A large margin piecewise linear classifier with fusion of deep features in the diagnosis of COVID-19

Neda Azouji [a], Ashkan Sami [a,*], Mohammad Taheri [a], Henning Müller [b]

[a] *Department of Computer Science and Engineering and IT, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran*
[b] *Department of Business Information Systems University of Applied Sciences Western Switzerland, Sierre (HES SO), Switzerland*

A R T I C L E I N F O

A B S T R A C T

The world has experienced epidemics of coronavirus infections several times over the last two decades. Recent studies have shown that using medical imaging techniques can be useful in developing an automatic computer-aided diagnosis system to detect pandemic diseases with high accuracy at an early stage. In this study, a large margin piecewise linear classifier was developed to diagnose COVID-19 compared to a wide range of viral pneumonia, including SARS and MERS, using chest x-ray images. In the proposed method, a preprocessing pipeline was employed. Moreover, deep pre- and post-rectified linear unit (ReLU) features were extracted using the well-known VGG-Net19, which was fine-tuned to optimize transfer learning. Afterward, the canonical correlation analysis was performed for feature fusion, and fused deep features were passed into the LMPL classifier. The introduced method reached the highest performance in comparison with related state-of-the-art methods for two different schemes (normal, COVID-19, and typical viral pneumonia) and (COVID-19, SARS, and MERS pneumonia) with 99.39% and 98.86% classification accuracy, respectively.

1. Introduction

The COVID-19 virus is very contagious and has been spreading worldwide, leading to an ongoing pandemic. The world has encountered occurrences of other severe coronavirus infections over the last few decades, including [1]:

- The Severe Acute Respiratory Syndrome (SARS) epidemic originating in Guangdong, China in 2002–2003;
- The Middle East Respiratory Syndrome (MERS) epidemic originating in Jeddah, Saudi Arabia in 2011;
- The current COVID-19 pandemic that originated in Wuhan, China in 2019.

Various types of medical images can be used to diagnose this disease. Two commonly used types are Computed Tomography (CT) and x-rays of the chest [2].

In emerging diseases such as COVID-19, the number of available samples is initially small and, sometimes, insufficient to train deep learning models well because they tend to overfit the training data. The present study introduces a large margin piecewise linear (LMPL) classifier as a proper learning method to get the most out of the available training data. We took advantage of the efficiency of large margin classifiers to construct a piecewise linear model. The new algorithm, called LMPL, is a novel formulation with interesting benefits, such as more accurate results than the traditional approaches like support vector machine (SVM) classifiers and k-Nearest Neighbors (kNN). The proposed LMPL addresses the following challenges that are of great importance in machine learning:

- *Small sample size:* Medical datasets are difficult to find due to privacy constraints that limit data sharing;
- *Overfitting to training data:* This problem limits the generality of models to unseen data, but it is essential, particularly in transfer learning tasks, where the distribution of source and target can be so different.
- *Nonlinearity:* Most classes or clusters in real-world problems are not linearly separable.
- *Multiclass classification:* Some classifiers, e.g., SVMs, have a binary structure and need extensions to solve this problem, such as the One versus One (OvO) and One versus All (OvA) approaches.

---

* Corresponding author.
E-mail addresses: azouji@shirazu.ac.ir (N. Azouji), sami@shirazu.ac.ir (A. Sami), motaheri@shirazu.ac.ir (M. Taheri), henning.mueller@hevs.ch (H. Müller).

- *Multi-modality:* When a class has several modalities, its samples are located in various positions of the search space. This makes it difficult for the most common classifiers to work well.

In this paper, the VGG-net model [3] was used for feature extraction. Indeed, the deep VGG-Net19 model was retrained by transfer learning to learn deep features of x-ray images. The deep features were obtained from the seventh fully connected layer, i.e., the last layer before the output. The activation values of both pre- and post-rectified linear units (ReLU) were marked as deep features with feature vectors of dimension 4096. A preprocessing pipeline was developed for quality improvement of x-ray images. The canonical correlation analysis (CCA) was performed for feature fusion [4], and the fused features were passed to the LMPL classifier.

The introduced approach was applied for two classification tasks in this study.

