IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 April 2024, accepted 26 May 2024, date of publication 4 June 2024, date of current version 11 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3409566

RESEARCH ARTICLE

Metric-Based Frame Selection and Deep Learning Model With Multi-Head Self Attention for Classification of Ultrasound Lung Video Images

EBRAHIM A. NEHARY, (Graduate Student Member, IEEE), SREERAMAN RAJAN, (Senior Member, IEEE), AND CARLOS ROSSA, (Senior Member, IEEE)
Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: Ebrahim A. Nehary (ebrahimali@cmail.carleton.ca)

This work was supported in part by the Marion Surgical under Grant ALLRP 550307-20, and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

ABSTRACT Detection of COVID-19 manifestations in lung ultrasound (US) images has gained attention in recent times. The current state-of-the-art technique for distinguishing a healthy lung from COVID-19 infected or bacterial pneumonia infected lung uses non-adjacent frames or equally spaced frames from the video. However, the frame content or correlation between the selected frames has not been taken into consideration for frame selection. In this paper, a metric-based frame selection approach is proposed for three-way classification of lung US videos, and the influence of the frame selection method on image classification accuracy is studied. A deep learning model comprising of a pre-trained model (VGG16) for feature extraction, multi-head attention for feature calibration, global averaging for feature reduction, and a dense layer for classification is proposed. The pre-trained model is re-trained using cross-entropy loss with balanced weights to handle class imbalance. Two types of classification approaches are considered: i) few frames in a video are selected using the proposed metrics; and (ii) all frames in a video are considered. With VGG16 as the pre-trained model, a mean balanced sensitivity of COVID-19, bacterial pneumonia, and healthy classes with 0.82, 0.89, and 0.87, respectively was achieved using 5-fold cross-validation. The results show that even random selection of frames performs better than fixed frame selection and the proposed frame selection method outperforms the state-of-art fixed frame selection irrespective of the type of backbone model used for lung US classification.

INDEX TERMS Ultrasound, COVID-19, frame selection, deep learning, frame classification, video classification, entropy, SVD.

I. MOTIVATION AND LITERATURE SURVEY

COVID-19, which was declared as a pandemic by the World Health Organization (WHO), led to worldwide shutdowns. By mid-2022, the number of reported cases exceeded half a billion and the virus caused more than 6 million deaths across the globe [1]. The virus severely affected the respiratory system, and its manifestations ranged from minor symptoms, like cough and fever, to organ failure and death [2], [3], [4], [5]. The virus spread via direct contact with an infected person or indirectly through airborne or droplet transmission [6], [7].

Effective and reliable screening is essential to limit the spread of the virus and to provide immediate treatment to those infected [5], [8], [9], [10], [11], should the pandemic situation recur. The reverse transcription-polymerase chain reaction (RT-PCR) is the gold standard test used to detect COVID-19 [12]. However, this test is laborious and takes

The associate editor coordinating the review of this manuscript and approving it for publication was Sandra Costanzo.

a long time to provide the results [13]. In addition, RT-PCR tests have a sensitivity as low as 70% with high statistical variance [8], [14], [15], [16], [17], [18]. According to [19], medical imaging may be employed as an alternative or complementary screening tool for COVID-19 and may provide quicker results than RT-PCR tests.

Computed Tomography (CT) and chest X-rays (CXR) have been used for screening pulmonary diseases [20], including COVID-19. However, these modalities are ionizing, expensive, not portable, and require transferring the patients between different rooms or facilities for imaging purposes, which can result in the further spreading of the disease [5], [9]. An alternative to these imaging modalities is lung ultrasound (LUS) [21], [22]. An LUS machine is portable and therefore, an examination of the lung can be performed at the bedside. As LUS machines are also easy to disinfect when compared to X-ray or CT machines, the time between examinations is significantly reduced. LUS machines are also low-cost, non-ionizing and non-invasive, making them accessible in a wide range of facilities [23], [24].

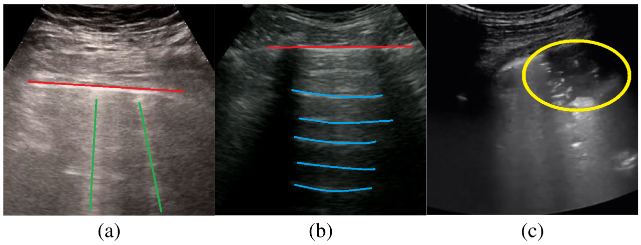

FIGURE 1. Samples of US images with different types of lines. A-line shows in blue, pleural line in red, B-line in green, and consolidation is the areas inside the yellow circle.

LUS has proven its effectiveness in diagnosing and monitoring a wide variety of respiratory diseases for decades, such as differentiating between bacterial and viral pneumonia in children [25], [26], [27]. LUS can differentiate between viral pneumonia (including COVID-19) and bacterial pneumonia based on pathological distinctions [28], [29], [30]. There are several studies that indicate the validity of the use of lung ultrasound to assess COVID-19 and its related effects [28], [31], [32], [33]. Deep learning has been used to classify LUS images as belonging to one of the three classes, namely a lung infected with Covid-19, a lung infected with bacterial pneumonia, or a healthy lung [3], [9], [19], [34], [35], [36], [37] or classifying LUS images into one of the following three classes: COVID-19, Non-COVID acute respiratory distress syndrome (NCOVID), and Hydrostatic pulmonary edema (HPE) [38]. Deep learning has also been used to predict the severity of the COVID-19 infection in the LUS images using the scores proposed in [39], [40], [41], [42], and [43]. Nevertheless, the scoring of severity primarily relies on observable patterns, rendering it appropriate solely for monitoring purposes [9]. Unfortunately, there are no standard protocols for even assessing the severity of COVID-19 from LUS currently [31]. Furthermore, there are no standard databases that could be used in any of the studies for deep learning. The only publicly available dataset with LUS video clips from different sources have been aggregated in [9]. However, this database is heterogeneous but is still useful for deep learning purposes. Using this database, one may be able to develop deep learning algorithm to be in place when protocols for severity assessment and databases with a large population study is publicly available for research use.

Classification methods aim at identifying features that differentiate a healthy lung from one that is infected by the virus. These features include the pleural line (the interface between the wall and the lung tissue), A-lines (hyperechoic horizontal lines in the images), B-lines (hyperechoic vertical lines), and consolidations (hypoechoic regions in the images) [3], [11], [44], [45], as shown in Figure 1. Several deep learning methods have been devised to detect B-lines in the images, which are a distinct indication of infection [46], [47], [48], [49]. As an example, in [34] these spatial features were extracted from the images using convolutional neural networks (CNN) and input to a Long Short-Term Memory (LSTM) algorithm to exploit their temporal dependency. Alternatively, it is also possible to classify the images without extracting such features [50].

Current state-of-the-art methods available in the literature for detecting COVID-19 in LUS images are as follows. Two frame-based classification models were proposed in [9]. In the first model, every frame in the video was classified while in the second model, the classification result was provided for a set of successive five frames at a time. The selected frames were treated as five input channels for the classification model. However, at the video level, one would not know if the whole video contained a healthy lung or one with Covid-19 or bacterial pneumonia. It should be noted that these five frames were correlated and therefore, each of the frames would not necessarily provide any new information to the classification model. The proposed classification model in [9] used VGG16, a global average layer followed by a densenet.

In [34], instead of successive frames, a set of nonadjacent, equally spaced frames were used for classification, with spatial features extracted by CNN and temporal features extracted by Long-short term memory (LSTM). With a view to handling learning with limited data, a two-step process is proposed: first biomarkers (B-line and A-line) were obtained using a model trained with a private dataset, and in the second stage, LUS was classified using a model guided by the biomarkers from the first stage [47]. In both steps, a CNN was used to extract the spatially encoded features used as input to a transformer encoder to capture temporal features across the frames. A set of frames was selected for classification based on the assumption that all the videos have almost the same frame rate. These selected frames were simultaneously (multiple streams) used for training the model.

Several deep learning models such as POCOVID-NET, Mini-COVIDNet were proposed in [19] and trained on each frame that was selected from the video under the assumption that all the videos have the same frame-rate.

From every video, a set of frames separated by a constant number of frames was selected for training. Although the model achieved a good result for COVID-19, and bacterial pneumonia-infected lung classes, it yielded a poor result for the healthy lung class. Linear Vision transformer was implemented in [51] to classify the US images into binary classes (bacterial pneumonia vs COVID-19, healthy vs bacterial pneumonia and healthy vs Covid-19) and multiclass (bacterial pneumonia, COVID-19 infected, healthy lungs). However, it is unclear if the frames of a given video appeared both in the training and the testing sets and also no information is available about the selection of frames for training. A multi-layer fusion model was proposed in [52] and trained using an artifact-free dataset. However, the training procedure was not elaborated. Readers are further directed to [37] for a detailed review of deep learning applications in lung ultrasound imaging of COVID-19 patients.

Despite ultrasound imaging being safe, radiation-free, relatively inexpensive and portable, the classification of COVID-19 using LUS images has only attracted limited research. Most papers available in the literature follow the same procedure: extract frames from a video, aggregate all the frames, and divide them into a training set and a test set. Unfortunately, this leads to data leakage and therefore, results cannot be trusted and trained models cannot be generalized. Instead, a video-based data division into training and test sets would yield better-generalized models. As there is no standardized dataset available for training and testing, the dataset is generally formed by collecting several publicly available images. These datasets are obtained using different ultrasound machines, under different conditions. This introduces heterogeneity in the dataset, which may not yield high classification accuracy.

In the context of natural videos, various deep learning models have been proposed to summarize them by selecting specific frames as representations. These models encompass recurrent models [53], convolutional networks [54], reinforcement learning approaches [55], [56], and adversarial networks [57]. For example, the hierarchical recurrent neural network and tensor-trained embedding layer [53], pre-trained models for feature extraction, graph and assignment-learning graph using graph neural networks [54], and reinforcement learning with diversity and representation reward function [56] have been proposed for video summarization. However, natural videos differ from ultrasound videos. Natural videos can be summarized with less ambiguity, and generally, there would be no subjectiveness. In order to summarize an ultrasound video, one would seek anatomical features and artifacts relevant to diagnosis. However, diagnoses are often subjective. Therefore, an ensemble deep learning approach with reinforcement learning has been proposed for ultrasound (LUS) video summarization [55]. It consists of an ensemble of a classification model, a segmentation model, and an autoencoder model. The objective of the work in [55] is to compress LUS videos using the ensemble model, including the classification model, while the objective of the proposed method is to summarize the video to remove redundancy and improve classification results. Also, the models presented in [55] cannot be directly applied to the study proposed in this paper because the videos are limited in number and are of varying time duration with different frame rates, and there are no classification masks available in the dataset.

A significant limitation of previously published work is that frames are extracted from the LUS videos generally at regular time intervals, regardless of the information they contain. Irrespective of whether the frames were extracted from the videos with regular time intervals or at irregular time intervals, no consideration was given to the correlation that is inherently present between the frames. Both the quality of the images in the frame and the correlation between frames will affect the training process. Frame selection has strong implications for classification accuracy. However, this dependency has not been addressed in the current literature in detail.

Recognizing the importance of frame selection for classifying LUS images, in this paper, we propose a method to select frames from a video based on the quantitative information content of each frame individually or the similarity of a series of adjacent frames. Furthermore, we propose a deep learning model with multi-head self-attention to automatically extract features and refine them to remove redundancy. In this work, frame-based training is adopted where a frame, along with its class label, is presented. This training strategy is adopted to circumvent training issues with a limited amount of videos when video-based training is considered. The training set and test set are formed using distinctly different videos. This is made to ensure that there is no data leakage. The models are trained with a fixed number of chosen frames from each video in the training set using the class label provided for the corresponding videos.

