--	100	98.3	--
[49]	Transferred DenseNet2021	CT	1252 COVID-19, 1229 non-COVID-19	2481	98.18	97.76	98.20	98.17	97.98
[50]	CycleGAN + Inception	X-ray	4044 COVID-19, 5500 non-COVID-19	9544	94.2	--	--	--	--
Proposed	cGAN-UNet + CNN	CT	1252 COVID-19, 1229 non-COVID-19	2481	99.20	98.43	100	98.37	99.20

REFERENCES

- [1] Ceraolo, C., Giorgi, F.M. (2020). Genomic variance of the 2019-nCoV coronavirus. *Journal of Medical Virology*, 92(5): 522-528. <https://doi.org/10.1002/jmv.25700>
- [2] WHO (World Health Organization). (2020). Considerations for quarantine of individuals in the context of containment for coronavirus disease (COVID-19): interm guidance. Technical Report. <https://apps.who.int/iris/handle/10665/331497>, accessed on Jan. 20, 2021.
- [3] Cucinotta, D., Vanelli, M. (2020). WHO declares Covid-19 a pandemic. *Acta Biomedica*, 91(1): 157-160. <https://doi.org/10.23750/abm.v91i1.9397>
- [4] WHO (World Health Organization). (2020). Weekly Epidemiological Update -15 Dec 2020 (Covid-19). Technical Report.

- <https://www.who.int/publications/m/item/weekly-epidemiological-update---15-december-2020>.
- [5] Jawerth, N. (2020). How is the COVID-19 virus detected using real time RT-PCR? IAEA Bulletin, 61(2): 8-11.
- [6] Herper, M., Branswell, H. (2020). Shortage of crucial chemicals creates new obstacle to US coronavirus testing. STAT News. <https://www.statnews.com/2020/03/10/shortage-crucial-chemicals-us-coronavirus-testing/>.
- [7] Won, J., Lee, S., Park, M., Kim, T.Y., Park, M.G., Choi, B.Y., Kim, D., Chang, H., Heo, W.D., Kim, V.N., Lee, C.J. (2020). Development of a laboratory-safe and low-cost detection protocol for SARS-CoV-2 of the Coronavirus Disease 2019 (COVID-19). *Experimental Neurobiology*, 29(2): 107-119. <https://doi.org/10.5607/en.20009>
- [8] Wang, L.D., Lin, Z.Q., Wong, A. (2020). COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific Reports*, 10(1): 19549. <https://doi.org/10.1038/s41598-020-76550-z>
- [9] Lin, C., Ding, Y.X., Xie, B., Sun, Z.J., Li, X.G., Chen, Z.X., Niu, M. (2020). Asymptomatic novel coronavirus pneumonia patient outside Wuhan: The value of CT images in the course of the disease. *Clinical Imaging*, 63: 7-9. <https://doi.org/10.1016/j.clinimag.2020.02.008>
- [10] Wang, K., Kang, S., Tian, R., Zhang, X., Zhang, X., Wang, Y. (2020). Imaging manifestations and diagnostic value of chest CT of coronavirus disease 2019 (COVID-19) in the Xiaogan area. *Clinical Radiology*, 75(5): 341-347. <https://doi.org/10.1016/j.crad.2020.03.004>
- [11] Meng, H., Xiong, R., He, R.Y., Lin, W.C., Hao, B., Zhang, L., Lu, Z.L., Shen, X.K., Fan, T., Jiang, W.Y., Yang, W.B., Li, T., Chen, J., Geng, Q. (2020). CT imaging and clinical course of asymptomatic cases with COVID-19 pneumonia at admission in Wuhan, China. *The Journal of Infection*, 81(1): e33-e39. <https://doi.org/10.1016/j.jinf.2020.04.004>
- [12] Ai, T., Yang, Z.L., Hou, H.Y., Zhan, C.N., Chen, C., Lv, W.Z., Tao, Q., Sun, Z.Y., Xia, L.M. (2020). Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. *Radiology*, 296(2): E32-E40. <https://doi.org/10.1148/radiol.2020200642>
- [13] Fang, Y.C., Zhang, H.Q., Xie, J.C., Lin, M.J., Ying, L.J., Pang, P.P., Ji, W.B. (2020). Sensitivity of chest CT for COVID-19: Comparison to RT-PCR. *Radiology*, 296: E115-E117. <https://doi.org/10.1148/radiol.2020200432>
- [14] Mossa-Basha, M., Meltzer, C.C., Kim, D.C., Tuite, M.J., Kolli, K.P., Tan, B.S. (2020). Radiology department preparedness for COVID-19: Radiology scientific expert review panel. *Radiology*, 296(2): E106-E112. <https://doi.org/10.1148/radiol.2020200988>
- [15] Survivor, P. (2019). Shortage of radiologists. <https://www.pseudomyxomasurvivor.org/shortage-of-radiologists/>.
