IJCI V10-01 (2023) 01–17
International Journal of Computers and Information (IJCI)
Available online at *https://ijci.journals.ekb.eg/*

A Comparative Analysis of COVID-19 Diagnosis Using Lung Ultrasound Based on Convolutional Neural Networks

Ola G. Elkhouly[a], Mohamed G. Malhat[a], Arabi E. Keshk[a], Maha M. Elsabaawy[b]

[a]*Department of Computer Science, Faculty of Computers and Information, Menofia University, Menofia, Egypt*
[b]*Department of Hepatology, National Liver Institute, Menofia, Egypt*
ola.galal803@ci.menofia.edu.eg, m.gmalhat@yahoo.com, arabi.keshk@ci.menofia.edu.eg, maha.ahmed1@liver.menofia.edu.eg

Abstract

The COVID-19 pandemic resulted in millions of infections which led to increased demands on health systems around the world. Due to the shortage of diagnostic tools and the stress on radiologists, the need to utilize computer-assisted methods to diagnose COVID-19 has increased. There have been many attempts to use deep learning to accelerate the process of COVID-19 diagnosis. However, there is still an opportunity for further improvements in the results. In this paper, we present a comparative study for COVID-19 diagnosis using multiple convolutional neural networks, as they are the most widely used architectures in classification problems. We trained the convolutional neural networks (CNNs) using 5-fold cross-validation. We used lung ultrasound images proposed in the Point of Care Ultrasound (POCUS) dataset. InceptionV1 achieved the highest results with accuracy and balanced accuracy of 84.3% and 81.8%, respectively. Qualitatively, employed architectures show a variation in performance depending on the internal layers of each architecture. A deep learning architecture can distinguish similar-looking lung ultrasound pathology, including COVID-19, that may be difficult to distinguish by pathologists and radiologists.

Keywords: Convolutional Neural Networks; COVID-19; Lung Ultrasound; Medical Imaging

1. Introduction

Corona Virus Disease of 2019 (COVID-19) is a disease caused by a virus known as severe acute respiratory syndrome (SARS-CoV-2) [1]. As of May 2022, there have been over 522 million confirmed cases and over six million deaths [2]. COVID-19 was declared by WHO as a pandemic on March 11, 2020 [3]. This pandemic has encouraged scientists to start searching for ways to detect it. The key to successfully limiting the spread is early detection, isolation, and patient care [4].

The available diagnostic methods are based on the detection of viral antigens, human antibodies, or viral genes. The most accurate method for identifying viral genes is Reverse Transcription-Polymerase Chain Reaction (RT-PCR) [5]. Although WHO has emphasized the significance of testing to fight COVID-19, most countries don't have enough labs and resources to do so. Swab procedures and the quality of the lab itself seem to have a significant impact on the results of the RT-PCR [6].

Convolutional neural networks (CNNs) and deep learning neural networks have succeeded in a variety of medical image classification tasks [42–45]. Complex feature extraction is easier when using CNN architectures. However, large quantities of data are required in this process. This study analyzes the performance of various CNN architectures in classifying ultrasound images into three classes: COVID-19, bacterial pneumonia, and healthy.

For imaging data, Chest X-ray (CXR) and chest Computed Tomography (CT) are the often-used techniques in the COVID-19 diagnosis procedure. The diagnosis using chest CT scan may be quicker than using RT-PCR, which can take up to two days [7] and requires many tests for conclusive findings. A CT scan takes about 10 to

30 minutes [8]. However, chest CT has many disadvantages, such as: it exposes patients to radiation and it is also expensive. Pregnant women risk exposing their unborn child to radiation during a CT scan [9], and ionizing radiation is harmful to children [10]. Furthermore, some people are allergic to iodine contrast dyes, which are often used for CT scans. It also requires sterilization and poses a serious danger of infection among healthcare workers [11]. X-rays are the most often used first-line diagnostic imaging method. It shows low sensitivity and specificity for COVID-19 (e.g., 89% of chest X-rays in 493 COVID-19 cases were found to be normal) [12, 13].

Ultrasound is being used more often in point-of-care settings in the medical field to diagnose acute respiratory diseases [14–16]. It is used to diagnose diseases such as pleural effusion, alveolar consolidation, interstitial syndrome, and pneumothorax using pathological patterns including B-lines, A-lines, and barcode signs [17]. It is a more broadly accessible, cost-effective, safe, and real-time imaging method, which is our fundamental motivation to explore this method of diagnosis.

A significant disadvantage of traditional diagnosis of COVID-19 using LUS is that observing COVID-19 specific patterns is not easy and requires experienced physiologists. Deep learning (DL) has proved effective in medical imaging [25], and some research is now looking towards DL-based methods to help with lung disease detection [18]. In COVID-19 patients, lung ultrasound (LUS) has a greater diagnostic sensitivity than CXR [19]. Furthermore, radiologists found parallels between LUS and CT [20,21]. Studies have shown that LUS for COVID-19 has diagnostic accuracy equivalent to CT [22,23] and is much more sensitive in identifying lung imaging biomarkers [24]. COVID-19 patients have a significant visual appearance in subpleural lesions other than tuberculosis, cardiogenic pulmonary edema, and bacterial pneumonia. In particular, the B-lines, consolidations, uneven pleural lines, pleural effusion, and lung sliding that were shown in CT scans disappeared in LUS along with the thickened pleura, consolidation shadow, and ground-glass opacity. Recently, diagnosis methods used in medical ultrasound analysis have gained increasing attention, which is based on the revelations in deep learning and computer vision. LUS scans can be used as a screening, detection, and follow-up technique for any lung disease, hypothetically.

In this paper, a comparison of several deep learning techniques has been done to evaluate their performance of using LUS for COVID-19 detection. The used architectures could be used as a tool to create assistance tools for pathologists and radiologists to accelerate the process of COVID-19 diagnosis.

The organization of the paper is as follows. Section 2 provides a brief review of recent papers relevant to this paper, followed by related challenges. Section 3 presents a comparison evaluation of many common CNN architectures. The dataset creation procedure and the preparation pipeline are presented in Section 4.1. Finally, Section 4.2 and 4.3 presents the performance results with a discussion of our architectures on the lung ultrasound dataset.

2. Related Work

The literature on exploiting medical imaging and deep learning to diagnose and classify many diseases has recently exploded. More publications are appearing on CT and CXR imaging of COVID-19 compared to ultrasound. Before COVID-19, many published papers on using ultrasound with deep learning and specifically lung ultrasound focus on B-line detection as the most common task [26-28]. Others focus on pulmonary lesions [29-33], the extraction of pleural lines [34], or lung cancer diagnosis [35].

COVID-19 diagnosis using lung ultrasound with deep learning architectures has been studied in four papers so far (Roy et al., 2020 [36]; Born et al., 2021 [37,38]; Diaz-Escobar et al., 2021 [39]). Roy et al. [36] utilized the disease score scheme given by Soldati et al. [40] in the detection of COVID-19 disease severity. They also utilized a large dataset, but it is not open source. Furthermore, they didn't perform any image processing to enhance the images.