For testing, three different strategies are considered: (a) classify individual chosen frames of a video; (b) classify the video based on the classification of the chosen frames; (c) classify the video based on the classification of all the frames. The latter two are termed video classification, while the first method is termed frame classification. To achieve video classification, the model output for frames of videos in the test set that are used as input is collected, and a voting process is used to determine the class of the test video. To summarize, the main contributions of this work are the following:

(a) A metric-based technique to choose the frames for classification training and testing is proposed. We consider several metrics and compare their effect on classification performance with current state-of-the-art methods. Our metric-based frame choice provides superior performance than the constant frame rate or non-adjacent selection currently adopted in the literature.

(b) A new algorithm that uses a combination of pre-trained models with multi-head self-attention for feature extraction, redundancy removal, and classification is proposed. Experimental evaluation shows that frame selection with our proposed model outperforms state-of-the-art algorithms when the balance performance of three classes is taken into account.

(c) We used various pre-training models in the new classification model as proposed in b) and compared their results. In addition, we implement the model proposed in [19] and compare the performance of our algorithm against it.

II. FRAME SELECTION

Our proposed deep learning model is used to classify the COVID-19 US images into three classes, namely, COVID-19, bacterial pneumonia, and healthy. In our work, we propose various criteria for selecting frames from the video and then employ the chosen frames to train the model using frame-based training.

A. CURRENT STATE-OF-THE-ART FRAME SELECTION CRITERIA

Since this work proposes using metric-based selection criteria to choose frames from the videos for frame-based training, we first summarize the previously reported methods, namely constant frame rate and non-adjacent methods.

Assuming that all videos have a constant frame rate (f_c), the frame rate of extraction (f_e) for constant frame rate selection is given by

$$f_e = \frac{f_v}{f_c} \tag{1}$$

where f_v is the frame rate of the video. The immediate non-overlapping f_e frames are considered for the next frame selection, and the process is repeated till all frames are exhausted. Another method proposed in [34] chooses frames with the condition that the selected frames should not be adjacent to one another.

In these methods, frames are selected based on their temporal or spatial information (merely the position or location of the frame in the video) regardless of their features. In contrast, in our proposed work, a set of frames is selected based on certain well-defined metrics that give some measure of the frame's content.

B. PROPOSED FRAME SELECTION CRITERIA

The proposed metrics are categorized into four different groups.

1) *Norm-based metrics:* Metrics related to the norm of the frames;
2) *Entropy and Fractal dimension-based metrics:* Metrics based on entropy and fractal dimensionality of the frames;
3) *Similarity-based metrics:* Metrics related to similarity measures between frames;
4) *Other metrics*: Metrics that do not belong to the previous three groups.

The description of chosen metrics under each of the groups is described below.

1) NORM-BASED METRICS

- Sum of singular values: The energy in the frame is considered a metric for frame selection. In order to determine the energy in the frame, the sum of singular values (SVD) or Frobenius norm is considered. The SVD of frame f_i is given

$$SVD_{f_i} = U\Sigma V^T \tag{2}$$

where the U and V are unitary matrices containing the right and left singular vectors respectively and Σ is the diagonal matrix containing the assorted singular values σ_{jj} in the descending order along its diagonal. Then, the energy measure is $\sum_{jj=1}^{r} \sigma_{jj}$ where r is the rank of the frame f_i.

- Frobenius norm: The Frobenius norm of frame f_i is

$$||F_{f_i}|| = \sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)} \tag{3}$$

where a_{ij} are the gray scale pixels values and m, n are dimensions of frame f_i.

To select the frames, the metric obtained from each frame is collected, and a vector of length equal to the number of frames in the video is formed. This vector is sorted in descending order (in case maxima are needed) or in ascending form (in case minima are needed). If l frames from the videos need to be selected, then the frames corresponding to the first l maxima or minima are selected. If the median of each frame needs to be considered, the following steps are followed: The vector with metrics is sorted, and the first median value is computed and removed from the vector. Again, the second median value is computed for the rest of the vector and removed from the vector. This operation is repeated until l such medians are calculated. The frames corresponding to those l medians are then selected from the video for training.

2) ENTROPY AND FRACTAL DIMENSION-BASED METRICS

Another criterion for frame selection would be considering the frame's entropy and fractal dimension. Petrosian (FD_p), Katz (FD_k), Higuchi (FD_h), and detrended (FD_d) metrics [58], [59], [60], [61], [62], [63] are considered. For computing these metrics, the frame is first vectorized by stacking all the rows and noted as a vectorized frame x.

- Petrosian (FD_p):

$$FD_p = \frac{\log_{10} n}{\log_{10} n + \log_{10}\left(\frac{n}{n+0.4N_d}\right)} \tag{4}$$

where n is the length of x and N_d is the number of sign changes in the first derivative of the x.

- Katz (FD_k):

$$FD_k = \frac{\log_{10}(\frac{L}{a})}{\log_{10}(\frac{d}{L}) + \log_{10}(\frac{L}{a})} \tag{5}$$

where L is the sum of the Euclidean distance between successive points in the vectorized frame x or the total length of the curve, d is the distance between the first point in the x and the point that provides the maximum distance, and a is the average distance between successive points in the vectorized frame x.

- Higuchi (FD_h): To compute FD_h, k series, x_m^k from the original vectorized frame $x(n) = x(1), x(2), \cdots, x(N)$ are constructed as follows:

$$x_m^k = \left\{x(m), x(m+k), \cdots, x(m + \left\lfloor \frac{N-m}{k} \right\rfloor k)\right\}$$

where $m = 1, 2, \cdots, k$, N is the vectorized frame length, m is the initial time value, k is the discrete the time interval between points, and $\lfloor \cdot \rfloor$ is an integer. Then

$$L(k) = \sum_{m=1}^{k} \frac{\sum_{i=1}^{A} |x(m+ik) - x(m+(i-1)k)|(n-1)}{\lfloor \frac{N-m}{k} \rfloor k} \tag{6}$$

where $A = \lfloor (N-m)/k \rfloor$ for each k is computed. Finally, FD_h is the linear regression between $\ln(L(k))$ versus $\ln(1/k)$.

- Detrended (FD_d):
To determine the detrended fluctuation function, FD_d, the following steps are taken:
1) Calculate the cumulative sum of the vectorized frame x.
2) Create a set of windows with sizes S ranging from 4 to N, where N is the length of the x, using logarithmic spacing.
3) For each window size $s \in S$, divide the cumulative sum signal into a set of windows W with length s and a 50% overlap.
4) Use linear regression to remove the linear trend from each window in W, to obtain the detrended signal.
5) Calculate the standard deviation of the detrended signal for each window in W.
6) Calculate the fluctuation function as the mean of the standard deviations of all the windows.
7) Calculate FD_d as the linear regression of the fluctuation function on a logarithmic axis.

In the entropy and fractal dimension-based metrics, the input frame is flattened to form a vector with a length equal to mn, where m, and n are the width and height of the frame, respectively. Then, these metrics are calculated and return a single value for each frame. The output is a vector with a length equal to the number of frames in the video. Similar to Norm-based metrics, the output vector is sorted, and the first l maximum, minimum, or median is selected, where l is equal to the number of frames extracted from each video.

3) SIMILARITY-BASED METRICS

Another method considers the relationship between frames in a video. For this method, we use the structural similarity index (SSIM) [64], and the peak signal ratio (PSNR) to find the association between frames within a video.

- SSIM: The SSIM between two frames x and y is given by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{7}$$

where μ_x is the mean and σ_x is the standard deviation of frame x, μ_y and σ_y are the mean and standard deviation of frame y, C_1 and C_2 are constants. The mean of a frame x is given by

$$\mu_x = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(i,j)$$

where m, n are width and height of the frame and $I(i,j)$ is intensity of the pixel (i,j) of the frame. While the standard deviation of frame x is

$$\sigma_x = \sqrt{\frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - \mu_x)^2}{mn}}$$

- PSNR: It is given by

$$PSNR(x, y) = 20 \log_{10} \frac{MAX}{\frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [x(i,j) - y(i,j)]^2} \tag{8}$$

where MAX is the maximum pixel intensity amongst the images of frames x and y, and m and n are the width and height of the image.

SSIM and PSNR require two frames to measure the similarity score between them. In our work, all pairs of combinations of frames from the same video are fed as input to SSIM or PSNR to compute the similarity score. For example, assuming that the video has only five frames, then SSIM is

$$SSIM(f1, \ldots, f5) = \begin{bmatrix} 0 & s_{12} & s_{13} & s_{14} & s_{15} \\ 0 & 0 & s_{23} & s_{24} & s_{25} \\ 0 & 0 & 0 & s_{34} & s_{35} \\ 0 & 0 & 0 & 0 & s_{45} \end{bmatrix} \tag{9}$$

where s_{ij} indicates the similarity score between frame i and frame j. Then, the zeros in the matrix are removed, and the remaining values are flattened to form a vector. This vector is sorted. Then, the first $l/2$ maximum, minimum, or median similarity scores are obtained if it is required to select l frames. For example, It is required to extract four frames based on the SSIM maximum. Assume the first maximum values are s_{14} and s_{35}. Then, frames 1, 3, 4, and 5 will be extracted from the video. In similar, PSNR has followed the same steps.

4) OTHER METRICS

Non-reference Image Quality Assessment (NIQA) and random selections are given for completeness.

- NIQA: It is employed to assess the frame quality using the scene statistics of locally normalized luminance coefficients [65]. Every frame produces one value. After all the frames have been evaluated, a vector containing the assessment of each of the frames is formed. This vector has a length equal to the number of frames in the video. Again, the minimum, maximum, and median values of this vector are obtained as given previously. Then, frames corresponding to those values are extracted and used for classification. If l number of frames needs to be chosen, then l number of minima. maxima and median values are found, and the frames corresponding to those values are chosen.
- Random: Frames are selected randomly from the video as the baseline [66]. Random selection is used for comparison with other frames' selection criteria.

All these criteria are used to select the frames and the classification performance based on training using frames chosen using these criteria are compared. The role of the number of frames retrieved from each video on the classification performance is also tested. Ten, twenty, and thirty frames are considered for this comparison. The proposed classification model is VGG16 as the backbone and is trained using these selected frames.

III. DEEP LEARNING MODEL

The overall framework of our proposed model is shown in Figure 2. It consists of three parts, i.e., feature extraction, attention, and classification layers. In order to reduce the time required to train the model, a pre-trained VGG16 model is used to extract features from the US images.

Multi-head self-attention is employed to re-calibrate and refine the extracted features. Then, the last global average pooling layer with a dense layer is used to provide the frame classification. The voting step is used to provide video classification and is utilized during the testing phase. Thus, the training phase provides frame classification, while during the testing phase, every frame in the video is classified, and a majority voting is used to declare the class of the video.

1) FEATURE EXTRACTION

Our proposed model begins with feature extraction using a pre-trained model. A pre-trained deep learning model is one trained with a large dataset, such as ImageNet, and is then used as a feature extractor with a new and smaller dataset, such as the LUS images dataset considered in this paper. Our proposed feature extractor is based on VGG16 model. Also, other pre-trained deep learning models such as VGG19 [67], DenseNet121, DenseNet169, DenseNet201 [68], MobileNet, MobileNetV2 [50], ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2 [69], NasNetLarge, NasNetMobile [70], and Xception [71] are also used to compare purposes. All these pre-trained models are primarily trained for the classification of more than one thousand different objects of classes using a big ImageNet dataset. In our work, early convolutional layers in the pre-trained model are frozen, and late layers are fine-tuned.