I. Distinguishing COVID-19 from typical viral pneumonia and normal x-ray images
II. Distinguishing COVID-19 from the two severe CoV family members, i.e., SARS and MERS

1.1. Contribution

The contributions of this paper are outlined as follows:

- In order to learn discriminant and robust feature representations, a novel COVID-19 diagnostic method is introduced that integrates a deep feature extraction and fusion methodology.
- To gain the most informative knowledge from limited labeled data, an LMPL classifier is introduced using a cellular structure to better exploit the limited training data.

2. Related work

Since the spread of COVID-19, there has been an increasing effort to advance deep models for COVID-19 detection using medical images, such as CT scans and x-rays. In a review of the literature, some of the recent developments in deep networks for COVID-19 diagnosis are mentioned herein. Ozturk et al. [5] developed DarkCovidNet to diagnose COVID-19 using raw chest x-ray images. Their network reached the accuracy of 98.08% and 87.02% in binary and multiclass schemes. Taik et al. [6] introduced the COVIDGR-1.0 dataset and proposed the COVID-SDNet method that combines segmentation, data-augmentation, and data transformation. They achieved a high generalization accuracy of 97.72%. Mahmud et al. [7] proposed a deep CNN called CovXNet, which uses depthwise dilated convolution for efficient feature extraction from x-ray images. At first, a large number of normal and Non-COVID pneumonia (viral and bacterial) were used to train the proposed CovXNet. Then, transfer learning was used to fine-tune and retrain the net by a smaller number of COVID-19 and other pneumonia cases. CovXNet was designed and trained by various image resolutions, and a stacking algorithm was employed for optimization. CovXNet achieved an accuracy of 90.2% for multiclass classification into normal, COVID19, viral, and bacterial pneumonia. Rahman et al. [8] compiled a large x-ray dataset with its corresponding lung mask to identify COVID-19. They investigated the effect of different image enhancement techniques on COVID-19 detection. Various deep CNNs were trained on plain and segmented lung images. The method reached the best performance with 96.2% accuracy in plain images by ChexNet and 95.11% classification accuracy in segmented lung images by DenseNet201. Jain et al. [9] compared multiple deep CNN models, including Inception V3, Xception, and ResNeXt, among which Xception had the best performance with the highest accuracy of 97.97% in classifying COVID-19 using chest x-ray images. Toğaçar et al. [10] restructured the COVID-19 dataset by

employing the fuzzy color technique as a preprocessing method. The images organized together with the original images were stacked. SqueezNet and MobileNetV2 were used as deep feature extractors, along with the SVM classifier, and trained on the stacked dataset. They obtained 96.28% classification accuracy using MobileNetV2. Hemdan et al. [11] introduced COVIDX-Net, which consists of seven CNN models. They employed a deep learning model to detect COVID-19 using x-ray images. Karakanis et al. [12] proposed a lightweight deep learning architecture to detect COVID-19 from x-ray images. The authors used a conditional generative adversarial network for image augmentation by generating syntactic x-ray images. They achieved 98.7% accuracy in binary classification between COVID-19 and normal pneumonia and 98.3% accuracy in a three-class model with normal, COVID-19, and bacterial pneumonia. Wang and Wong [13] developed an accurate deep network named COVID-Net for COVID-19 diagnosis with a classification accuracy of 92.4% in the two-class problem. Ioannis et al. [14] used 224 COVID-19 images to develop a deep model. They achieved the accuracy of 98.75% for a two-class problem and 93.48% for a three-class problem. Jin et al. [15] introduced a hybrid ensemble method that includes three steps to identify COVID-19 using x-ray images. At first, pre-trained Alexnet is used as a feature extractor; then, the ReliefF method is adopted to sort extracted deep features; and after reducing features' dimension, an SVM classifier is trained on selected features. They obtained 98.64% overall accuracy in multiclass classification of normal, COVID-19, and viral pneumonia. Narin et al. [16] employed the ResNet50 model for COVID-19 diagnosis and obtained an accuracy of 98% using chest x-ray images. Sethy and Behera [17] extracted deep features from x-ray images employing several CNN models and classified them with an SVM classifier. They reported the best performance for the model using the ResNet50 feature extraction and an SVM classifier. Minaee et al. [18] developed a deep learning framework to predict COVID-19 in chest x-rays. They fine-tuned four CNN models, including ResNet18, ResNet50, SqueezeNet, and DenseNet, and achieved promising results in several tasks. Various other studies have also been recently conducted on COVID-19 detection, employing several deep learning models with CT images [2,19–23] and Lung Ultrasound (LUS) [24].