Born et al. [38] developed a deep learning architecture that derives a disease intensity score from LUS scans to help radiologists in the process of diagnosing COVID-19-related lung diseases. The work and its extension by Born et al. [37,38] attempt to classify healthy patients, bacterial pneumonia, and COVID-19

from LUS frames. The POCUS dataset of LUS imaging is introduced and used in this paper. Born et al. [37,38] collected the POCUS dataset from several sources and posted it to their GitHub repository for public use [41]. The architecture presented by Born et al. [37] is based on a modified convolutional VGG16 architecture known as POCOVID-net, with 89% accuracy and 82% balanced accuracy. In their extended work [38], POCUS dataset was expanded, and four additional architectures were assessed: two VGG-segment architectures using a modified version of the segmentation ensemble introduced in the original work, a NasNetMobile architecture, and a VGG network coupled with class activation maps (calculate class-specific heatmaps) [37]. The POCOVID-net architecture obtained an accuracy and balanced accuracy of 87% in the extended work [38] and had the best 5-fold cross-validation results when compared to other architectures. Moreover, they provided a preprocessing pipeline to improve the image quality.

While previous paper had focused on the classification of LUS images using a VGG16 convolutional architecture, Diaz-Escobar et al. [39] went a step further. The convolutional layers of the VGG16 architecture, which is pre-trained on ImageNet, create the POCOVID-net architecture. They swapped out the POCOVID-net [37,38] VGG16 base architecture with one of the following: Inception-V3, Xception, VGG19, and ResNet50 CNN architectures, freezing the last fully connected layers. They conclude that the InceptionV3-based architecture accomplished the highest accuracy and balanced accuracy of 89.1% and 89.3%, respectively. According to the results presented in [36–39], there is still an opportunity for improvement in COVID-19 detection using LUS scans based on artificial intelligence algorithms.

3. **Methodology**

Particularly, the following CNN architectures were utilized: (sorted chronologically): LeNet-5 [46], AlexNet [47], InceptionV1 [48], InceptionV3 [49], ResNet50 [50], InceptionV4 [51], Inception-ResNet-V2 [51], DenseNet121 [52], ResNext50 [53], Xception [54] and POCOVID-Net [38].

In this paper, we used the original implementation of CNN architectures without any pre-initialized weights except for POCOVID-Net. Per fold, each architecture was trained with early stopping, 100 epochs, and a batch size of 32. For the training, we used the Adam optimization [55] architecture with a learning rate of 0.0001 and the categorical cross-entropy loss function and a dropout [56] of 0.5. Depending on the architecture, different numbers of trainable layers were used. Table 1 shows CNNs total weights. The number of hidden network layers, learning rate, dropout, epoch number, batch size, optimizer and other variables should be changed in order to further improve the network performance. This will happen in a subsequent contribution.

3.1. LeNet-5

One of the most basic architectures is LeNet-5 [46]. It includes two convolutional layers and three fully-connected layers. The LeNet-5 network architecture is shown in Fig.1. The average-pooling layer was known as a subsampling layer, and it featured trainable weights, which is not common in the design of CNNs currently. This design has evolved into the typical blueprint for building CNNs by stacking convolutions with activation functions, pooling layers, and finalizing the network with one or more fully-connected layers. The original aim of LeNet-5 was to be able to perform optical character recognition. The original image set that was designed for LeNet-5 consisted of images of numbers from 0–9 in black and white. Compared to other architectures, the architecture is highly compact and straightforward. Although this architecture is basic and simple, we had to use it in comparison because it is fundamental and all the upcoming architectures are based on it.

All images are resized to 32 x 32 pixels, converted to grayscale color mode, and fed through the convolutional layers of the architecture. The results are generated with a total of 82,231 trainable parameters. We modified the last fully-connected layer to classify the input images into three classes.

3.2. AlexNet

AlexNet [47] is a classic convolutional neural network architecture that was introduced in 2012 at the ImageNet Large Scale Visual Recognition Challenge. It consists of eight layers: five convolutional and three

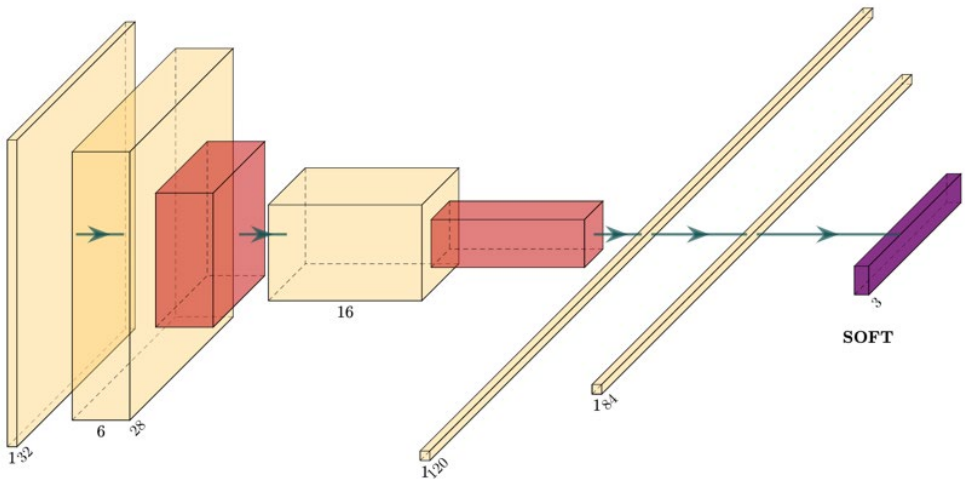

Fig. 1. Architecture of LeNet-5 [46]. Created by PlotNeuralNet [57].

Table 1. CNNs total weights.

Architecture	Total Weights
LeNet-5	82,231
AlexNet	58,299,139
InceptionV1	5,610,259
InceptionV3	21,808,931
ResNet50	23,593,859
InceptionV4	41,179,011
Inception-ResNet-V2	54,341,347
DenseNet121	7,040,579
ResNeXt50	26,515,203
Xception	20,867,627
POCOVID-Net	14,747,971

fully-connected layers. Rectified Linear Units (ReLUs) were initially implemented by them as activation functions. ReLU got rid of the vanishing gradient problem. Furthermore, it does not limit the output, unlike other activation functions, so there is not that much loss of features. They also introduced the use of dropout [56] techniques to avoid overfitting. It was the first major CNN architecture that used GPUs for training, which led to faster training of architectures. However, the depth of this architecture is very low. Hence, it struggles to learn features from image sets. The AlexNet network architecture is shown in Fig.2.

All images are reduced in size to 227 x 227 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 58,296,387 trainable parameters and 2,752 non-trainable parameters.

3.3. *InceptionV1*

When prior architectures were just going deeper to enhance performance and accuracy while sacrificing computational expense, Inception Nets reached a milestone as a CNN classifier. It is worth mentioning that the primary distinguishing feature of the design is the rapid adoption of computer resources inside the network.