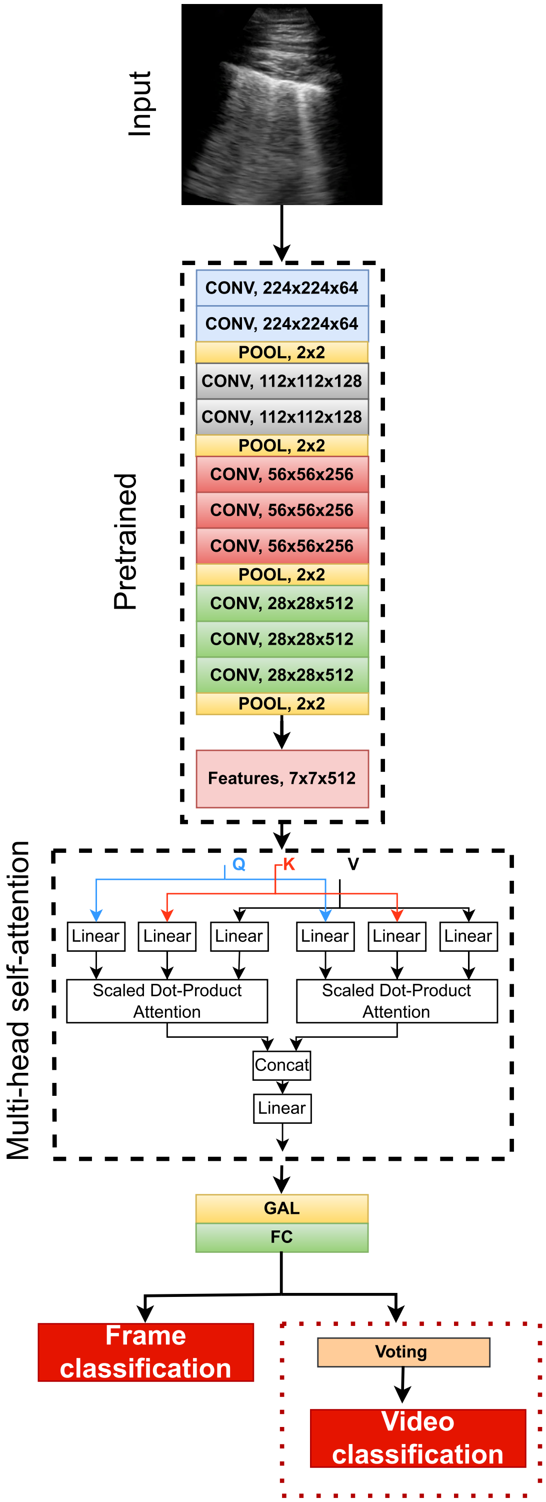

FIGURE 2. Overall diagram of our proposed model. Voting is employed for testing (video classification) only as indicated by the red dot rectangle.

All fully connected layers are completely modified to fit our problem at hand.

The size of features differs from one pre-trained model to another. They are determined by the location from which these features are taken in the respective pre-trained model since the model has a different number of trainable parameters in the fine-tuned layers. Figure 2 shows the various parts of the proposed model with the pre-trained model as the feature extractor. The extracted features are fed as input to the multi-head self-attention module for refining and recalibrating these features as shown in Figure 2.

2) MULTI-HEAD SELF-ATTENTION MODEL

It is implemented following the original paper [72]. It is called self-attention because query (Q), key (K), and value (V) are the same. Thus, Q, K, and V are the features that are provided by the pre-trained model. The first step of the multi-head self-attention model is to project Q, K, and V using a linear projection (i.e., dense layer without activation function) to provide transformed features Q', K', V'. The linear transformation brings down the length of the query, key, and value to d_k. In our experiments, d_k was set to 1024. Then, scaled dot-product attention (SDPA) is applied to the output of linear projections as shown below:

$$SDPA(Q', K', V') = softmax\left(\frac{Q'K'^T}{\sqrt{d_k}}\right)V', \tag{10}$$

where T denotes the transpose, and d_k is the length of the linearly transformed features. Thus, the SDPA may be considered as the scaling of the linearly transformed value V'. We also randomly drop out the SDPA by making the scaling zero (dropout layer). The combined operation of linear projections and SDPA is called as a single self-attention head in deep learning literature. To obtain multi-head self-attention units, N self-attention units, each with a different linear transformation of the same dimension, are obtained. Then, the output from N SDPA is concatenated and fed to the final linear projection layer to produce the multi-head self-attention output with the same size as the dimension of the input key. In our work, we have used 32 attention units ($N = 32$), 1024 neurons in the fully connected layer, and a 0.5 dropout ratio. The output from multi-head self-attention is pooled using a global average layer and is followed by a dense layer with a unit number equal to the number of classes, which is three in our case, to obtain the frame classification results as shown in Figure 2.

3) VOTING

Each video clip consists of a number of frames. Training model using videos for video classification purposes is arduous for many reasons. Firstly, the amount of available US video clips (183 in total) is insufficient to train a deep learning model. Secondly, as each video has a different frame rate, the number of frames varies from one video to another. The deep learning model, on the other hand, requires a constant number of frames as inputs. In [9], [34] the authors have tackled this challenge by taking a fixed amount of frames from each video. These frames are assumed to be representative of the video. However, they ended up training the deep learning model with a limited number of frames rather than the actual whole video. Finally, as the duration of the videos may not be the same, every video may not be efficiently represented, thus adding additional difficulty in training the model using video. In this paper, we bypass this issue by using the idea of voting to obtain the final prediction of video classification during the test phase.

Majority voting is utilized as extra post-processing in the test phase to offer video classification, as shown by a red hash rectangle in Figure 2. For each frame, our proposed model provides three probabilities of class association, that is, the probability that the frame belongs to the COVID-19 class, the probability that the frame belongs to the bacterial pneumonia class, and the probability that the frame belongs to the healthy class. If the video has M frames, then the probabilities of M frames belonging to COVID-19 is a vector with a length equal to M as $P_{COVID-19} = \{pc_1, pc_2, pc_3, \cdots, pc_M\}$. Similarly, bacterial pneumonia and healthy classes output probability ($P_{Pneumonia}$, and $P_{Healthy}$) are a vector with the same length M. Finally, the voting is defined by the equation

$$voting = MAX\left\{\overline{P_{COVID-19}}, \overline{P_{Pneumonia}}, \overline{P_{Healthy}}\right\}, \tag{11}$$

where MAX is the maximum, and $\overline{P_{(\cdot)}}$ represents average of the probability vector.

IV. EXPERIMENTS AND RESULTS

In this paper, US COVID-19 images datasets from [9] and [73], containing four classes, i.e., COVID-19, bacterial pneumonia, (non-COVID-19) viral pneumonia, and healthy lung are used. The dataset consists of 202 videos and 59 images recorded using either convex or linear ultrasound probes from 216 patients. The dataset has been collected from 41 distinct resources. Approximately half of the dataset is provided by Northumbria Data (Northumbria Healthcare NHS Foundation Trust serves many people in the North East of the United Kingdom) and Neuruppin Data (from Brandenburg Medical School Theodor Fontane in Neuruppin, Germany). Northumbria and Neuruppin data are employed GE Healthcare US device and scanned on patients following BLUE protocol [74]. Moreover, Northumbria utilized RT-PCR tests to validate or dismiss COVID-19 diagnoses and implemented standard care, such as thoracic X-ray and CT, to detect bacterial pneumonia. However, only 42% of the data contains information on age and gender, and symptom descriptions are available for only 30% of the data. In this work, we use only video data from convex probes. Since the data was gathered from different sources, it has various formats and illumination conditions, which increase the heterogeneity in the set. The dataset consists of 64, 49, 3, and 66 videos that belong to COVID-19, bacterial pneumonia, viral pneumonia, and healthy lung classes, respectively. The videos from viral pneumonia are excluded from this study due to their limited representation. Two medical experts, a pediatric physician with more than 10 years of clinical LUS experience and an academic US course instructor, have reviewed and confirmed the labeling of all videos in this dataset. The experts carefully annotated the videos with visible LUS patterns [9], [73]. Each video is pre-processed as given in [9] to remove the artifacts on the borders, such as texts or measure bars.

Before frame extraction, the dataset video clips are divided into five-fold cross-validation to train and evaluate our proposed model. Thus, the frame extracted from a chosen video appears either in the training or testing fold but not in both. This avoids the problem of correlation between frames that are extracted for the same video. Most of the work in the literature, barring very few papers such as [19] and [9], provide results where the training and the test data are correlated.

Our proposed model is trained using this dataset with pre-trained VGG16 as the backbone, as shown in Figure 2. Each experiment is repeated five times using the five-fold cross-validations and provides the mean over this five-fold. Dataset is resized using bilinear interpolation to fit the requirement of the inputs of the model. This is done to ensure that all the videos in the training and test set have the same size of frames. Data augmentation is carried out only on the training set. For augmentation, a horizontal flip, rotation with 10°, width and height shift with 0.1, shear with a range of 0.2, and zoom with a factor of 0.2 are applied. Adam optimizer is used with a learning rate of 0.0001 and 100 epochs with early stopping methods using the validation loss as a monitor with a patience value of 20. The learning rate is reduced by a factor of 0.001 when the learning stagnates with a patient of 20 too. Cross-entropy loss is used for the training process. The performance of our proposed deep learning framework for US image classifications is evaluated using accuracy, sensitivity, precision, F1_score, and area under the curve (AUC).

Two types of video classification, namely frame-based and video-based, were achieved by voting techniques in the test phase. For selected frames video, the following steps were undertaken: (a) The test video clip was run through frame selection criteria to extract M frames, (b) the extracted frames were fed into the trained model to provide the predicted label probability for each of the selected frames. Thus the M predicted class probabilities for every frame were collated as follows: $P_{COVID-19} = \{pc_1, pc_2, pc_3, \cdots, pc_M\}$, $P_{Pneumonia} = \{pp_1, pp_2, pp_3, \cdots, pp_M\}$, and $P_{Healthy} = \{ph_1, ph_2, ph_3, \cdots, ph_M\}$, and (c) the voting technique was implemented using these class probabilities for each selected frame as input to predict the video clip's final label. Voting was achieved using the equation 11.

The video classification based on all the frames was achieved as follows: (a) All frames in the test video were extracted as no frame criterion was used in the testing process. The frame duration and the number of frames may be different from one video clip to another depending on the video duration and frame rate (b) the trained model used these extracted frames as input to provide the predicted class probability vectors $P_{COVID-19} = \{pc_1, pc_2, pc_3, \cdots, pc_N\}$, $P_{Pneumonia} = \{pp_1, pp_2, pp_3, \cdots, pp_N\}$, and $P_{Healthy} = \{ph_1, ph_2, ph_3, \cdots, ph_N\}$ where N is a number of the frame in the video, and (c) similarly, voting is done based on equation 11 to produce the video clip's final label. Note that in this case, the number of frames is different, and therefore, each video has been averaged differently, unlike the selected frames video classification where every video has a fixed number of selected frames.

The classification results are shown in Table 1 with a maximum of 10 frames extracted from each video using various frame selection criteria proposed in this paper. The table displays a heat map where the assigned color depends on the value of the cell. Color maps are used to emphasize cells with high values in green, moderate values in yellow, and low values in red. Based on the frame selection criterion, the table is divided into four categories (position-based, norm-based, entropy and fractal dimension-based, similarity-based, and other metrics). Table 1 shows the performance of the frame and video classifications.