A few studies have focused on the diagnosis of SARS and MERS with x-ray chest images. In a study on MERS, Hamimi [25] reported that features like the indicators of pneumonia could be found in the chest CT scan and x-ray images. Xie et al. [26] introduced a model to distinguish SARS from typical pneumonia using X-ray images. They employed three conventional classifiers: neural networks, C4.5, and Classification And Regression Tree (CART). Tahir et al. [27] employed CNN models for COVID-19 detection using a dataset containing SARS and MERS images. The authors developed a novel image preprocessing technique and deep learning algorithms and reported the outperformance of four CNN methods, including SqueezeNet, ResNet18, Inceptionv3, and Dense-Net201, among which InceptionV3 achieved the highest accuracy (98.22%). Abbas et al. [28] proposed a method based on DeTraC deep CNN to classify COVID-19 using chest x-ray images. The DeTraC model consisted of three phases, i.e., Decomposition, Transfer, and Composition. DeTraC uses a class decomposition mechanism to investigate class boundaries with any irregularities in the image dataset. They achieved 93.1% accuracy in a three-class problem, including normal, COVID-19, and SARS pneumonia. Xuanyang et al. [29] introduced a SARS detection system using a lung segmentation technique and feature extraction. They applied and compared several classification methods, including neural networks, decision trees, and regression trees. The highest detection accuracy was achieved from the regression trees. However, for pneumonia classification, NN-based models produced reasonably good results.

As presented in this section, researchers worldwide have introduced several approaches to detect COVID-19, as well as SARS or MERS pneumonia. Since the features of lung infections are the same in these diseases, it is problematic for an expert to distinguish them; so, well-
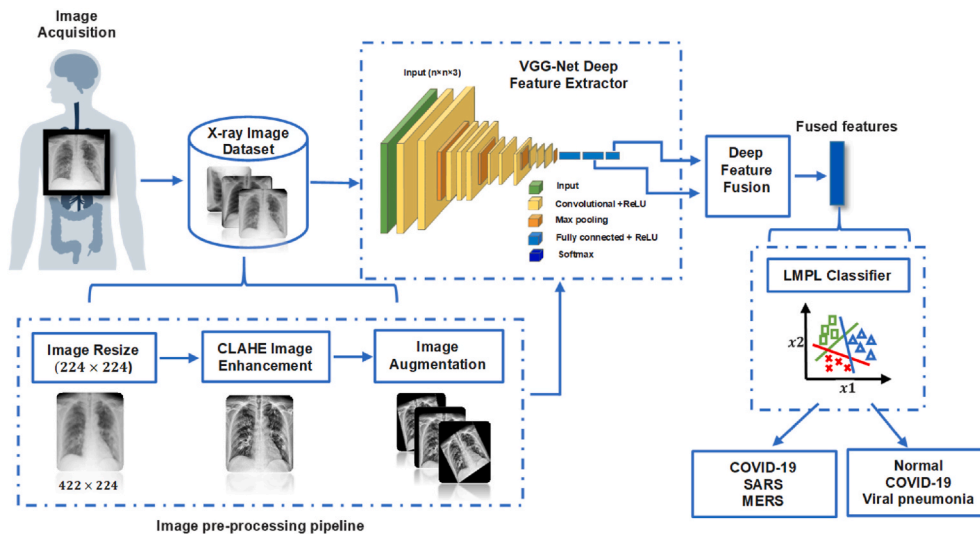
Fig. 1. Schematic of the overall proposed diagnostic system.

Table 1
Details of the X-ray image dataset.