- [16] Stephen, O., Sain, M., Maduh, U.J., Jeong, D. (2019). An efficient deep learning approach to pneumonia classification in healthcare. *Journal of Healthcare Engineering*, 2019: 4180949. <https://doi.org/10.1155/2019/4180949>
- [17] Biswas, M., Kupppili, V., Saba, L., Edla, D.R., Suri, H.S., Cuadrado-Godia, E., Laird, J.R., Marinhoe, R.T., Sanches, J.M., Nicolaidis, A., Suri, J.S. (2019). State-of-the-art review on deep learning in medical imaging. *Frontiers in Bioscience*, 24: 392-426. <https://doi.org/10.2741/4725>
- [18] Arsalan, M., Kim, D.S., Owais, M., Park, K.R. (2020). OR-Skip-Net: Outer residual skip network for skin segmentation in non-ideal situations. *Expert Systems with Applications*, 141: 112922. <https://doi.org/10.1016/j.eswa.2019.112922>
- [19] Park, B., Park, H., Lee, S.M., Seo, J.B., Kim, N. (2019). Lung segmentation on HRCT and volumetric CT for diffuse interstitial lung disease using deep convolutional neural networks. *Journal of Digital Imaging*, 32(6): 1019-1026. <https://doi.org/10.1007/s10278-019-00254-8>
- [20] Mittal, M., Goyal, L.M., Kaur, S., Kaur, I., Verma, A., Hemant, D.J. (2019). Deep learning based enhanced tumor segmentation approach for MR brain images. *Applied Soft Computing*, 78: 346-354. <https://doi.org/10.1016/j.asoc.2019.02.036>
- [21] Müller, D., Rey, I.S., Kramer, F. (2020). Automated chest CT image segmentation of COVID-19 lung infection based on 3D U-Net. *ArXiv Preprint arXiv: 2007.04774*. <https://doi.org/10.48550/arXiv.2007.04774>
- [22] Liu, X., Faes, L., Kale, A.U., et al. (2019). A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The Lancet Digital Health*, 1(6): E271-E297. [https://doi.org/10.1016/S2589-7500\(19\)30123-2](https://doi.org/10.1016/S2589-7500(19)30123-2)
- [23] Zheng, C.S., Deng, X.B., Fu, Q., Zhou, Q., Feng, J.P., Ma, H.Y., Liu, W.Y., Wang, X.G. (2020). Deep learning-based detection for COVID-19 from chest CT using weak label. *MedRxiv*. <https://doi.org/10.1101/2020.03.12.20027185>
- [24] Shan, F., Gao, Y.Z., Wang, J., Shi, W.Y., Shi, N.N., Han, M.F., Xue, Z., Shen, D.G., Shi, Y.X. (2020). Lung infection quantification of covid-19 in CT images with deep learning. *ArXiv Preprint: arXiv2003.04655*. <https://doi.org/10.48550/arXiv.2003.04655>
- [25] Fan, D.P., Zhou, T., Ji, G.P., Zhou, Y., Chen, G., Fu, H.Z., Shen, J.B., Shao, L. (2020). Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Images. *IEEE Trans. Medical Imaging*, 39(8): 2626-2637. <https://doi.org/10.1109/TMI.2020.2996645>
- [26] Xing, Y.F., Zhong, L., Zhong, X. (2020). An encoder-decoder network based FCN architecture for semantic segmentation. *Wireless Communications and Mobile Computing*, 2020: 8861886. <https://doi.org/10.1155/2020/8861886>
- [27] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015)*, pp. 234-241. https://doi.org/10.1007/978-3-319-24574-4_28
- [28] Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O. (2016). 3D U-Net: Learning dense volumetric segmentation from sparse annotation. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2016)*, pp. 424-432. https://doi.org/10.1007/978-3-319-46723-8_49
- [29] Zhou, T.X., Canu, S., Ruan, S. (2020). An automatic COVID-19 CT segmentation based on U-Net with attention mechanism. *International Journal of Imaging Systems and Technology*, 31(1): 16-27.