For the frame classification, a maximum SVD with sensitivity, specificity, precision, F1_score, and accuracy yields the best result of 0.8553, 0.9254, 0.8502, 0.8496, and 0.8501, respectively. Minimum SSIM, maximum entropy, maximum Frobenius, median Petrosian, median Katz, median Higuchi, and maximum detrended also provide a close result similar to maximum SVD as demonstrated by the heatmap (green color) in Table 1. Constant frame selection criteria produce the lowest outcomes, even worse than random selection. The maximum value in the norm-based category (SVD, Frobenius) delivers the best performance since it retrieves the frames in the video clips with the most energy. While the performance provided by the median value in entropy and fractal dimension-based category (Entropy, Petrosian, Katz, Higuchi, and Detrended) outperforms or comes close to their maximum value. Choosing minimum values in similarity-based (SSIM, PSNR) criteria yields the best results, as expected because it overcomes the correlation problem between the video's neighbor frames. The application of quality criteria in frame selection gives acceptable outcomes. In the future, it will be necessary to apply various quality evaluations that may capture relevant characteristics in the frames. Furthermore, video classification (both types) operates similarly to frame classification since it is dependent on the predicted label of the frame, as demonstrated in Table 1. Minimum SSIM achieves the best result for the video classification based on the selected frame while the best results of video classification using the whole video are achieved by the maximum SVD.

To determine if selection criteria would provide better or worse results than random selection, F1_score is used since F1_score is a good choice for evaluating the imbalance data as it takes into account both sensitivity and precision measures

TABLE 1. The results of select frames with maximum of 10 frames using different metrics and our model with VGG16 as the backbone.

Metrics		Frame classification					Video classification based on selected frames					Video classification based on whole video				
		Sensitivity	Specificity	Precision	F1_score	Accuracy	Sensitivity	Specificity	Precision	F1_score	Accuracy	Sensitivity	Specificity	Precision	F1_score	Accuracy
Position-based																
constant		0.7859	0.8909	0.7856	0.7714	0.7895	0.7999	0.9039	0.8222	0.7925	0.8112	0.818	0.9056	0.8215	0.8064	0.809
non-adjacent		0.8263	0.9089	0.8243	0.8148	0.8171	0.8334	0.9129	0.8375	0.8222	0.8251	0.8382	0.9156	0.8401	0.8278	0.8305
Norm-based																
SVD	min	0.796	0.8942	0.8013	0.7883	0.7891	0.8172	0.9039	0.828	0.8089	0.8093	0.8026	0.8968	0.8089	0.792	0.7925
	med	0.801	0.8985	0.8118	0.7967	0.7982	0.8343	0.9151	0.8473	0.8293	0.8311	0.8295	0.9123	0.8447	0.8263	0.8257
	max	0.8553	0.9254	0.8502	0.8496	0.8501	0.8634	0.9301	0.8577	0.8564	0.8583	0.8832	0.9401	0.8853	0.8802	0.88
Frobenius	min	0.7879	0.8899	0.7964	0.7804	0.7813	0.7928	0.893	0.8048	0.7844	0.7872	0.8165	0.9041	0.8142	0.8082	0.8087
	med	0.8266	0.9109	0.8283	0.8212	0.822	0.8402	0.9183	0.8401	0.8344	0.8362	0.8469	0.9209	0.8509	0.8412	0.8417
	max	0.8442	0.9188	0.8363	0.8354	0.8364	0.8385	0.9162	0.8304	0.8292	0.8303	0.8682	0.9319	0.865	0.8615	0.8634
Entropy and fractal dimension-based																
Entropy	min	0.7968	0.8957	0.7976	0.7879	0.7903	0.8201	0.908	0.8232	0.8116	0.8147	0.8391	0.9175	0.8428	0.8356	0.8362
	med	0.8309	0.9119	0.8339	0.8264	0.8248	0.8492	0.9214	0.8506	0.8427	0.8419	0.8543	0.9238	0.8541	0.8473	0.847
	max	0.8308	0.9138	0.8315	0.826	0.8271	0.8489	0.9231	0.8531	0.8462	0.8473	0.8282	0.91	0.8316	0.8203	0.8204
Petrosian	min	0.7958	0.8942	0.7922	0.7864	0.7871	0.797	0.8948	0.7947	0.7852	0.7875	0.8444	0.9182	0.8442	0.8378	0.8365
	med	0.8479	0.9207	0.8465	0.8411	0.8414	0.8532	0.9239	0.8544	0.8457	0.8474	0.859	0.9265	0.8645	0.8526	0.853
	max	0.8141	0.9046	0.8152	0.8083	0.8094	0.8267	0.9105	0.8297	0.819	0.8204	0.8418	0.9186	0.8429	0.8343	0.8363
Katz	min	0.8131	0.9049	0.8183	0.8089	0.8115	0.8163	0.9063	0.8221	0.8102	0.823	0.8541	0.925	0.8646	0.8516	0.8526
	med	0.8377	0.9156	0.8363	0.8303	0.8305	0.8574	0.9238	0.8528	0.8484	0.8474	0.8338	0.9127	0.8346	0.8268	0.8257
	max	0.8254	0.9116	0.8332	0.8217	0.8231	0.8368	0.9182	0.8534	0.834	0.836	0.859	0.9278	0.8572	0.8503	0.8524
Higuchi	min	0.82	0.9089	0.8258	0.8169	0.8193	0.8431	0.9199	0.8488	0.8401	0.8417	0.8398	0.9175	0.8429	0.8355	0.8363
	med	0.8352	0.9145	0.834	0.8297	0.8299	0.8469	0.9206	0.8494	0.8427	0.842	0.8511	0.9211	0.8487	0.8428	0.842
	max	0.8328	0.9132	0.8362	0.8279	0.8278	0.8539	0.9231	0.8577	0.8472	0.8474	0.8511	0.9209	0.8431	0.8403	0.8414
Detrended	min	0.8228	0.9073	0.8262	0.8151	0.8168	0.8432	0.9178	0.8444	0.8335	0.8365	0.8448	0.9179	0.8498	0.8362	0.8365
	med	0.832	0.9129	0.8263	0.8238	0.8243	0.8326	0.9133	0.8315	0.8249	0.8255	0.8444	0.9187	0.8436	0.8369	0.8366
	max	0.8337	0.9145	0.8305	0.8272	0.8291	0.8354	0.9153	0.8378	0.8285	0.8308	0.8543	0.9242	0.8521	0.8467	0.847
Similarity-based																
SSIM	min	0.8338	0.9139	0.8329	0.8268	0.8271	0.8708	0.9325	0.8688	0.8636	0.8637	0.8516	0.9235	0.8535	0.8459	0.847
	med	0.807	0.904	0.8166	0.8023	0.8075	0.8061	0.9047	0.8252	0.8025	0.8093	0.8261	0.914	0.8475	0.8196	0.8236
	max	0.8053	0.8994	0.8036	0.7939	0.8006	0.822	0.9099	0.83	0.813	0.8205	0.8426	0.9211	0.8484	0.8382	0.8422
PSNR	min	0.8278	0.9115	0.8266	0.8253	0.8249	0.8619	0.9283	0.8618	0.8578	0.858	0.8505	0.9232	0.8484	0.8451	0.847
	med	0.8086	0.9018	0.8112	0.8058	0.8054	0.8233	0.9088	0.8288	0.8202	0.8201	0.8355	0.9151	0.8403	0.8304	0.8311
	max	0.8166	0.9064	0.8185	0.8101	0.8109	0.8366	0.9161	0.8414	0.831	0.8309	0.8568	0.9264	0.8595	0.853	0.8529
Other metrics																
Quality	min	0.8149	0.9045	0.8234	0.8095	0.81	0.8394	0.918	0.8551	0.8365	0.8371	0.8326	0.912	0.845	0.8282	0.8258
	med	0.8325	0.9132	0.8295	0.8232	0.8254	0.8437	0.9189	0.8417	0.8349	0.8363	0.8437	0.9186	0.8432	0.8358	0.8365
	max	0.8223	0.9072	0.8237	0.8129	0.8131	0.8393	0.9166	0.8386	0.8302	0.8309	0.8271	0.9108	0.8292	0.8192	0.8198
Random		0.8037	0.8983	0.801	0.7954	0.7952	0.8327	0.913	0.8302	0.8244	0.8252	0.8429	0.9189	0.8408	0.8349	0.8362

as

$$F1\_score = \frac{2 \times sensitivity \times precision}{sensitivity + precision}$$

Figure 3 (a) shows the F1_score of all methods (as a bar chart). The random selection is highlighted in red. It can be seen from the figure that random selection outperforms constant, maximum SSIM, minimum SVD, minimum entropy, minimum Petrosian, and minimum Frobenius as expected. Because the minimum SVD and minimum Frobenius retrieve frames with the lowest energy, minimum entropy and minimum Petrosian yield frames with less information, and maximum SSIM return frames that are similar to each other, which increase the correlation issue, these criteria provide worse results than the random method. Further random selection outperforms the constant method which is the most commonly used frame selection criterion in almost all state-of-the-art methods in the literature. Random selection delivers better results than the constant and non-adjacent choice for frames in both types of video classifications, as shown in Figure 3 (b) and Figure 3 (c).

The receiver operating characteristic curve (ROC) and area under the curve (AUC) for each class (COVID-19 and bacterial pneumonia, and healthy) based on the various selection methods are calculated. The best result is provided by the SVD_max with mean AUC 0.94, 0.97, and 0.91 for COVID-19, bacterial pneumonia, and healthy classes, respectively. Furthermore, random selection performs better than constant selection for all classes based on the AUC scores. The best overall average AUC score for COVID-19 is again provided by SVD_max with 0.938, and the worst result is 0.876 provided by the constant method. This further confirms that the choice of frame selection in the state-of-the-art methods in the literature is not optimal. For the bacterial pneumonia class, SVD_max has the best result with 0.968. While Detrended_max achieves 0.934. The overall standard deviations for COVID-19 and bacterial pneumonia are larger than the standard deviation of the healthy class due to the high heterogeneity in the videos of the bacterial pneumonia class when compared to the healthy class. We repeat the previous experiments with an increased number of frames to 20 and 30. The results further affirm that the proposed choice of frame selection is better than the constant method, which is the choice adopted in the state-of-the-art methods in the literature.

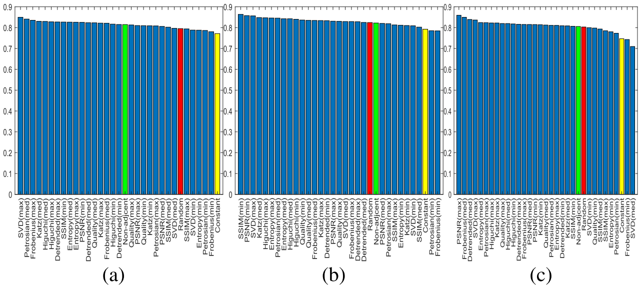

FIGURE 3. F1_score for different criteria. (a) frame classification, (b) video classification based on the selected 10 frames, and (c) video classification based on the whole frame in the video. F1_score of random selection, constant selection, and non-adjacent demonstrate in red, yellow, and green, respectively for clarity and comparison.

V. COMPARATIVE EVALUATION WITH THE CURRENT STATE-OF-THE-ART

In order to study the impact of the backbone model on the classification results, we implemented our proposed model

TABLE 2. Results of our proposed model using various backbone pretrained models. Frames were selected using SVD_{max} with a maximum of 10 frames per video. ∗ indicates the reimplemented model proposed in previous studies.