Types	No. of X-ray Images	Source Database
Covid-19	423	SIRM-ITALIAN [31]
		Novel Coronavirus 2019 Dataset [32]
		Radiopaedia [33]
		Chest Imaging (Spain) at thread reader [34]
		COVID-19 Radiography Database [35]
SARS	134	SARS, MERS X-ray Images Dataset [36]
MERS	144	SARS, MERS X-ray Images Dataset [36]
Normal	1341	Chest X-ray Images (pneumonia) [37]
Typical Viral Pneumonia	1345	Chest X-ray Images (pneumonia) [37]

trained models are needed to develop an accurate and useful method in the treatment and decision-making. This study devises an effective method to distinguish the current COVID-19 from a wide range of viral pneumonia, including the same CoV family members like SARS and MERS.

3. Materials and methods

This section provides the details of the proposed COVID-19 diagnostic method. After preprocessing, in order to enhance x-ray images, the popular VGG-Net is used as a deep feature extractor from x-ray images, and the CCA feature fusion technique is applied to merge the extracted features. The LMPL classifier is introduced for coronavirus detection. Fig. 1 illustrates a schema of the proposed COVID-19 diagnostic model.

3.1. X-ray dataset

In this research, several open-access chest x-ray datasets are combined as a new large dataset on the coronavirus family. The combined dataset consists of six classes whose details are summarized in Table 1. The total number of images of the dataset is 3387. The dataset is balanced by image augmentation to reach 6179 images, as mentioned in *Sec. III. 3) Image augmentation.*

3.2. Image preprocessing pipeline

Medical images often contain a low dynamic range. This affects the overall performance of image processing approaches. Hence, it may become harder to evaluate them visually [8]. The preprocessing is performed to enhance the image information and visual quality through contrast enhancement. Another critical issue in medical imaging is imbalanced data, where the number of positive cases is lower than that of the negative ones.

1) Resizing x-ray images

In order to feed the images into CNNs, they should be resized to the input dimensions of the deep network. For VGG-Net, images are resized to 224 × 224 pixels.

2) Contrast limited adaptive histogram equalization

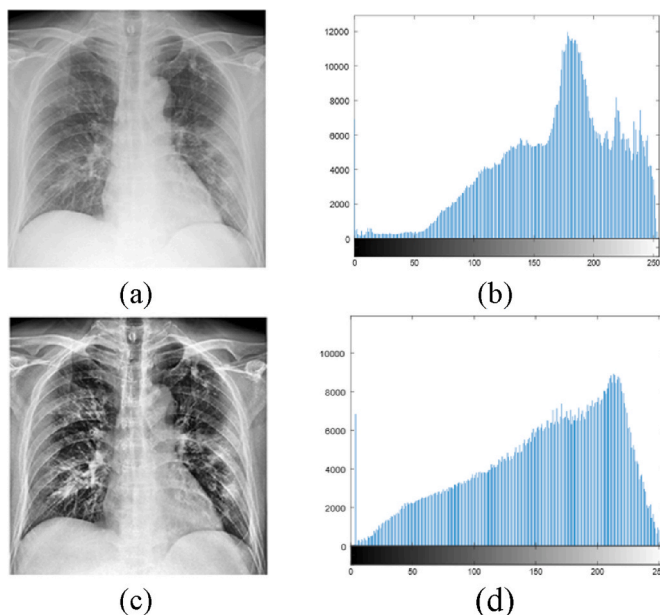Histogram equalization (HE) is an ordinary method widely used to

Fig. 2. CLAHE image enhancement. (a) COVID-19 X-ray image, (b) Histogram of (a), (c) CLAHE enhanced COVID-19 image, and (d) Histogram of (b).
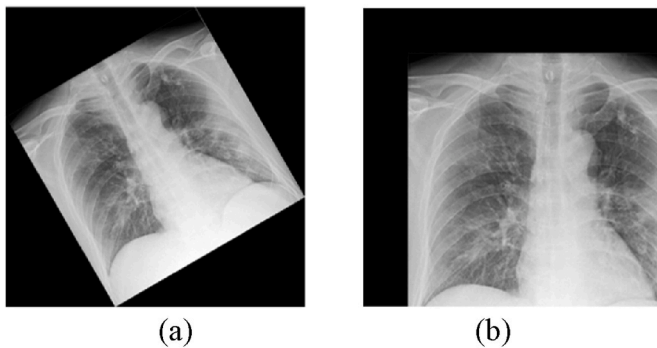
Fig. 3. Image augmentation. (a) Image rotation by 30° counter-clockwise, and (b) Image translation in the horizontal and vertical direction by 15%.