- <https://doi.org/10.1002/ima.22527>
- [30] Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J. (2020). UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, 39(6): 1856-1867. <https://doi.org/10.1109/TMI.2019.2959609>
- [31] Chen, X., Yao, L., Zhang, Y. (2020). Residual attention u-net for automated multi-class segmentation of covid-19 chest CT images. *ArXiv Preprint: arXiv2004.05645*. <https://doi.org/10.48550/arXiv.2004.05645>
- [32] Wu, Y.H., Gao, S.H., Mie, J., Xu, J., Fan, D.P., Zhang, R.G., Cheng, M.M. (2021). JCS: An explainable covid-19 diagnosis system by joint classification and segmentation. *IEEE Trans. Image Processing*, 30: 3113-3126. <https://doi.org/10.1109/TIP.2021.3058783>.
- [33] Chen, J., Wu, L.L., Zhang, J., et al. (2020). Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography: A prospective study. *Scientific Reports*, 10: 19196. <https://doi.org/10.1101/2020.02.25.20021568>
- [34] Milletari, F., Navab, N., Ahmadi, S. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *4th International Conference on 3D Vision (3DV)*, pp. 565-571. <https://doi.org/10.1109/3DV.2016.79>
- [35] Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., Glocker, B., Rueckert, D. (2018). Attention U-Net: Learning where to look for the pancreas. *ArXiv Preprint: arXiv1804.03999*. <https://doi.org/10.48550/arXiv.1804.03999>
- [36] Zhang, P.Y., Zhong, Y.X., Deng, Y.L., Tang, X.Y., Li, X.Q. (2020). CoSinGAN: Learning COVID-19 infection segmentation from a single radiological image. *Diagnostics*, 10(11): 901. <https://doi.org/10.3390/diagnostics10110901>
- [37] Zhang, J.P., Xie, Y.T., Li, Y., Shen, C.H., Xia, Y. (2020). Covid-19 screening on chest x-ray images using deep learning based anomaly detection. *ArXiv Preprint: arXiv2003.12338*.
- [38] Loey, M., Smarandache, F., Khalifa, N.E.M. (2020). Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning. *Symmetry*, 12(4): 651. <https://doi.org/10.3390/sym12040651>
- [39] Apostolopoulos, I.D., Mpesiana, T.A. (2020). COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2): 635-640. <https://doi.org/10.1007/s13246-020-00865-4>
- [40] Yang, X.Y., He, X.H., Zhao, J.Y., Zhang, Y.C., Zhang, S.H., Xie, P.T. (2020). Covid-Ct-Dataset: A CT scan dataset about COVID-19. *ArXiv Preprint: arXiv2003.13865*. <https://doi.org/10.48550/arXiv.2003.13865>
- [41] Hemdan, E.E., Shouman, M.A., Karar, M.E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose COVID-19 in X-ray images. *ArXiv Preprint: arXiv2003.11055*. <https://doi.org/10.48550/arXiv.2003.11055>
- [42] Sethy, P.K., Behera, S.K., Ratha, P.K., Biswas, P. (2020). Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. *International Journal of Mathematical, Engineering and Management Sciences*, 5(4): 643-651. <https://doi.org/10.33889/IJMEMS.2020.5.4.052>
- [43] Song, Y., Zheng, S.J., Li, L., Zhang, X., Zhang, X.D., Huang, Z.W., Chen, J.W., Wang, R.X., Zhao, H.Y., Chong, Y.T., Shen, J., Zha, Y.F., Yang, Y.D. (2021). Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(6): 2775-2780. <https://doi.org/10.1109/TCBB.2021.3065361>
- [44] Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O., Acharya, U.R. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in Biology and Medicine*, 121: 103792. <https://doi.org/10.1016/j.compbimed.2020.103792>