Models		COVID-19				Pneumonia				Healthy				
		Sensitivity	Specificity	Precision	F1-Score	Sensitivity	Specificity	Precision	F1-Score	Sensitivity	Specificity	Precision	F1-Score	Accuracy
VGG16	Mean	0.8206	0.9334	0.8800	0.8450	**0.8909**	0.9442	0.8567	**0.8723**	0.8797	**0.9139**	**0.8649**	**0.8672**	**0.8628**
	Std	0.1147	0.0494	0.0868	0.0775	0.0528	0.0273	0.0514	0.0385	0.0884	0.0667	0.0885	0.0473	0.0455
VGG19	Mean	0.7302	0.9345	0.8619	0.7900	0.8403	0.9316	0.8144	0.8248	0.8847	0.8533	0.7873	0.8305	0.8162
	Std	0.1129	0.0357	0.0902	0.1035	0.1226	0.0267	0.0825	0.0904	0.0466	0.0953	0.1041	0.0665	0.0805
DenseNet121	Mean	0.8050	0.9268	0.8638	0.8322	0.7853	0.9296	0.8428	0.7948	0.8642	0.8732	0.7976	0.8228	0.8215
	Std	0.0787	0.0341	0.0714	0.0674	0.1259	0.0843	0.1532	0.0473	0.1402	0.0432	0.0288	0.0550	0.0494
DenseNet169	Mean	0.7455	0.8902	0.8169	0.7704	0.8352	0.9534	0.8798	0.8433	0.8252	0.8440	0.7771	0.7891	0.7989
	Std	0.1445	0.0541	0.0370	0.0737	0.1156	0.0549	0.1325	0.0416	0.1105	0.1412	0.1112	0.0386	0.0397
DenseNet201	Mean	0.7736	0.9210	0.8528	0.8092	0.8422	0.9271	0.8338	0.8195	**0.9022**	0.9014	0.8464	0.8651	0.8315
	Std	0.0497	0.0398	0.0768	0.0421	0.1323	0.0809	0.1517	0.0606	0.1136	0.0686	0.0962	0.0525	0.0475
InceptionResNetV2	Mean	0.7994	0.9414	**0.8941**	0.8409	0.7272	0.9514	0.8514	0.7719	0.8730	0.8108	0.7246	0.7907	0.8073
	Std	0.0668	0.0363	0.0611	0.0305	0.1814	0.0290	0.0891	0.1087	0.0617	0.0596	0.0425	0.0388	0.0399
MobileNet	Mean	0.6215	0.9354	0.8618	0.6829	0.7217	0.9055	0.6164	0.6499	0.8395	0.7394	0.7271	0.7498	0.7186
	Std	0.2961	0.0442	0.0310	0.2275	0.4156	0.1064	0.3765	0.3689	0.1390	0.3459	0.2440	0.1597	0.1966
MobileNetV2	Mean	0.7939	0.9239	0.8643	0.8248	0.7511	0.9332	0.8314	0.7738	0.8173	0.8226	0.7411	0.7535	0.7858
	Std	0.0464	0.0513	0.0775	0.0339	0.1016	0.0879	0.1714	0.0850	0.1723	0.0958	0.1318	0.0492	0.0414
NASNetLarge	Mean	0.6679	0.8770	0.7669	0.7113	0.8148	0.8729	0.7043	0.7550	0.7405	0.8508	0.7341	0.7361	0.7362
	Std	0.0909	0.0353	0.0373	0.0592	0.0716	0.0391	0.0510	0.0566	0.0882	0.0280	0.0312	0.0565	0.0341
NASNetMobile	Mean	0.7751	0.8943	0.8138	0.7919	0.8836	0.9224	0.8122	0.8433	0.7698	0.8841	0.7922	0.7732	0.8021
	Std	0.0886	0.0345	0.0565	0.0586	0.0346	0.0430	0.0909	0.0407	0.1137	0.0509	0.0661	0.0405	0.0291
ResNet50	Mean	0.6222	0.8528	0.7259	0.6659	0.6044	0.9281	0.7837	0.6668	0.8112	0.7307	0.6392	0.7115	0.6867
	Std	0.0947	0.0561	0.0356	0.0498	0.1377	0.0710	0.1637	0.1054	0.1099	0.1214	0.0705	0.0646	0.0590
ResNet50V2	Mean	0.8116	0.9166	0.8550	0.8291	0.8024	0.9502	0.8570	0.8265	0.8536	0.8609	0.7796	0.8105	0.8216
	Std	0.0907	0.0294	0.0464	0.0392	0.0570	0.0255	0.0567	0.0317	0.0671	0.0622	0.0782	0.0370	0.0142
ResNet101	Mean	0.5858	0.8688	0.7241	0.6412	0.6865	0.8372	0.6554	0.6456	0.6658	0.7569	0.6341	0.6244	0.6465
	Std	0.1730	0.0345	0.0647	0.1254	0.1291	0.1227	0.2049	0.0679	0.2571	0.1569	0.1429	0.1596	0.1039
ResNet101V2	Mean	0.7814	0.9107	0.8468	0.8062	0.8432	0.9296	0.8157	0.8278	0.8088	0.8675	0.7745	0.7882	0.8065
	Std	0.0821	0.0494	0.0852	0.0216	0.0656	0.0280	0.0628	0.0506	0.0436	0.0490	0.0744	0.0315	0.0179
ResNet152	Mean	0.6677	0.8849	0.7858	0.7207	0.7285	0.9047	0.7347	0.7256	0.7675	0.7727	0.6482	0.6996	0.7132
	Std	0.0728	0.0707	0.1149	0.0866	0.1382	0.0279	0.0679	0.0807	0.1477	0.0345	0.0536	0.0932	0.0730
ResNet152V2	Mean	0.7122	0.9275	0.8632	0.7779	0.8586	0.9110	0.7789	0.8153	0.8453	0.8609	0.7785	0.8086	0.8012
	Std	0.0487	0.0478	0.0774	0.0337	0.0696	0.0229	0.0587	0.0504	0.0417	0.0566	0.0770	0.0453	0.0248
Xception	Mean	0.8261	0.8989	0.8315	0.8247	0.8238	0.9159	0.7848	0.7941	0.7727	0.8886	0.8117	0.7852	0.8033
	Std	0.0906	0.0474	0.0693	0.0471	0.1349	0.0404	0.0828	0.0656	0.0897	0.0910	0.0947	0.0533	0.0501
Covid_cap* [75]	Mean	0.6599	**0.9501**	0.8833	0.7493	0.8079	0.8029	0.6268	0.6793	0.7425	0.8492	0.7443	0.7306	0.7261
	Std	0.1308	0.0688	0.1604	0.1245	0.2343	0.1347	0.1322	0.1317	0.2064	0.0988	0.1299	0.1383	0.1105
Pocovid_model* [33]	Mean	0.7767	0.9310	0.8626	0.8147	0.7886	0.9617	0.8795	0.8283	0.8552	0.8091	0.7297	0.7861	0.8084
	Std	0.1430	0.0247	0.0665	0.1075	0.1441	0.0287	0.1099	0.1163	0.0500	0.1128	0.0937	0.0734	0.0897
Minicovid* [19]	Mean	0.8628	0.9102	0.8593	**0.8584**	0.8412	0.9565	**0.8879**	0.8572	0.8475	0.9038	0.8370	0.8354	0.8505
	Std	0.0524	0.0522	0.0603	0.0222	0.0904	0.0356	0.0868	0.0336	0.1054	0.0487	0.0653	0.0363	0.0208
Nasnet model* [19]	Mean	0.5408	0.8088	0.6714	0.5649	0.7559	0.7646	0.6000	0.6435	0.4873	0.7947	0.6515	0.4665	0.5781
	Std	0.2355	0.1805	0.1813	0.1844	0.1108	0.1739	0.2221	0.1242	0.3521	0.2206	0.1347	0.2254	0.1219
Resnet50* [19]	Mean	0.8700	0.8909	0.8273	0.8473	0.8056	0.9329	0.8144	0.8053	0.8028	0.9124	0.8509	0.8225	0.8275
	Std	0.0484	0.0475	0.0704	0.0532	0.0916	0.0251	0.0659	0.0467	0.0917	0.0619	0.0827	0.0654	0.0489
Mobilenet* [19]	Mean	**0.8746**	0.4817	0.5408	0.6377	0.3022	**0.9872**	0.7514	0.3536	0.5431	0.8971	0.7788	0.6136	0.5947
	Std	0.1669	0.2640	0.1636	0.0592	0.3644	0.0176	0.4245	0.3940	0.1781	0.1149	0.1476	0.0707	0.1052

with several backbone models like VGG16, VGG19 [67], DenseNet121, DenseNet169, DenseNet201 [68], MobileNet, MobileNetV2 [50], ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2 [69], NasNet-Large, NasNetMobile [70], and Xception [71]. For a fair comparison with the state-of-the-art models, we have implemented the models mentioned in [19] using our proposed training procedures. Identical training and test datasets are used throughout the comparative studies. In [19], a lightweight model called Minicovid was proposed and compared to other five deep learning models, namely Covid_cap, Nasnet model, Pocovid model, Resnet50, and Mobilenet. Focal loss was used during the training process. In our work, the previously mentioned models were implemented, and instead of focal loss, we used cross-entropy loss, which was also used to obtain the model from pretrained models. However, instead of following the training procedure in [19], the training procedure mentioned in Section IV was adopted because the training procedure given in [19] was biased to detect COVID-19 and the healthy class was missed.

Minicovid is a scaled-down version of the Mobilenet model [50]. Minicovid employs depth-wise separable convolution and point-wise convolutions instead of traditional convolution layers. This modified Mobilenet model is followed by an output block instead of the original fully connected layers. The output block consists of a fully connected layer with 64 units, a ReLU activation function, a dropout of 0.5, batch normalization, a fully connected layer with 3 units (equal number of classes), and a softmax activation function. Covid_cap is proposed in [75] to detect COVID-19 in the chest X-ray (CXR). This model is comprised of four convolution layers followed by three capsule layers. The Pocovid model is proposed in [33]. It uses VGG16 as the backbone is followed by the output block. While, Nasnet, Resnet50, Mobilenet are the modifications of NasNetMobile [70], ResNet50 [69], and MobileNetV2 [50], respectively. The modifications happened by replacing the fully connected layers of these models (Nasnet, Resnet50, and Mobilnet) with the output block.

Table 2 shows the mean and standard deviation (std) results for frames extracted based on the SVD_max with a maximum of 10 frames per video using different pre-trained backbone models and the models that are proposed in [19], [75] and [33]. Our proposed model with VGG16 as the backbone has the best classification accuracy of 0.8628. It also achieves balanced sensitivity for the COVID-19, bacterial pneumonia, and healthy classes with 0.8206 ± 0.1147,

TABLE 3. Resuts of video classification using majority voting and svdmax methods. ∗ indicates the reimplemented model proposed in previous studies with our training algorithm.