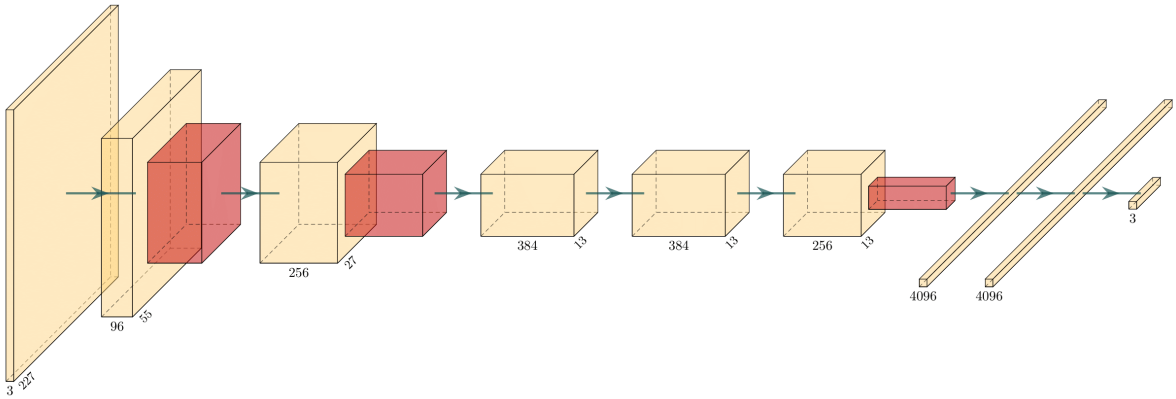

Fig. 2. Architecture of AlexNet [47]. Created by PlotNeuralNet [57].

InceptionV1 [48] has 22 layers and over 5 million parameters. It depends on filtering the same region with different kernels and then concatenating all features. Research on approximating sparse structures resulted in the introduction of an Inception module's architecture. They also added two auxiliary classifiers to enhance discrimination in the classifier's lower stages, improve the gradient signal that is transmitted back, and provide additional regularization. The auxiliary networks, the branches that are connected to the auxiliary classifier, are discarded at inference time.

All images are reduced in size to 224 x 224 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 5,595,699 trainable parameters and 14,560 non-trainable parameters.

3.4. *InceptionV3*

InceptionV3 [49] is a successor to InceptionV1 [48], with about 24M parameters. InceptionV3 has many tweaks to the optimizer, loss function, and adding batch normalization [58], also to the auxiliary layers in the auxiliary network. InceptionV3 is considered one of the first designs to use batch normalization.

It differs from InceptionV1 in three ways: It factorizes a n×n convolutions into asymmetric convolutions: 1×n and n×1 convolutions. It factorizes 5×5 convolution into two 3×3 convolution operations. Finally, it replaces 7×7 with a series of 3×3 convolutions. The reduction of the input dimension of the layers helps in avoiding representational bottlenecks, which is the main motivation of InceptionV3. Also, by using factorization methods, we can do more efficient computations.

All images are reduced in size to 299 × 299 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 21,774,499 trainable parameters and 34,43 non-trainable parameters.

3.5. *ResNet50*

We have seen nothing except an increase in the number of layers in the design and improved performance over the last several CNNs. However, as network depth increases, accuracy becomes saturated and rapidly declines. They addressed this problem with ResNet50 [50] by using skip connections while building deeper architectures. They were not the first to use skip connections, but they popularized using them. The basic building blocks for ResNets are the convolutional and identity blocks. ResNet is one of the early adopters of batch normalization [58] with 26M parameters. Networks with several layers may be readily taught without raising the training error percentage.

All images are reduced in size to 224 x 224 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 23,540,739 trainable parameters and 53,120 non-trainable parameters.

3.6. *InceptionV4*

InceptionV4 [51] has 43M parameters and it is considered an improvement from InceptionV3 [49]. The main differences are the stem group, shown in Fig.3, and a few tweaks to the Inception-C module, shown in Fig.4. They additionally used consistent choices for the Inception blocks across all grid sizes.

InceptionV4 introduced specialized reduction blocks that are used to modify the width and height of the grid. Reduction blocks weren't present in the earlier versions; however, the functionality was still present. Those residual connections lead to dramatic improvements in training speed. InceptionV4 works better because of the increased architecture size. The InceptionV4 network architecture is shown in Fig.5. (B)

All images are reduced in size to 299 × 299 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 41,115,843 trainable parameters and 63,168 non-trainable parameters.

Fig. 3. Stem block.

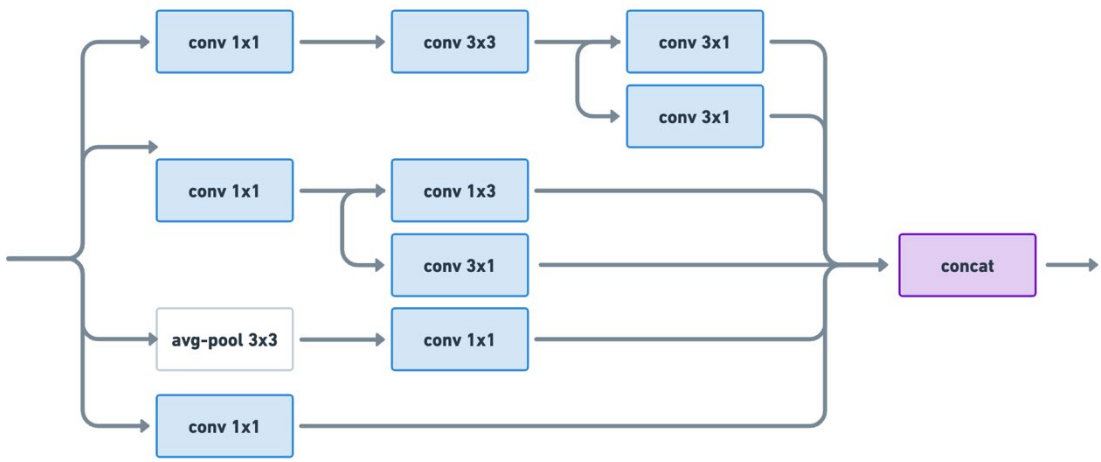

Fig. 4. Inception C block.

3.7. *Inception-ResNet-v2*

Inception-ResNet-v2 [51] is a variation of the InceptionV3 [49] architecture, and it is considerably deeper than the previous InceptionV3. In this neural network, the inception blocks have been simplified, containing fewer parallel towers than in the previous InceptionV3. A batch norm and a ReLU activation function are used after each convolutional layer. The Inception-ResNet-V2 network architecture is shown in Fig.5. (A)

All images are reduced in size to 299 × 299 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 54,280,803 trainable parameters and 60,544 non-trainable parameters.

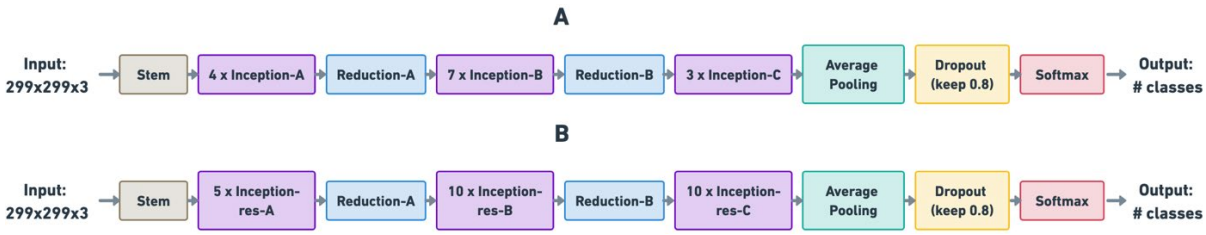

Fig. 5. **(A)** Overall architecture of the InceptionV4 network [51]. **(B)** Schema for Inception-ResNet-v2 network. The output sizes in the figure correspond to the activation vector tensor shapes of Inception-ResNet-v2 [51].