- [45] Wang, S.H., Kang, B., Ma, J.L., Zeng, X.J., Xiao, M.M., Guo, J.D., Cai, M.F., Yang, J.Y., Li, Y.D., Meng, X.F., Xu, B. (2020). A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *MedRxiv*. <https://doi.org/10.1101/2020.02.14.20023028>
- [46] Narin, A., Kaya, C., Pamuk, Z. (2020). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, 24(3): 1207-1220. <https://doi.org/10.1007/s10044-021-00984-y>
- [47] Ghoshal, B., Tucker, A. (2020). Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection. *ArXiv Preprint: arXiv2003.10769*. <https://doi.org/10.48550/arXiv.2003.10769>
- [48] Li, X., Li, C., Zhu, D. (2020). Covid-MobileXpert: On-Device Covid-19 patient triage and follow-up using chest X-rays. *ArXiv Preprint: arXiv2004.03042*. <https://doi.org/10.48550/arXiv.2004.03042>
- [49] Lahsaini, I., El Habib Daho, M., Chikh, M.A. (2021). Deep transfer learning based classification model for Covid-19 using chest CT-scans. *Pattern Recognition Letters*, 152: 122-128. <https://doi.org/10.1016/j.patrec.2021.08.035>
- [50] Bargshady, G., Zhou, X.J., Barua, P.D., Gururajan, R., Li, Y.F., Acharya, U.R. (2021). Application of CycleGAN and transfer learning techniques for automated detection of Covid-19 using X-Ray images. *Pattern Recognition Letters*, 153: 67-74. <https://doi.org/10.1016/j.patrec.2021.11.020>
- [51] Xu, X.W., Jiang, X.G., Ma, C.L., Du, P., Li, X.K., Lv, S.Z., Yu, L., Ni, Q., Chen, Y.F., Su, J.W., Lang, G.J., Li, Y.T., Zhao, H., Liu, J., Xu, K.J., Ruan, L.X., Sheng, J.F., Qiu, Y.Q., Wu, W., Liang, T.B., Li, L.J. (2020). A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering*, 6(10): 1122-1129. <https://doi.org/10.1016/j.eng.2020.04.010>
- [52] Li, L., Qin, L.X., Xu, Z.G., Yin, Y.B., Wang, X., Kong, B., Bai, J.J., Lu, Y., Fang, Z.H., Song, Q., Cao, K.L., Liu, D.L., Wang, G.S., Xu, Q.Z., Fang, X.S., Zhang, S.Q., Xia, J., Xia, J. (2020). Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. *Radiology*, 296(2): E65-E71. <https://doi.org/10.1148/radiol.2020200905>
- [53] Amyar, A., Modzelewski, R., Li, H., Ruan, S. (2020). Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation. *Computers in Biology and Medicine*, 126: 104037.

- <https://doi.org/10.1016/j.compbiomed.2020.104037>
- [54] Karakanis, S., Leontidis, G. (2021). Lightweight deep learning models for detecting COVID-19 from chest X-ray images. *Computers in Biology and Medicine*, 130: 104181. <https://doi.org/10.1016/j.compbiomed.2020.104181>
- [55] Paiva, O. (2020). Helping radiologists to help people in more than 100 countries! Coronavirus cases. COVID-19 CT Images Segmentation Dataset. <https://www.kaggle.com/datasets/andrewmvd/covid19-ct-scans>, accessed on Jan. 20, 2021.
- [56] Soares, E., Angelov, P., Biaso, S., Froes, M.H., Abe, D.K. (2020). SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. *MedRxiv*. <https://doi.org/10.1101/2020.04.24.20078584>
- [57] Dawood, F.A.A., Rahmat, R.W., Dimon, M.Z., Nurliyana, L., Kadiman, S.B. (2011). Automatic boundary detection of wall motion in two-dimensional echocardiography images. *Journal of Computer Science*, 7(8): 1261-1266.
- [58] Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. *Graphic Gems IV*. Academic Press Inc.
- [59] Wikipedia. (2020). Adaptive histogram equalization. https://en.wikipedia.org/w/index.php?title=Adaptive_histogram_equalization&oldid=936814673, accessed on Jan. 20, 2021.