Models		COVID-19				Pneumonia				Healthy				
		Sensitivity	Specificity	Precision	F1-Score	Sensitivity	Specificity	Precision	F1-Score	Sensitivity	Specificity	Precision	F1-Score	Accuracy
VGG16	Mean	0.8454	0.9300	0.8777	0.8577	**0.9396**	0.9476	0.8727	**0.9040**	0.8777	**0.9473**	**0.9157**	**0.8903**	**0.8839**
	Std	0.0977	0.0512	0.0916	0.0702	0.0556	0.0330	0.0590	0.0484	0.0843	0.0575	0.0842	0.0178	0.0285
VGG19	Mean	0.7140	0.9292	0.8435	0.7715	0.8486	0.9397	0.8357	0.8367	0.9099	0.8565	0.8093	0.8506	0.8208
	Std	0.1562	0.0412	0.1176	0.1404	0.1574	0.0344	0.1042	0.1057	0.0619	0.1317	0.1518	0.0925	0.1052
DenseNet121	Mean	0.8709	0.9379	0.8910	**0.8795**	0.8166	0.9487	0.8829	0.8298	0.8751	0.8995	0.8302	0.8451	0.8581
	Std	0.0510	0.0394	0.0744	0.0503	0.1663	0.0621	0.1255	0.0741	0.1457	0.0443	0.0586	0.0688	0.0417
DenseNet169	Mean	0.7488	0.9035	0.8295	0.7802	0.9036	0.9563	0.8967	0.8855	0.8352	0.8633	0.8103	0.8099	0.8217
	Std	0.1445	0.0365	0.0206	0.0891	0.1384	0.0610	0.1366	0.0711	0.1206	0.1504	0.1269	0.0588	0.0606
DenseNet201	Mean	0.7876	0.9300	0.8701	0.8243	0.8451	0.9203	0.8237	0.8146	0.9108	0.9142	0.8692	0.8804	0.8403
	Std	0.0797	0.0502	0.0906	0.0650	0.1274	0.0880	0.1629	0.0587	0.1170	0.0791	0.1232	0.0735	0.0584
InceptionResNetV2	Mean	0.8097	**0.9648**	**0.9302**	0.8652	0.7579	0.9640	0.8929	0.8061	0.9097	0.8145	0.7441	0.8162	0.8337
	Std	0.0356	0.0377	0.0722	0.0484	0.1894	0.0358	0.1214	0.1171	0.0774	0.0843	0.0754	0.0577	0.0590
MobileNet	Mean	0.6106	0.9379	0.8831	0.6661	0.7232	0.9149	0.6237	0.6571	0.8658	0.7351	0.7377	0.7676	0.7236
	Std	0.3144	0.0499	0.0708	0.2684	0.4194	0.0906	0.3720	0.3728	0.1360	0.3795	0.2551	0.1814	0.2145
MobileNetV2	Mean	0.8263	0.9292	0.8746	0.8465	0.7574	0.9413	0.8645	0.7872	0.8218	0.8322	0.7587	0.7629	0.8017
	Std	0.0738	0.0412	0.0684	0.0407	0.1255	0.0931	0.1905	0.0914	0.1828	0.1027	0.1371	0.0580	0.0384
NASNetLarge	Mean	0.7497	0.8853	0.7966	0.7718	0.9014	0.8942	0.7556	0.8207	0.7542	0.9064	0.8142	0.7816	0.7912
	Std	0.0637	0.0271	0.0412	0.0495	0.0959	0.0225	0.0516	0.0612	0.0964	0.0158	0.0500	0.0705	0.0343
NASNetMobile	Mean	0.8071	0.9213	0.8553	0.8293	**0.9396**	0.9323	0.8383	0.8821	0.8185	0.9177	0.8518	0.8281	0.8465
	Std	0.0921	0.0188	0.0538	0.0682	0.0556	0.0414	0.1005	0.0499	0.0782	0.0554	0.0954	0.0203	0.0179
ResNet50	Mean	0.6560	0.8505	0.7265	0.6855	0.6252	0.9336	0.8202	0.6850	0.8016	0.7526	0.6593	0.7189	0.7022
	Std	0.1124	0.0519	0.0702	0.0756	0.1606	0.0804	0.1833	0.1034	0.1079	0.1293	0.0843	0.0693	0.0541
ResNet50V2	Mean	0.8550	0.9209	0.8650	0.8570	0.8342	0.9699	0.9068	0.8676	**0.9110**	0.9056	0.8494	0.8745	0.8674
	Std	0.0992	0.0202	0.0418	0.0514	0.0923	0.0172	0.0541	0.0689	0.0775	0.0553	0.0747	0.0315	0.0309
ResNet101	Mean	0.5865	0.8769	0.7432	0.6502	0.7056	0.8330	0.6509	0.6523	0.6601	0.7593	0.6410	0.6200	0.6508
	Std	0.1677	0.0574	0.1160	0.1372	0.1663	0.1247	0.2094	0.0878	0.2737	0.1579	0.1501	0.1615	0.1096
ResNet101V2	Mean	0.8063	0.9212	0.8690	0.8260	0.8360	0.9407	0.8386	0.8354	0.8482	0.8762	0.7986	0.8157	0.8262
	Std	0.1369	0.0466	0.0750	0.0566	0.0545	0.0183	0.0447	0.0212	0.0715	0.0744	0.1034	0.0371	0.0276
ResNet152	Mean	0.6756	0.8864	0.7980	0.7308	0.7273	0.8960	0.7223	0.7191	0.7731	0.7852	0.6614	0.7096	0.7173
	Std	0.0939	0.0846	0.1503	0.1163	0.1420	0.0450	0.1046	0.1018	0.1645	0.0283	0.0494	0.1032	0.0782
ResNet152V2	Mean	0.7210	0.9208	0.8583	0.7798	0.8520	0.8976	0.7540	0.7952	0.8315	0.8775	0.8092	0.8136	0.7976
	Std	0.0564	0.0571	0.0856	0.0265	0.1045	0.0443	0.0962	0.0772	0.0604	0.0816	0.1264	0.0527	0.0230
Xception	Mean	0.8329	0.9114	0.8506	0.8371	0.8768	0.9252	0.8066	0.8346	0.7875	0.8987	0.8394	0.8071	0.8259
	Std	0.1090	0.0565	0.0825	0.0696	0.1643	0.0258	0.0695	0.1014	0.0869	0.1127	0.1230	0.0834	0.0832
Covid_cap* [75]	Mean	0.6760	**0.9466**	0.8800	0.7568	0.8196	0.8074	0.6267	0.6800	0.7527	0.8664	0.7811	0.7519	0.7386
	Std	0.1438	0.0731	0.1643	0.1304	0.2949	0.1334	0.1362	0.1828	0.2022	0.1167	0.1631	0.1407	0.1216
Pocovid_model* [33]	Mean	0.7619	0.9292	0.8554	0.8022	0.7901	0.9545	0.8651	0.8219	0.8469	0.8071	0.7271	0.7802	0.8005
	Std	0.1599	0.0258	0.0736	0.1209	0.1175	0.0308	0.1038	0.0890	0.0493	0.1150	0.0984	0.0700	0.0885
Minicovid* [19]	Mean	0.8654	0.9122	0.8648	0.8611	0.8196	0.9617	**0.9088**	0.8522	0.8634	0.8995	0.8299	0.8414	0.8529
	Std	0.0653	0.0631	0.0745	0.0283	0.1150	0.0455	0.0966	0.0480	0.0920	0.0443	0.0598	0.0334	0.0196
Nasnet model* [19]	Mean	0.5633	0.8395	0.7034	0.5896	0.8255	0.7842	0.6553	0.6929	0.5161	0.7982	0.7362	0.4981	0.6136
	Std	0.2825	0.1728	0.1908	0.2370	0.1699	0.1778	0.2394	0.1232	0.3419	0.2370	0.2175	0.2104	0.1219
Resnet50* [19]	Mean	0.8642	0.8955	0.8334	0.8476	0.7910	0.9331	0.8106	0.7976	0.8183	0.9090	0.8505	0.8307	0.8287
	Std	0.0483	0.0474	0.0657	0.0485	0.0788	0.0301	0.0928	0.0669	0.1043	0.0680	0.1044	0.0847	0.0556
Mobilenet* [19]	Mean	**0.8829**	0.4869	0.5460	0.6458	0.3068	**0.9857**	0.5467	0.3476	0.5679	0.9144	0.8143	0.6460	0.6083
	Std	0.1474	0.2598	0.1710	0.0613	0.3865	0.0196	0.5025	0.4088	0.1721	0.1009	0.1526	0.0830	0.1098

0.8909 ± 0.0528, and 0.8797 ± 0.0884, respectively. Further, VGG16 also returns a small standard deviation in comparison to other models, as demonstrated in Table 2. Also, the best second results are provided by our proposed framework with DenseNet201 model as the backbone. It achieves an accuracy of 0.8315 ± 0.0475. The models proposed in [19], [75], and [33] deliver classification results with a large standard deviation. Further, these models are also biased. For instance, Covid_cap model achieves a sensitivity of 0.6599 ± 0.1308, 0.8075 ± 0.2343, and 0.7425 ± 0.2064 for COVID-19, bacterial pneumonia, and healthy classes, respectively.

To provide the video classification, all frames in the test videos are selected and fed to the trained model to give them the probability related to the class association. Then, the majority voting method is employed to yield the video label as given in the equation 11. our proposed model with VGG16 provides the best accuracy of 0.8839 as shown in Table 3. It also achieves a balanced sensitivity for COVID-19, bacterial pneumonia, and healthy classes and a small standard deviation with 0.8454 ± 0.0977, 0.9396 ± 0.0556, and 0.8777 ± 0.0843, respectively while the model using ResNet50V2 as the backbone gives the best result with an accuracy of 0.8674 ± 0.0309.

The dataset used in this work lacks certain information. It does not contain patient-specific information such as age and gender, disease-specific information such as symptoms and severity, or instrument-specific information such as imaging frequency and depth [9], [73]. It is unclear if any of this information would have helped in improving the classification. Additionally, lung surface changes may not be noticeable in early-stage COVID-19 patients, and therefore, detection would be very difficult. According to [9], heterogeneity in the dataset could induce bias in the classification; however, this heterogeneity is useful in developing models that can be generalized. Furthermore, LUS imaging patterns may not be disease specific [9], [73], [76], so also the patterns in computed tomography (CT) and chest radiograph (CRX) [77], [78]. Therefore, the results in this paper should not be considered for diagnostic purposes directly and should be used in conjunction with a physician's diagnosis. Given the fact that there are currently no publicly available datasets that are homogeneous, the current work with the available dataset does provide directions for the development of models for COVID-19 classification. Thus, the results provided in this paper can help in developing models for classifying severity in the LUS when homogeneous data becomes publicly available.

All the experiments show that our proposed method outperforms the models proposed in [19], [75], and [33]. Also, our experiments confirm that the frame selection and training approach proposed in this paper, deliver improved classification results when compared to those presented in [19] with frames selected with a fixed frame separation.

VI. CONCLUSION

In this paper, different frame selection criteria for video classification of US lung images using deep learning models were presented. A deep learning model comprised of feature extraction using pre-trained models such as VGG16, attention using multi-head self-attention model, and a global average layer and a dense layer for final classification with cross-entropy loss for learning was proposed.

Our proposed model consists of voting to deliver the video classification using majority voting. The proposed frame selection with our deep learning model outperformed the state-of-the-art models such as the Minicovid and state-of-the-art frame selections, namely the constant frame rate and the non-adjacent methods. Further, it is also shown that even random choice of frame outperformed the frame selection procedure used in state-of-the-art US lung image classification techniques. Frame selection using SVD_max, with VGG16 as the deep learning backbone model achieved the best results with an accuracy of 0.8628, and 0.8839 for frames and video classifications of healthy, bacterial pneumonia, and COVID-19 LUS images, respectively. Unlike the state-of-the-art approaches, the proposed method also obtained a balanced sensitivity for COVID-19, bacterial pneumonia, and healthy classes with a small standard deviation.