3.8. DenseNet

The issues with CNNs arise as they get deeper. This is since the path for data from the input layer to the output layer gets lengthy and it might vanish before getting to the other side. The connectivity pattern between layers proposed in other architectures is made simpler by DenseNets [52]. DenseNets utilize the capability of the network through feature reuse rather than obtaining representational power through very deep or wide architectures. Contrary to popular belief, DenseNets require fewer parameters than a comparable classic CNN since no duplicate feature maps need to be learned.

In DenseNet [52], each layer is directly connected to any or all alternative layers, and every layer has direct access to loss functions and original input signals. The feature-maps of all preceding layers are concatenated and used as inputs for any particular layer, and its feature-maps area unit is used as inputs into all subsequent layers. DenseNet has four dense blocks and transition layers between two consecutive dense blocks. Every dense block consists of many convolution layers, and every transition layer has a convolutional layer and an average pooling layer. The output layer is a fully-connected layer with a softmax activation performed with three neurons for three class classifications.

DenseNet improved data flow through the network and reduced the vanishing gradient problem. It enhanced feature reuse and reduced overfitting by using dense connections. Also, each layer provides a collective knowledge of the network. However, this neural network is very expensive in terms of space and time complexity.

All images are reduced in size to 299 × 299 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 6,956,931 trainable parameters and 83,648 non-trainable parameters.

3.9. ResNeXt50

ResNeXt50 [53] has about 25M parameters. The main difference between ResNeXts is the addition of parallel branches within each module rather than sequential layers. Their novel contribution is to scale up the number of cardinalities within a module. They were the first to introduce cardinality. It refers to the number of transformations in the set. 32 topology blocks make up the architecture, hence 32 is the cardinality value. Because of using the same topology, fewer parameters are required while more layers are added to this architecture.

All images are resized to 224 x 224 pixels, and fed through the convolutional layers of the architecture. The results are generated with a total of 26,446,979 trainable parameters and 68,224 non-trainable parameters.

3.10. Xception

Xception [54] is a modification of the Inception architecture that uses depthwise separable convolutions in place of the normal Inception modules. They were the first to introduce CNN based entirely on depthwise separable convolution layers.

Xception takes the Inception hypothesis to an extreme. The Inception hypothesis consists of firstly, cross-channel (or cross-feature map) correlations are captured by 1×1 convolutions. Consequently, spatial correlations within each channel are captured via the regular 3×3 or 5×5 convolutions. Taking this idea to an extreme means performing 1×1 to every channel, then performing a 3×3 to each output. This is identical to replacing the Inception module with depthwise separable convolutions.

All images are reduced in size to 299 × 299 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 20,813,099 trainable parameters and 54,528 non-trainable parameters.

3.11. POCOVID-Net

POCOVID-Net [38] uses the convolutional part of VGG16 [59], followed by one hidden layer of 64 neurons with ReLU activation, a dropout of 0.5 [56], and batch normalization [58], and further by the output layer with softmax activation. It is pre-trained on ImageNet. The POCOVID-Net network architecture is shown in Fig.6.

All images are reduced in size to 224 × 224 pixels and given as input to the convolutional layers of the architecture. The results are generated with a total of 14,747,843 trainable parameters.

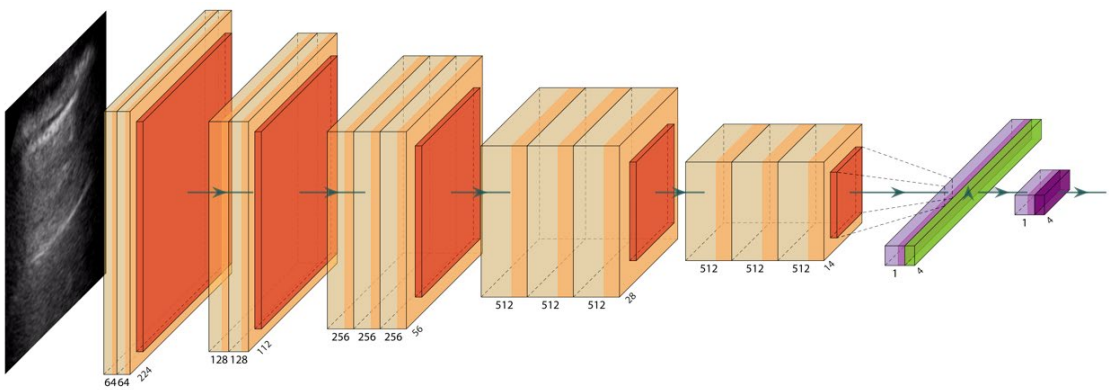

Fig. 6. POCOVID-Net Architecture [38].

4. Results and Discussion

4.1. Dataset

We utilized the POCUS dataset collected by Born et al. for all the experiments in this research [41]. The dataset combines data from collaborating hospitals as well as publicly available resources from the web (e.g., publications and educational websites), and it can be found on their GitHub repository, which is open to the public. To our knowledge, this is the most comprehensive source of COVID-19 LUS data available to the public. The POCUS dataset contains 85 images and 197 videos with a convex probe and 14 images and 64 videos with a linear probe. Fig.7 shows the distribution of the samples. The POCUS dataset gathered from 41 different sources, such as clinical information supplied by hospitals or academics teaching ultrasound courses, LUS recordings published in other scholarly journals, community platforms, public medical repositories, and health-tech companies. As follows, the videos varied in length and frame rate (160±144 frames per second, 2510Hz). Furthermore, not all videos have patient metadata since there are many different data sources.

The preprocessing of the images and videos is performed as Born et al. [38] suggested. The preprocessing phase is employed to maintain the numerical stability of the architectures and reduce the covariance shift. At a frame rate of 3Hz, ultrasound videos were separated into images. Before being scaled to 224x244 pixels, all photos were cropped to a quadratic window, eliminating measure bars, text, and artifacts on the borders. Fig.8

shows samples of ultrasound images after preprocessing. To avoid overfitting, data augmentation adjustments such as rotations (up to 10 degrees), flips (horizontally and vertically), and shifts (up to 10 percent) were implemented. Our dataset consisted of 2941 images after video sampling and image preprocessing, with 1305 images corresponding to COVID-19, 446 images corresponding to bacterial pneumonia, and 1190 images corresponding to healthy images. Fig.9 shows the number of images in each class after video sampling.

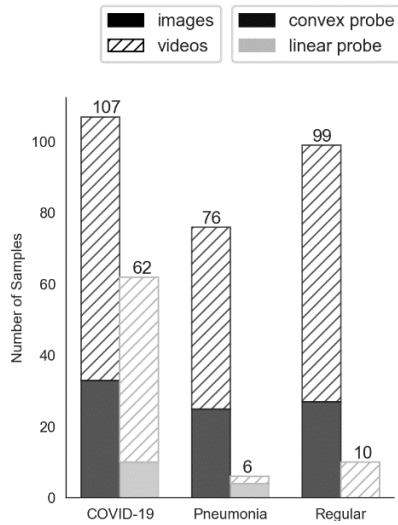

Fig. 7. Distribution of images and videos. Most samples use the convex probe.