- [60] Saglam, A., Baykan, N.A. (2017). Effects of color spaces and distance norms on graph-based image segmentation. *3rd International Conference on Frontiers of Signal Processing (ICFSP 2017)*, pp. 130-135. <https://doi.org/10.1109/ICFSP.2017.8097156>
- [61] Isola, P., Zhu, J.Y., Zhou, T.H., Efros, A.A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pp. 5967-5976. <https://doi.org/10.1109/CVPR.2017.632>
- [62] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11): 139-144. <https://doi.org/10.1145/3422622>
- [63] Denton, E., Chintala, S., Szlam, A., Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *28th International Conference on Neural Information Processing Systems (NIPS 2015)*, pp. 486-1494.
- [64] Hu, Z.F., Fu, Y.Q., Luo, Y., Xu, X., Xia, Z.G., Zhang, H.W. (2020). Speaker recognition based on short utterance compensation method of generative adversarial networks. *International Journal of Speech Technology*, 23: 443-450. <https://doi.org/10.1007/s10772-020-09711-0>
- [65] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative adversarial nets. In *27th International Conference on Neural Information Processing Systems (NIPS 2014)*, pp. 2672-2680.
- [66] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training GANs. *ArXiv Preprint: arXiv1606.03498*
- [67] Mirza, M., Osindero, S. (2014). Conditional generative adversarial nets. *ArXiv Preprint: arXiv1411.1784*. <https://doi.org/10.48550/arXiv.1411.1784>
- [68] Mino, A., Spanakis, G. (2018). LoGAN: Generating logos with a generative adversarial neural network conditioned on color. In *17th IEEE International Conference on Machine Learning and Applications (ICMLA 2018)*, pp. 965-970. <https://doi.org/10.1109/ICMLA.2018.00157>
- [69] Brownlee, J. (2019). How to implement GAN hacks in Keras to train stable models. *Machine Learning Mastery*.
- [70] Kodym, O., Španěl, M., Herout, A. (2018). Segmentation of head and neck organs at risk using cnn with batch dice loss. *ArXiv Preprint: arXiv1812.02427*. <https://doi.org/10.48550/arXiv.1812.02427>
- [71] Sabour, S., Frosst, N., Hinton, G.E. (2017). Dynamic routing between capsules. *ArXiv Preprint: arXiv1710.09829*. <https://doi.org/10.48550/arXiv.1710.09829>
- [72] Python Programming Language Web Page. <https://www.python.org/about/>, accessed on Jan. 20, 2021.
- [73] Abadi, M., Barham, P., Chen, J.M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX conference on Operating Systems Design and Implementation (OSDI 2016)*, pp. 265-283.
- [74] Yeghiazaryan, V., Voiculescu, I. (2015). *An Overview of Current Evaluation Methods Used in Medical Image Segmentation*. Oxford.
- [75] Ma, J., Wang, Y.X., An, X.L., Ge, C., Yu, Z.Q., Chen, J.N., Zhu, Q.J., Dong, G.Q., He, J., He, Z.Q., Cao, T.J., Zhu, Y.T., Nie, Z.Q., Yang, X.P. (2020). Towards Data-efficient learning: A benchmark for COVID-19 CT lung and infection segmentation. *Medical Physics*, 48(3): 1197-1210. <https://doi.org/10.1002/mp.14676>
- [76] Badrinarayanan, V., Kendall, A., Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12): 2481-2495. <https://doi.org/10.1109/TPAMI.2016.2644615>
- [77] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J. (2017). Pyramid scene parsing network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pp. 6230-6239. <https://doi.org/10.1109/CVPR.2017.660>
- [78] Ngong, I.C. (2021). Uretken Cekismeli Ag ve Unet Kullanilarak Segmente Edilmis Tomografi Goruntulerinden COVID-19 Siniflandirmasinda Farkli Derin Ogrenme Mimarilerinin Kullanimi, Master Thesis, Konya Technical University, Turkey. https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=v7BkNnneptnbhn8rNR77LcapT_8U8xBFIgvtENgqFj-AOGLcwCK8_PROnMeTjVh

NOMENCLATURE

G	Generator in GAN
$G(z)$	Fake data for GAN
x	Real data
z	Noise data for GAN
D	Discriminator in GAN
L	Objective function of GAN
E	Expected value of GAN
$P_{data}(x)$	Distribution of training examples
L_2	Regularization
$p(y)$	Predicted mask of CNN

L_{DIS} Discriminator loss
 L_{GEN} Generator loss