Data augmentation was used to adjust the class size of the underrepresented class. Instead of data augmentation, approaches such as one-versus-one and one-versus-all methods for multi-class imbalanced training [79] may be considered. An automatic way of selecting the number of frames from a given video may be developed instead of the proposed ad-hoc fixed number of frames from a video for training and testing purposes. A different approach for automatically selecting the number of frames per video during training is also considered, which may culminate in a new training procedure for deep learning. The choice of different loss functions may also be an option that can be considered as part of future work.

Despite the drawbacks related to the dataset presented in the last section, our proposed frame selection criteria and model can still be applied for training and testing on other datasets, such as scoring of COVID-19 severity or artifact detection datasets such as [41] and [76], if and when they are made publicly available. The frame selection method can also be used for the classification of videos in other areas of deep learning, such as video-based drone classification. The classification model trained using the given datasets may be used as a pre-trained model when homogeneous datasets become available.

REFERENCES

[1] *Centers for Disease Control and Prevention (CDC)*. Accessed: Jul. 11, 2022. [Online]. Available: https://coronavirus.jhu.edu/map.html

[2] T. Chen, D. I. Wu, H. Chen, W. Yan, D. Yang, G. Chen, K. Ma, D. Xu, H. Yu, H. Wang, and T. Wang, "Clinical characteristics of 113 deceased patients with coronavirus disease 2019: Retrospective study," *BMJ*, vol. 368, pp. 1091–1103, Mar. 2020.

[3] N. El-Rashidy, S. Abdelrazik, T. Abuhmed, E. Amer, F. Ali, J.-W. Hu, and S. El-Sappagh, "Comprehensive survey of using machine learning in the COVID-19 pandemic," *Diagnostics*, vol. 11, no. 7, pp. 1155–1199, Jun. 2021.

[4] C.-C. Lai, T.-P. Shih, W.-C. Ko, H.-J. Tang, and P.-R. Hsueh, "Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges," *Int. J. Antimicrobial Agents*, vol. 55, no. 3, Mar. 2020, Art. no. 105924.

[5] C. McDermott, M. Lacki, B. Sainsbury, J. Henry, M. Filippov, and C. Rossa, "Sonographic diagnosis of COVID-19: A review of image processing for lung ultrasound," *Frontiers Big Data*, vol. 4, pp. 612561–612572, Mar. 2021.

[6] P. C. Resende, F. G. Naveca, R. D. Lins, F. Z. Dezordi, M. V. F. Ferraz, E. G. Moreira, D. F. Coelho, F. C. Motta, A. C. D. Paixao, L. Appolinario, and R. S. Lopes, "The ongoing evolution of variants of concern and interest of SARS-CoV-2 in Brazil revealed by convergent indels in the amino (N)-terminal domain of the spike protein," *Virus Evol.*, vol. 7, no. 2, pp. 1–11, Dec. 2021.

[7] E. Volz et al., "Assessing transmissibility of SARS-CoV-2 lineage B. 1.1. 7 in England," *Nature*, vol. 593, no. 7858, pp. 266–269, 2021.

[8] A. Ebadi, P. Xi, A. MacLean, S. Tremblay, S. Kohli, and A. Wong, "COVIDx-US—An open-access benchmark dataset of ultrasound imaging data for AI-driven COVID-19 analytics," 2021, *arXiv:2103.10003*.

[9] J. Born, N. Wiedemann, M. Cossio, C. Buhre, G. Brändle, K. Leidermann, J. Goulet, A. Aujayeb, M. Moor, B. Rieck, and K. Borgwardt, "Accelerating detection of lung pathologies with explainable ultrasound image analysis," *Appl. Sci.*, vol. 11, no. 2, p. 672, Jan. 2021.

[10] R. Aljondi and S. Alghamdi, "Diagnostic value of imaging modalities for COVID-19: Scoping review," *J. Med. Internet Res.*, vol. 22, no. 8, Aug. 2020, Art. no. e19673.

[11] X. Qian, R. Wodnicki, H. Kang, J. Zhang, H. Tchelepi, and Q. Zhou, "Current ultrasound technologies and instrumentation in the assessment and monitoring of COVID-19 positive patients," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 11, pp. 2230–2240, Nov. 2020.

[12] W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, and W. Tan, "Detection of SARS-CoV-2 in different types of clinical specimens," *Jama*, vol. 323, no. 18, pp. 1843–1844, 2020.

[13] L. Wang, Z. Q. Lin, and A. Wong, "COVID-net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Nov. 2020.

[14] J. Watson, P. Whiting, and J. Brush, "Interpreting a COVID-19 test result," *Brit. Med. J.*, vol. 369, pp. 1–7, May 2020.

[15] J. Kanne, B. Little, J. Chung, B. Elicker, and L. Ketai, "Essentials for radiologists on COVID-19: An update-radiology scientific expert panel," *Radiology*, vol. 296, pp. 113–114, Aug. 2020.

[16] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, "Correlation of chest CT and RT-PCR testing in Coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases," *Radiology*, vol. 296, pp. 32–40, Aug. 2020.

[17] C. West, V. Montori, and P. Sampathkumar, "COVID-19 testing: The threat of false-negative results," in *Proc. Mayo Clinic*, vol. 95, 2020, pp. 1127–1129.

[18] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, and W. Ji, "Sensitivity of chest CT for COVID-19: Comparison to RT-PCR," *Radiology*, vol. 296, no. 2, pp. 115–117, Aug. 2020.

[19] N. Awasthi, A. Dayal, L. R. Cenkeramaddi, and P. K. Yalavarthy, "Mini-COVIDNet: Efficient lightweight deep neural network for ultrasound based point-of-care detection of COVID-19," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 68, no. 6, pp. 2023–2037, Jun. 2021.

[20] J.-E. Bourcier, J. Paquet, M. Seinger, E. Gallard, J.-P. Redonnet, F. Cheddadi, D. Garnier, J.-M. Bourgeois, and T. Geeraerts, "Performance comparison of lung ultrasound and chest X-ray for the diagnosis of pneumonia in the ED," *Amer. J. Emergency Med.*, vol. 32, no. 2, pp. 115–118, Feb. 2014.

[21] L. J. Staub, R. R. M. Biscaro, and R. Mauricí, "Accuracy and applications of lung ultrasound to diagnose ventilator-associated pneumonia: A systematic review," *J. Intensive Care Med.*, vol. 33, no. 8, pp. 447–455, Aug. 2018.

[22] D. A. Lichtenstein, ‘‘Ultrasound examination of the lungs in the intensive care unit,’’ *Pediatric Crit. Care Med.*, vol. 10, no. 6, pp. 693–698, 2009.

[23] J. Diaz-Escobar, N. E. Ordóñez-Guillén, S. Villarreal-Reyes, A. Galaviz-Mosqueda, V. Kober, R. Rivera-Rodriguez, and J. E. Lozano Rizk, ‘‘Deep-learning based detection of COVID-19 using lung ultrasound imagery,’’ *PLoS ONE*, vol. 16, no. 8, Aug. 2021, Art. no. e0255886.

[24] S. B. Desai, A. Pareek, and M. P. Lungren, ‘‘Deep learning and its role in COVID-19 medical imaging,’’ *Intell.-Based Med.*, vols. 3–4, Dec. 2020, Art. no. 100013.

[25] D. Malla, V. Rathi, S. Gomber, and L. Upreti, ‘‘Can lung ultrasound differentiate between bacterial and viral pneumonia in children?’’ *J. Clin. Ultrasound*, vol. 49, no. 2, pp. 91–100, Feb. 2021.

[26] A. Omran, H. Awad, M. Ibrahim, S. El-Sharkawy, S. Elfiky, and A. R. Rezk, ‘‘Lung ultrasound and neutrophil lymphocyte ratio in early diagnosis and differentiation between viral and bacterial pneumonia in young children,’’ *Children*, vol. 9, no. 10, p. 1457, Sep. 2022.

[27] V. Berce, M. Tomazin, M. Gorenjak, T. Berce, and B. Lovrencic, ‘‘The usefulness of lung ultrasound for the aetiological diagnosis of community-acquired pneumonia in children,’’ *Sci. Rep.*, vol. 9, no. 1, pp. 17957–17967, Nov. 2019.

[28] G. Tan, X. Lian, Z. Zhu, Z. Wang, F. Huang, Y. Zhang, Y. Zhao, S. He, X. Wang, H. Shen, and G. Lyu, ‘‘Use of lung ultrasound to differentiate coronavirus disease 2019 (COVID-19) pneumonia from community-acquired pneumonia,’’ *Ultrasound Med. Biol.*, vol. 46, no. 10, pp. 2651–2658, Oct. 2020.

[29] J. W. Tsung, D. O. Kessler, and V. P. Shah, ‘‘Prospective application of clinician-performed lung ultrasonography during the 2009 H1N1 influenza a pandemic: Distinguishing viral from bacterial pneumonia,’’ *Crit. Ultrasound J.*, vol. 4, no. 1, pp. 1–10, Dec. 2012.

[30] B. P. Jones, E. T. Tay, I. Elikashvili, J. E. Sanders, A. Z. Paul, B. P. Nelson, L. A. Spina, and J. W. Tsung, ‘‘Feasibility and safety of substituting lung ultrasonography for chest radiography when diagnosing pneumonia in children,’’ *Chest*, vol. 150, no. 1, pp. 131–138, Jul. 2016.

[31] A. Lombardi, M. De Luca, D. Fabiani, F. Sabatella, C. Del Giudice, A. Caputo, L. Cante, M. Gambardella, S. Palermi, R. Tavarozzi, and V. Russo, ‘‘Ultrasound during the COVID-19 pandemic: A global approach,’’ *J. Clin. Med.*, vol. 12, no. 3, pp. 1057–1072, 2023.

[32] L. Vetrugno, M. Baciarello, E. Bignami, A. Bonetti, F. Saturno, D. Orso, R. Girometti, L. Cereser, and T. Bove, ‘‘The ‘pandemic’ increase in lung ultrasound use in response to COVID-19: Can we complement computed tomography findings? A narrative review,’’ *Ultrasound J.*, vol. 12, no. 1, pp. 1–11, 2020.

[33] J. Born, G. Brändle, M. Cossio, M. Disdier, J. Goulet, J. Roulin, and N. Wiedemann, ‘‘POCOVID-net: Automatic detection of COVID-19 from a new lung ultrasound imaging dataset (POCUS),’’ 2020, *arXiv:2004.12084*.

[34] B. Barros, P. Lacerda, C. Albuquerque, and A. Conci, ‘‘Pulmonary COVID-19: Learning spatiotemporal features combining CNN and LSTM networks for lung ultrasound video classification,’’ *Sensors*, vol. 21, no. 16, pp. 5486–5511, Aug. 2021.

[35] V. V. Khanna, K. Chadaga, N. Sampathila, S. Prabhu, R. Chadaga, and S. Umakanth, ‘‘Diagnosing COVID-19 using artificial intelligence: A comprehensive review,’’ *Netw. Model. Anal. Health Informat. Bioinf.*, vol. 11, no. 1, pp. 1–23, Dec. 2022.

[36] J. Wang, X. Yang, B. Zhou, J. J. Sohn, J. Zhou, J. T. Jacob, K. A. Higgins, J. D. Bradley, and T. Liu, ‘‘Review of machine learning in lung ultrasound in COVID-19 pandemic,’’ *J. Imag.*, vol. 8, no. 3, pp. 65–83, Mar. 2022.

[37] L. Zhao and M. A. Lediju Bell, ‘‘A review of deep learning applications in lung ultrasound imaging of COVID-19 patients,’’ *BME Frontiers*, vol. 2022, pp. 1–17, Jan. 2022.