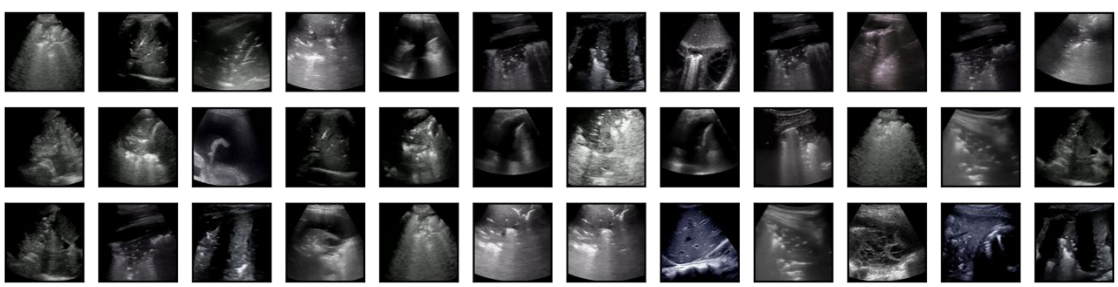

Fig. 8. Sample of ultrasound images in the dataset.

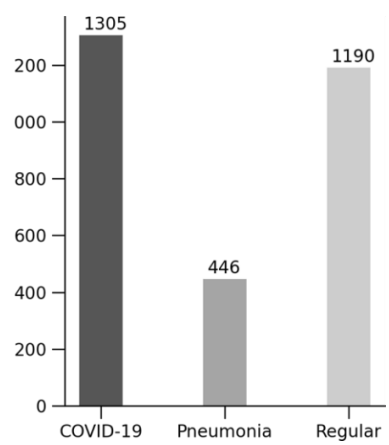

Fig. 9. The number of images in each class after video sampling.

Experiments were conducted on Google Colab with an NVIDIA Tesla P100 PCIe 16GB GPU. Additionally, Python was used to implement each architecture using Keras library.

All findings were obtained using 5-fold cross-validation using the number of samples per class. Because the data was split at the patient-level [38], it was ensured that the frames of a single video was only present at one-fold and that the number of videos per class was consistent across all folds

Fig.10 shows the leave-one-out technique of the 5-fold cross-validation. Cross-validation is a method of repeatedly rebuilding the architecture using different combinations of training and testing data. The accuracy results for each fold are averaged. The advantage of n-fold cross-validation is that the training data is maximized, and it is deterministic. The disadvantage is that it is extremely computationally intensive.

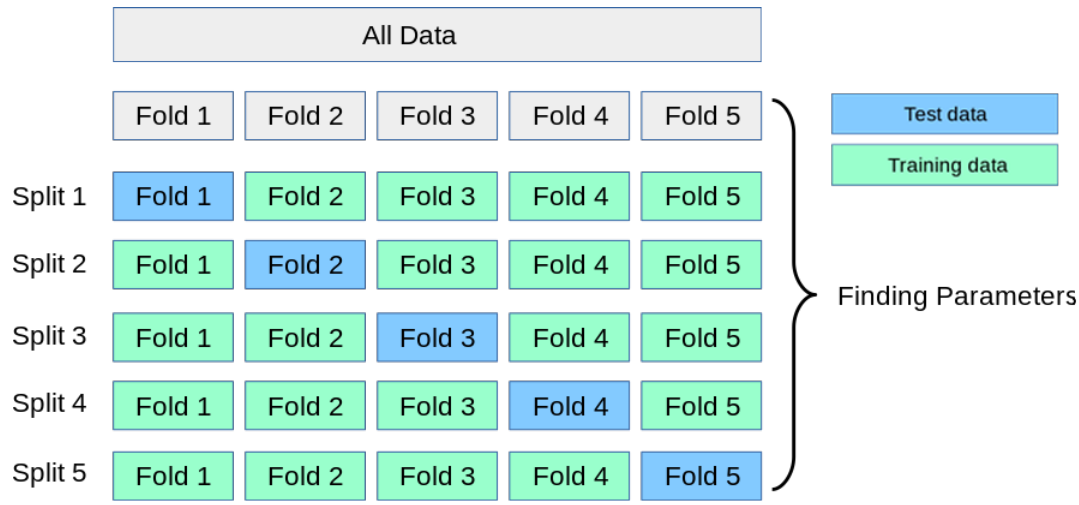

Fig. 10. Leave-one-out technique of the 5-fold cross-validation.

4.2. *Evaluation Metrics*

Different performance metrics were used to assess the trained architectures includes precision, recall, F1-Score, MCC, specificity, accuracy, and balanced accuracy for each architecture.

Precision is the ratio between the true positives (TP) and all the positives (i.e., true positives and false positives (FP)). It is the percentage of correctly classified cases among those classified as positive. (1) shows the formula to calculate the precision. In contrast to precision, which only focuses on the correct true positives out of all positive predictions, recall indicates missing positive predictions. Recall, which is also called sensitivity, is the ratio of correctly predicted positive cases (i.e., true positives) to the actual class observations (i.e., true positives and false negatives (FN)). (2) shows the formula to calculate the recall. The number of the actually correct predictions is indicated by both precision and recall. False-positives are considered in the precision metric, whereas false-negatives are considered in the recall metric.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

The weighted average of precision and recall is the F1-score. Neither accuracy nor recall tell the whole story. We might have high precision with low recall or low precision with high recall. The F1-score allows you to convey both concerns with a single metric. (3) shows the formula to calculate the recall.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

The Matthews correlation coefficient (MCC) is a more accurate statistical rate that only gives a high score if the prediction is correct in all four areas of the confusion matrix (true-positives, false-negatives, true-negatives, and false-positives) [60]. (4) shows the formula to calculate MCC.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4}$$

Specificity is the ability of a test to correctly identify people without the disease. It is the percentage of cases without the disease that are classified as negative. (5) shows the formula to calculate the specificity.

$$Specif icity = \frac{TN}{TN + FP} \tag{5}$$

Accuracy and balanced accuracy (for data from classes with imbalances) show the proportion of correct predictions among all data samples and are used to evaluate the overall approach performance. (6) shows the formula to calculate the accuracy, and (7) shows the formula to calculate the balanced accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Balanced\ Accuracy = \frac{Recall + Specif icity}{2} \tag{7}$$

The confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. It reports the number of correct and incorrect predictions for each class.

4.3. *Statistical Analysis*

In medical image analysis and computer vision applications, the ability to quantify conditions of high uncertainty is critical. The mean 5-fold cross-validation results from the various CNN architectures are shown in Table 2. It is worth noting that for COVID19 classification, it is more crucial to lower the number of false negatives (high recall) than false positives (low recall).

InceptionV1 outperformed the architectures with accuracy and balanced accuracy of 84.3% and 81.8%, respectively. It exceeds the accuracy of POCOVID-Net by 1.7% and even uses parameters 2.6 times less, with just 5.6M parameters, which has a huge effect on time and space complexity. Xception CNN achieves the highest precision in classifying COVID-19 and pneumonia with 93% and 80%, respectively. Most CNNs are struggling to distinguish between COVID-19 and pneumonia. In contrast, ResNeXt50 achieved the worst accuracy and balanced accuracy of 74.5% and 71.9%, respectively.