[38] R. Arntfield et al., ‘‘Development of a convolutional neural network to differentiate among the etiology of similar appearing pathological B lines on lung ultrasound: A deep learning study,’’ *BMJ Open*, vol. 11, no. 3, 2021, Art. no. e045120.

[39] O. Frank, N. Schipper, M. Vaturi, G. Soldati, A. Smargiassi, R. Inchingolo, E. Torri, T. Perrone, F. Mento, L. Demi, M. Galun, Y. C. Eldar, and S. Bagon, ‘‘Integrating domain knowledge into deep networks for lung ultrasound with applications to COVID-19,’’ *IEEE Trans. Med. Imag.*, vol. 41, no. 3, pp. 571–581, Mar. 2022.

[40] U. Khan, F. Mento, L. Nicolussi Giacomaz, R. Trevisan, A. Smargiassi, R. Inchingolo, T. Perrone, and L. Demi, ‘‘Deep learning-based classification of reduced lung ultrasound data from COVID-19 patients,’’ *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 69, no. 5, pp. 1661–1669, May 2022.

[41] S. Roy, W. Menapace, S. Oei, B. Luijten, E. Fini, C. Saltori, I. Huijben, N. Chennakeshava, F. Mento, A. Sentelli, and E. Peschiera, ‘‘Deep learning for classification and localization of COVID-19 markers in point-of-care lung ultrasound,’’ *IEEE Trans. Med. Imag.*, vol. 39, no. 8, pp. 2676–2687, Aug. 2020.

[42] L. Carrer, E. Donini, D. Marinelli, M. Zanetti, F. Mento, E. Torri, A. Smargiassi, R. Inchingolo, G. Soldati, L. Demi, F. Bovolo, and L. Bruzzone, ‘‘Automatic pleural line extraction and COVID-19 scoring from lung ultrasound data,’’ *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 11, pp. 2207–2217, Nov. 2020.

[43] A. G. Dastider, F. Sadik, and S. A. Fattah, ‘‘An integrated autoencoder-based hybrid CNN-LSTM model for COVID-19 severity prediction from lung ultrasound,’’ *Comput. Biol. Med.*, vol. 132, May 2021, Art. no. 104296.

[44] A. Thomas, G. Haljan, and A. Mitra, ‘‘Lung ultrasound findings in a 64-year-old woman with COVID-19,’’ *Can. Med. Assoc. J.*, vol. 192, no. 15, p. 399, Apr. 2020.

[45] A. Gudigar, U. Raghavendra, S. Nayak, C. P. Ooi, W. Y. Chan, M. R. Gangavarapu, C. Dharmik, J. Samanth, N. A. Kadri, K. Hasikin, and P. D. Barua, ‘‘Role of artificial intelligence in COVID-19 detection,’’ *Sensors*, vol. 21, no. 23, pp. 8045–8084, 2021.

[46] H. Kerdegari, N. T. H. Phung, A. McBride, L. Pisani, H. V. Nguyen, T. B. Duong, R. Razavi, L. Thwaites, S. Yacoub, A. Gomez, and V. Consortium, ‘‘B-line detection and localization in lung ultrasound videos using spatiotemporal attention,’’ *Appl. Sci.*, vol. 11, no. 24, p. 11697, Dec. 2021.

[47] T. Lum, M. Mahdavi, O. Frenkel, C. Lee, M. H. Jafari, F. T. Dezaki, N. V. Woudenberg, A. N. Gu, P. Abolmaesumi, and T. Tsang, ‘‘Imaging biomarker knowledge transfer for attention-based diagnosis of COVID-19 in lung ultrasound videos,’’ in *Proc. Int. Workshop Adv. Simplifying Med. Ultrasound*, 2021, pp. 159–168.

[48] R. Moshavegh, K. L. Hansen, H. Moller-Sorensen, M. B. Nielsen, and J. A. Jensen, ‘‘Automatic detection of B-lines in in vivo lung ultrasound,’’ *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 66, no. 2, pp. 309–317, Feb. 2018.

[49] O. Karakus, N. Anantrasirichai, A. Aguersif, S. Silva, A. Basarab, and A. Achim, ‘‘Detection of line artifacts in lung ultrasound images of COVID-19 patients via nonconvex regularization,’’ *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 11, pp. 2218–2229, Nov. 2020.

[50] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ‘‘MobileNets: Efficient convolutional neural networks for mobile vision applications,’’ 2017, *arXiv:1704.04861*.

[51] S. Perera, S. Adhikari, and A. Yilmaz, ‘‘Pocformer: A lightweight transformer architecture for detection of COVID-19 using point of care ultrasound,’’ in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 195–199.

[52] G. Muhammad and M. S. Hossain, ‘‘COVID-19 and non-COVID-19 classification using multi-layers fusion from lung ultrasound images,’’ *Inf. Fusion*, vol. 72, pp. 80–88, Aug. 2021.

[53] B. Zhao, X. Li, and X. Lu, ‘‘TTH-RNN: Tensor-train hierarchical recurrent neural network for video summarization,’’ *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3629–3637, Apr. 2021.

[54] J. Gao, X. Yang, Y. Zhang, and C. Xu, ‘‘Unsupervised video summarization via relation-aware assignment learning,’’ *IEEE Trans. Multimedia*, vol. 23, pp. 3203–3214, 2021.

[55] R. P. Mathews, M. R. Panicker, A. R. Hareendranathan, Y. T. Chen, J. L. Jaremko, B. Buchanan, K. V. Narayan, and G. Mathews, ‘‘Unsupervised multi-latent space RL framework for video summarization in ultrasound imaging,’’ *IEEE J. Biomed. Health Informat.*, vol. 27, no. 1, pp. 227–238, Jan. 2023.

[56] K. Zhou, Y. Qiao, and T. Xiang, ‘‘Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward,’’ in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–8.

[57] B. Mahasseni, M. Lam, and S. Todorovic, ‘‘Unsupervised video summarization with adversarial LSTM networks,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2982–2991.

[58] A. Petrosian, ‘‘Kolmogorov complexity of finite sequences and recognition of different preictal EEG patterns,’’ in *Proc. 8th IEEE Symp. Comput.-Based Med. Syst.*, Oct. 1995, pp. 212–217.

[59] C. Goh, B. Hamadicharef, G. T. Henderson, and E. C. Ifeachor, ‘‘Comparison of fractal dimension algorithms for the computation of EEG biomarkers for dementia,’’ in *Proc. Int. Conf. Comput. Intell. Med. Healthcare*, 2005, pp. 1–9.

[60] R. Esteller, G. Vachtsevanos, J. Echauz, and B. Litt, "A comparison of waveform fractal dimension algorithms," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 2, pp. 177–183, Feb. 2001.

[61] T. Higuchi, "Approach to an irregular time series on the basis of the fractal theory," *Phys. D, Nonlinear Phenomena*, vol. 31, no. 2, pp. 277–283, Jun. 1988.

[62] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of DNA nucleotides," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 49, no. 2, pp. 1685–1689, Feb. 1994.

[63] R. Hardstone, S.-S. Poil, G. Schiavone, R. Jansen, V. V. Nikulin, H. D. Mansvelder, and K. Linkenkaer-Hansen, "Detrended fluctuation analysis: A scale-free view on neuronal oscillations," *Frontiers Physiol.*, vol. 3, pp. 450–463, 2012.

[64] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[65] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012.

[66] E. A. Nehary, S. Rajan, and C. Rossa, "Comparison of COVID-19 classification via imagenet-based and RadImagenet-based transfer learning models with random frame selection," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Jul. 2023, pp. 1–6.

[67] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[68] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[70] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[71] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[73] J. Born, N. Wiedemann, M. Cossio, C. Buhre, G. Brandle, K. Leidermann, and A. Aujayeb, "L2 accelerating COVID-19 differential diagnosis with explainable ultrasound image analysis: An AI tool," *Thorax*, vol. 76, pp. 230–231, Jan. 2021.

[74] D. A. Lichtenstein, "BLUE-protocol and FALLS-protocol: Two applications of lung ultrasound in the critically ill," *Chest*, vol. 147, no. 6, pp. 1659–1670, 2015.

[75] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images," *Pattern Recognit. Lett.*, vol. 138, pp. 638–643, Oct. 2020.

[76] F. Mento, U. Khan, F. Faita, A. Smargiassi, R. Inchingolo, T. Perrone, and L. Demi, "State of the art in lung ultrasound, shifting from qualitative to quantitative analyses," *Ultrasound Med. Biol.*, vol. 48, no. 12, pp. 2398–2416, Dec. 2022.

[77] S. Altmayer, M. Zanon, G. S. Pacini, G. Watte, M. C. Barros, T.-L. Mohammed, N. Verma, E. Marchiori, and B. Hochhegger, "Comparison of the computed tomography findings in COVID-19 and other viral pneumonia in immunocompetent adults: A systematic review and meta-analysis," *Eur. Radiol.*, vol. 30, no. 12, pp. 6485–6496, Dec. 2020.

[78] J. Mendel, J. Lee, and D. Rosman, "Current concepts imaging in COVID-19 and the challenges for low and middle income countries," *J. Global Radiol.*, vol. 6, no. 1, pp. 1–10, Jun. 2020.

[79] A. Fernández, V. López, M. Galar, M. J. del Jesus, and F. Herrera, "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowl.-Based Syst.*, vol. 42, pp. 97–110, Apr. 2013.

EBRAHIM A. NEHARY (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in biomedical engineering from Cairo University, Cairo, Egypt, in 2013 and 2017, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Department of Systems and Computer Engineering, Carleton University, ON, Canada. His main research interests include deep learning, machine learning, biomedical signal and image analysis, and computer vision. He is a member of IEEE Young Professionals, IEEE Signal Processing Society, IEEE Instrumentation & Measurement Society, and IEEE Engineering in Medicine and Biology Society.

SREERAMAN RAJAN (Senior Member, IEEE) received the B.E. degree in electronics and communications from Bharathiyar University, Coimbatore, India, in 1987, the M.Sc. degree in electrical engineering from Tulane University, New Orleans, LA, USA, in 1992, and the Ph.D. degree in electrical engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2004. He joined Carleton University as a Tier 2 Canada Research Chair of Advanced Sensors Systems and Signal Processing with the Department of Systems and Computer Engineering, in July 2015, after working in industry and government research laboratories for two decades. He is currently a Full Professor. He was the Director of Ottawa-Carleton Institute for Biomedical Engineering (OCIBME), from 2020 to 2022. He has authored more than 200 journals and conference papers. His research interests include active and passive sensing and sensor signal processing, application of such sensors to a wide variety of problems in areas of civilian and defence applications, applied machine and deep learning, and quantum sensor signal/image processing. He chairs the IEEE Ottawa EMBS and AESS Chapters and has been the North America Director for IEEE CTSoc, since 2021. He is also an elected Member-at-Large of IEEE CTSoc. He received the IEEE MGA Achievement Award, in 2012, Queen Elizabeth II Diamond Jubilee Medal, in 2012, and W. S. Read Outstanding Service Award, in 2016. He has led several IEEE conferences and has been a reviewer of several IEEE journals and conferences.

CARLOS ROSSA (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees in mechanical engineering from the École Nationale d Ingnieurs de Metz, Metz, France, and the Ph.D. degree in mechatronics and robotics from Sorbonne Université (UPMC), Paris, France, under the auspices of the Commissariat ã l'Energie Atomique (CEA). He is currently an Associate Professor with the Department of Systems and Computer Engineering, Carleton University. His research interests include biomedical instrumentation, medical robotics, and image-guided percutaneous surgery.