Note as well that Inception-based architectures, particularly InceptionV1, InceptionV4, and Xception, achieve higher accuracy and balanced accuracy values than other architectures. This can be related to the architecture's ability to learn spatial patterns and detect features at varying scales. The common layers in Inception-based architectures are the inception module. This module combines different filter sizes into a single image block rather than being limited to a single filter size, which we then concatenate and pass onto the next layer. Furthermore, this goes in line with the findings of Diaz-Escobar et al. [39]. Fig.11 shows the confusion matrices of architectures with high performance, including InceptionV1, InceptionV4, Xception and POCOVID-Net.

Table 2. A comparison of the different CNN architectures using 5-fold cross-validation. The best architecture for each class and column is shown in bold.

Architecture	Classes	Precision	Recall	F1-Score	MCC	Specificity
LeNet-5 Accuracy: 76.6% Balanced: 72.3% #Params: 0.08 M	COVID-19	0.77±0.16	0.83±0.10	0.79±0.13	0.59±0.29	0.76±0.20
	Pneumonia	0.79±0.16	0.62±0.14	0.68±0.12	0.65±0.13	0.97±0.02
	Healthy	0.73±0.29	0.72±0.31	0.72±0.30	0.59±0.41	0.87±0.11
AlexNet Accuracy: 78.9% Balanced: 74.9% #Params: 58.3 M	COVID-19	0.75±0.09	0.84±0.19	0.79±0.13	0.61±0.22	0.77±0.10
	Pneumonia	0.79±0.18	0.65±0.12	0.72±0.14	0.68±0.16	0.97±0.02
	Healthy	0.85±0.20	0.75±0.14	0.79±0.14	0.69±0.20	0.90±0.13
InceptionV1 Accuracy: **84.3%** Balanced: **81.8%** #Params: 5.6 M	COVID-19	0.84±0.10	**0.88±0.09**	**0.86±0.09**	**0.73±0.20**	0.84±0.13
	Pneumonia	0.80±0.14	0.77±0.14	0.76±0.09	0.74±0.09	0.97±0.02
	Healthy	**0.86±0.17**	0.80±0.15	0.83±0.15	0.74±0.21	**0.93±0.07**
InceptionV3 Accuracy: 77.2% Balanced: 75.1% #Params: 21.8 M	COVID-19	0.77±0.23	0.78±0.24	0.77±0.22	0.59±0.41	0.79±0.19
	Pneumonia	0.70±0.23	0.73±0.23	0.71±0.23	0.65±0.29	0.94±0.06
	Healthy	0.81±0.27	0.74±0.26	0.77±0.25	0.66±0.37	0.89±0.13
ResNet50 Accuracy: 78.8% Balanced: 73.2% #Params: 23.6 M	COVID-19	0.79±0.14	0.86±0.11	0.81±0.12	0.62±0.28	0.75±0.24
	Pneumonia	0.70±0.12	0.63±0.17	0.66±0.15	0.61±0.16	0.96±0.02
	Healthy	0.83±0.21	0.71±0.29	0.74±0.25	0.66±0.32	0.92±0.10
InceptionV4 Accuracy: 84.2% Balanced: 80.9% #Params: 41.1 M	COVID-19	0.86±0.11	0.83±0.08	0.84±0.09	0.71±0.17	0.87±0.12
	Pneumonia	0.80±0.10	0.76±0.16	0.78±0.12	0.74±0.14	0.96±0.05
	Healthy	0.84±0.10	0.83±0.18	0.83±0.14	**0.75±0.16**	0.90±0.07
Inception-ResNet-V2 Accuracy: 79.6% Balanced: 75.6% #Params: 54.3 M	COVID-19	0.79±0.22	0.77±0.27	0.78±0.24	0.63±0.37	0.86±0.11
	Pneumonia	0.73±0.16	0.64±0.21	0.66±0.17	0.61±0.20	0.94±0.06
	Healthy	0.82±0.22	0.85±0.17	0.83±0.20	0.73±0.31	0.88±0.15
DenseNet121 Accuracy: 79.9% Balanced: 78% #Params: 7 M	COVID-19	0.83±0.20	0.75±0.28	0.78±0.26	0.66±0.34	0.91±0.07
	Pneumonia	0.70±0.17	0.74±0.11	0.71±0.13	0.66±0.17	0.93±0.07
	Healthy	0.80±0.20	0.85±0.14	0.82±0.17	0.70±0.29	0.85±0.16
ResNext50 Accuracy: 74.5% Balanced: 71.9% #Params: 26.5 M	COVID-19	0.76±0.16	0.79±0.16	0.77±0.15	0.55±0.32	0.76±0.22
	Pneumonia	0.69±0.28	0.66±0.26	0.64±0.21	0.58±0.28	0.90±0.09
	Healthy	0.83±0.14	0.71±0.27	0.74±0.22	0.66±0.23	0.92±0.05
Xception Accuracy: 78% Balanced: 76.5% #Params: 20.8 M	COVID-19	**0.93±0.05**	0.71±0.36	0.72±0.36	0.66±0.33	**0.94±0.07**
	Pneumonia	**0.80±0.18**	0.65±0.34	0.61±0.32	0.59±0.31	**0.97±0.03**
	Healthy	0.81±0.26	**0.94±0.07**	**0.84±0.20**	0.72±0.36	0.77±0.39
POCOVID-Net Accuracy: 82.6% Balanced: 81.7% #Params: 14.7 M	COVID-19	0.82±0.12	0.85±0.16	0.83±0.14	0.68±0.25	0.84±0.10
	Pneumonia	0.79±0.07	**0.84±0.10**	**0.81±0.07**	**0.77±0.09**	0.95±0.05
	Healthy	0.84±0.15	0.77±0.16	0.80±0.16	0.71±0.21	0.93±0.05

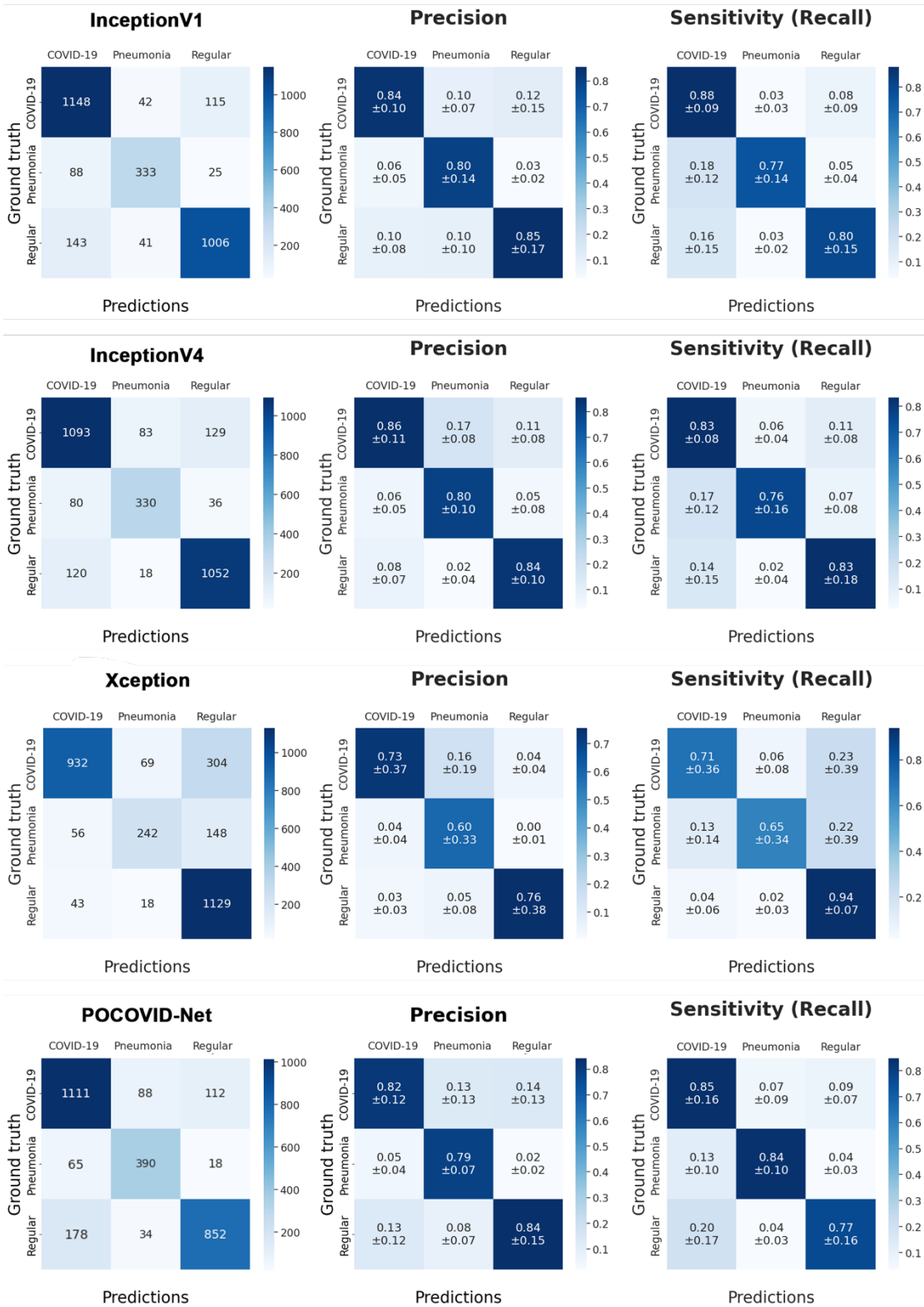

Fig. 11. 5-fold cross-validation results: Confusion Matrices.

5. Conclusion

Lung ultrasound is a potentially high-impact method for COVID-19 diagnosis. This paper compares and analyzes numerous CNN architectures using ultrasound images; publicly available in the POCUS dataset. On this basis, we conclude that CNNs can classify ultrasound images with high accuracy, which could be beneficial as an assistant tool for radiologists. The analysis emphasizes the importance of the Inception architectures. Future work should consider the potential effects of image processing in the process of COVID-19 diagnosis. In addition, a more robust and homogeneous dataset will lead to better results. Furthermore, it will be interesting to investigate the enhancements in classification results when using transfer learning and pre-trained architectures.

References

[1] M. Cascella, M. Rajnik, A. Aleem, S. C. Dulebohn, and R. Di Napoli, "Features, Evaluation, and Treatment of Coronavirus (COVID-19)," in StatPearls, Treasure Island (FL): StatPearls Publishing, 2022. Accessed: Jul. 13, 2022. Available: http://www.ncbi.nlm.nih.gov/books/NBK554776/

[2] WHO COVID-19 Situation Reports. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports

[3] WHO Director-General's opening remarks at the media briefing on COVID-19 - 18 August 2020. Available: https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---18-august-2020

[4] Y. Zhao *et al.*, "COVID19: A Systematic Approach to Early Identification and Healthcare Worker Protection," *Front. Public Health*, vol. 8, p. 205, May 2020.

[5] M. Yüce, E. Filiztekin, and K. G. Özkaya, "COVID-19 diagnosis —A review of current methods," Biosensors and Bioelectronics, vol. 172, p. 112752, Jan. 2021.

[6] Y. Yang et al., "Evaluating the accuracy of different respiratory specimens in the laboratory diagnosis and monitoring the viral shedding of 2019-nCoV infections," Infectious Diseases (except HIV/AIDS), preprint, Feb. 2020.

[7] X. Mei et al., "Artificial intelligence–enabled rapid diagnosis of patients with COVID-19," Nat Med, vol. 26, no. 8, pp. 1224–1228, Aug. 2020.

[8] P. Parag and T. C. Hardcastle, "Interpretation of emergency CT scans in polytrauma: trauma surgeon vs radiologist," African Journal of Emergency Medicine, vol. 10, no. 2, pp. 90–94, Jun. 2020.

[9] H. E. Davies, C. G. Wathen, and F. V. Gleeson, "The risks of radiation exposure related to diagnostic imaging and how to minimise them," BMJ, vol. 342, no. feb25 1, pp. d947–d947, Feb. 2011.

[10] A. Fucic, G. Brunborg, R. Lasan, D. Jezek, L. E. Knudsen, and D. F. Merlo, "Genomic damage in children accidentally exposed to ionizing radiation: A review of the literature," Mutation Research/Reviews in Mutation Research, vol. 658, no. 1–2, pp. 111–123, Jan. 2008.

[11] J. Qu, W. Yang, Y. Yang, L. Qin, and F. Yan, "Infection Control for CT Equipment and Radiographers' Personal Protection During the Coronavirus Disease (COVID-19) Outbreak in China," American Journal of Roentgenology, vol. 215, no. 4, pp. 940–944, Oct. 2020.

[12] MB Weinstock, A Echeniqu, and JW Russel, "Chest X-Ray Findings in 636 Ambulatory Patients with COVID-19 Presenting to an Urgent Care Center: A Normal Chest X-Ray Is no Guarantee," The Journal of Urgent Care Medicine, vol. 14, pp. 13–18, 2020.

[13] V. Vespro et al., "Chest X-ray findings in a large cohort of 1117 patients with SARS-CoV-2 infection: a multicenter study during COVID-19 outbreak in Italy," Intern Emerg Med, vol. 16, no. 5, pp. 1173–1181, Aug. 2021.

[14] F. Mojoli, B. Bouhemad, S. Mongodi, and D. Lichtenstein, "Lung Ultrasound for Critically Ill Patients," Am J Respir Crit Care Med, vol. 199, no. 6, pp. 701–714, Mar. 2019.

[15] R. Raheja, M. Brahmavar, D. Joshi, and D. Raman, "Application of Lung Ultrasound in Critical Care Setting: A Review," Cureus, Jul. 2019.

[16] A. Vishwa and S. Sharma, "Modified Method for Denoising the Ultrasound Images by Wavelet Thresholding," IJISA, vol. 4, no. 6, pp. 25–30, Jun. 2012.

[17] R. R. de Oliveira, T. P. Rodrigues, P. S. D. da Silva, A. C. Gomes, and M. C. Chammas, "Lung ultrasound: an additional tool in COVID-19," Radiol Bras, vol. 53, no. 4, pp. 241–251, Aug. 2020.

[18] S. K. Zhou et al., "A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises," Proc. IEEE, vol. 109, no. 5, pp. 820–838, May 2021.

[19] J. Pare *et al.*, "Point-of-care Lung Ultrasound Is More Sensitive than Chest Radiograph for Evaluation of COVID-19," *WestJEM*, vol. 21, no. 4, Jun. 2020.

[20] Chinese Critical Care Ultrasound Study Group (CCUSG), Q.-Y. Peng, X.-T. Wang, and L.-N. Zhang, "Findings of lung ultrasonography of novel corona virus pneumonia during the 2019–2020 epidemic," Intensive Care Med, vol. 46, no. 5, pp. 849–850, May 2020.

[21] M. J. Fiala, "Ultrasound in COVID-19: a timeline of ultrasound findings in relation to CT," Clinical Radiology, vol. 75, no. 7, pp. 553–554, Jul. 2020.
[22] Y. Yang, Y. Huang, F. Gao, L. Yuan, and Z. Wang, "Lung ultrasonography versus chest CT in COVID-19 pneumonia: a two-centered retrospective comparison study from China," Intensive Care Med, vol. 46, no. 9, pp. 1761–1763, Sep. 2020.
[23] A. W. E. Lieveld et al., "Diagnosing COVID-19 pneumonia in a pandemic setting: Lung Ultrasound versus CT (LUVCT) – a multicentre, prospective, observational study," ERJ Open Res, vol. 6, no. 4, pp. 00539–02020, Oct. 2020.
[24] Y. Tung-Chen et al., "Correlation between Chest Computed Tomography and Lung Ultrasonography in Patients with Coronavirus Disease 2019 (COVID-19)," Ultrasound in Medicine & Biology, vol. 46, no. 11, pp. 2918–2926, Nov. 2020.
[25] S. Liu et al., "Deep Learning in Medical Ultrasound Analysis: A Review," Engineering, vol. 5, no. 2, pp. 261–275, Apr. 2019.
[26] R. J. G. van Sloun and L. Demi, "Localizing B-Lines in Lung Ultrasonography by Weakly Supervised Deep Learning, In-Vivo Results," IEEE J. Biomed. Health Inform., vol. 24, no. 4, pp. 957–964, Apr. 2020.
[27] X. Wang, J. S. Burzynski, J. Hamilton, P. S. Rao, W. F. Weitzel, and J. L. Bull, "Quantifying lung ultrasound comets with a convolutional neural network: Initial clinical results," Computers in Biology and Medicine, vol. 107, pp. 39–46, Apr. 2019.
[28] S. Kulhare *et al.*, "Ultrasound-Based Detection of Lung Abnormalities Using Single Shot Detection Convolutional Neural Networks," in *Simulation, Image Processing, and Ultrasound Systems for Assisted Diagnosis and Navigation*, vol. 11042, D. Stoyanov, Z. Taylor, S. Aylward, J. M. R. S. Tavares, Y. Xiao, A. Simpson, A. Martel, L. Maier-Hein, S. Li, H. Rivaz, I. Reinertsen, M. Chabanas, and K. Farahani, Eds. Cham: Springer International Publishing, 2018, pp. 65–73.
[29] Y. Xu *et al.*, "Boundary Restored Network for Subpleural Pulmonary Lesion Segmentation on Ultrasound Images at Local and Global Scales," *J Digit Imaging*, vol. 33, no. 5, pp. 1155–1166, Oct. 2020.
[30] R. Barrientos et al., "Automatic detection of pneumonia analyzing ultrasound digital images," in 2016 IEEE 36th Central American and Panama Convention, San José, Costa Rica, Nov. 2016, pp. 1–4.
[31] P. Cisneros-Velarde et al., "Automatic pneumonia detection based on ultrasound video analysis," in 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, Aug. 2016, pp. 4117–4120
[32] M. Correa et al., "Automatic classification of pediatric pneumonia based on lung ultrasound pattern recognition," PLoS ONE, vol. 13, no. 12, p. e0206410, Dec. 2018.
[33] C. Mehanian et al., "Deep Learning-Based Pneumothorax Detection in Ultrasound Videos," in Smart Ultrasound Imaging and Perinatal, Preterm and Paediatric Image Analysis, vol. 11798, Q. Wang, A. Gomez, J. Hutter, K. McLeod, V. Zimmer, O. Zettinig, R. Licandro, E. Robinson, D. Christiaens, E. A. Turk, and A. Melbourne, Eds. Cham: Springer International Publishing, 2019, pp. 74–82.
[34] L. Carrer et al., "Automatic Pleural Line Extraction and COVID-19 Scoring From Lung Ultrasound Data," IEEE Trans. Ultrason., Ferroelect., Freq. Contr., vol. 67, no. 11, pp. 2207–2217, Nov. 2020.
[35] C.-H. Chen et al., "Computer-aided diagnosis of endobronchial ultrasound images using convolutional neural network," Computer Methods and Programs in Biomedicine, vol. 177, pp. 175–182, Aug. 2019.
[36] S. Roy et al., "Deep Learning for Classification and Localization of COVID-19 Markers in Point-of-Care Lung Ultrasound," IEEE Trans. Med. Imaging, vol. 39, no. 8, pp. 2676–2687, Aug. 2020.
[37] J. Born et al., "POCOVID-Net: Automatic Detection of COVID-19 From a New Lung Ultrasound Imaging Dataset (POCUS)", Jan. 2021, Accessed: Feb. 02, 2021.
[38] J. Born et al., "Accelerating Detection of Lung Pathologies with Explainable Ultrasound Image Analysis," Applied Sciences, vol. 11, no. 2, p. 672, Jan. 2021.
[39] J. Diaz-Escobar *et al.*, "Deep-learning based detection of COVID-19 using lung ultrasound imagery," *PLoS ONE*, vol. 16, no. 8, p. e0255886, Aug. 2021.
[40] G. Soldati et al., "Proposal for International Standardization of the Use of Lung Ultrasound for Patients With COVID -19: A Simple, Quantitative, Reproducible Method," J Ultrasound Med, vol. 39, no. 7, pp. 1413–1419, Jul. 2020.
[41] https://github.com/jannisborn/covid19_pocus_ultrasound/tree/master/data, Accessed: Jul. 13, 2022.
[42] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights Imaging, vol. 9, no. 4, pp. 611–629, Aug. 2018.
[43] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI," J. Magn. Reson. Imaging, vol. 49, no. 4, pp. 939–954, Apr. 2019.
[44] G. Litjens et al., "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, Dec. 2017.
[45] A. Fourcade and R. H. Khonsari, "Deep learning in medical image analysis: A third eye for doctors," Journal of Stomatology, Oral and Maxillofacial Surgery, vol. 120, no. 4, pp. 279–288, Sep. 2019.
[46] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017.
[48] C. Szegedy et al., "Going deeper with convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, Jun. 2015, pp. 1–9.

[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", Dec. 2015, Accessed: Feb. 13, 2022.

[51] . Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", Aug. 2016, Accessed: Feb. 17, 2022.

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, Jul. 2017, pp. 2261–2269.

[53] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks", Apr. 2017, Accessed: Feb. 17, 2022.

[54] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", Apr. 2017, Accessed: Feb. 13, 2022.

[55] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", Jan. 2017, Accessed: May 04, 2022.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014, Accessed: May 04, 2022.

[57] H. Iqbal, HarisIqbal88/PlotNeuralNet v1.0.0. v1.0.0, Zenodo, 2018.

[58] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Mar. 2015, Accessed: May 04, 2022.

[59] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Apr. 2015, Accessed: May 04, 2022.

[60] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020.