improve the contrast of an image and make algorithms more robust under varying conditions. HE produces a uniform distribution by computing the transformation function and it can help adjust an image by spreading out the most common intensity values of pixels in the range of 0 (black) to 255 (white). The contrast limited adaptive histogram equalization (CLAHE) [30] is an improved HE approach that enhances local contrast and edges in small image regions by calculating the equalization for each part in the image. Fig. 2 illustrates the image enhancement by the CLAHE method. It can be seen that the histogram is stretched around the entire spectrum and over all pixels of the image, which should facilitate the extraction of homogeneous features across all datasets.

3) Image augmentation

In order to extend the number of training samples and avoid the risk of overfitting, the data is augmented by applying different types of transformations. This makes the dataset suitable for deep learning. Deep

learning algorithms such as CNNs generally outperform a larger dataset rather than a smaller one. Besides, the imbalanced class distribution has a considerable impact on the effectiveness of the classification models. Consequently, the size of classes is balanced in the training set through data augmentation (see Fig. 3). Translation can be a very useful transformation to avoid positional bias in the data. In other words, translation is very helpful as most lesions can be located almost anywhere in the image.

In this work, rotation and translation operations are used to increase the number of images (upsampling) and balance the class instances with augmentation. Images are rotated 5, 10, 15, and 30°, and instances are also translated in a horizontal or vertical direction by 5%, 10%, 15% in the SARS and MERS classes that have few images, i.e., 134 and 144 ones, respectively. Finally, 1072 and 1152 images are obtained in the SARS and MERS classes, respectively. In the COVID-19 class with 423 instances, 15-degree rotation and 10% translation are performed to reach 1269 images. No augmentation is applied on normal and typical viral Pneumonia with 1341 and 1345 images. All augmented x-ray images can be found in the Zenodo repository (https://zenodo.org/record/4691987).

3.3. Deep feature extraction

Deep features are extracted by pre-trained CNN models. The fully connected (FC) layers are used to extract deep features. Since VGG-Net [3] was successful in extracting deep features for various tasks, the VGG-Net is employed as the main feature extractor to obtain robust automatic features.

The input image size in VGG networks is $224 \times 224 \times 2$. Fig. 4 illustrates an architectural overview of the VGG-Net. When applying VGG-Net as a deep feature extraction network, the classification layer, i. e., the last fully connected layer, is removed. The features can be extracted as raw values or after being transformed by a ReLU, where an output x is mapped to $max(0, x)$. The activation values from the last
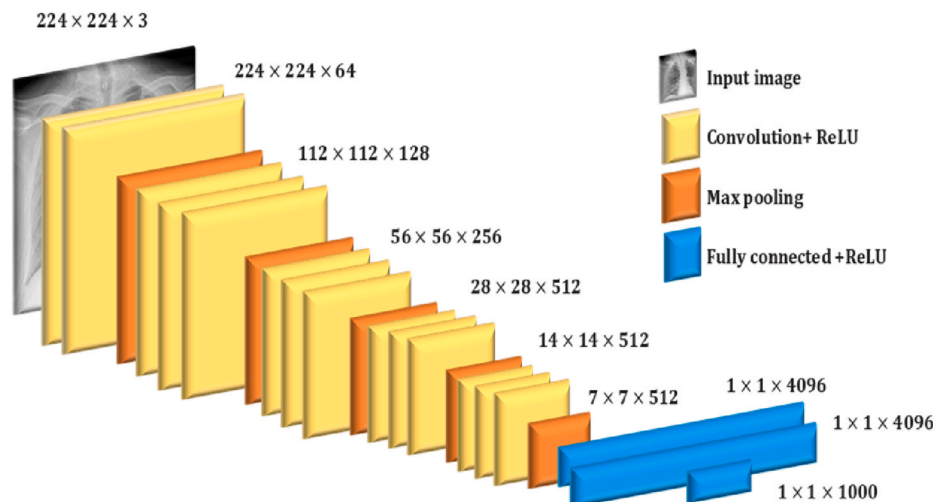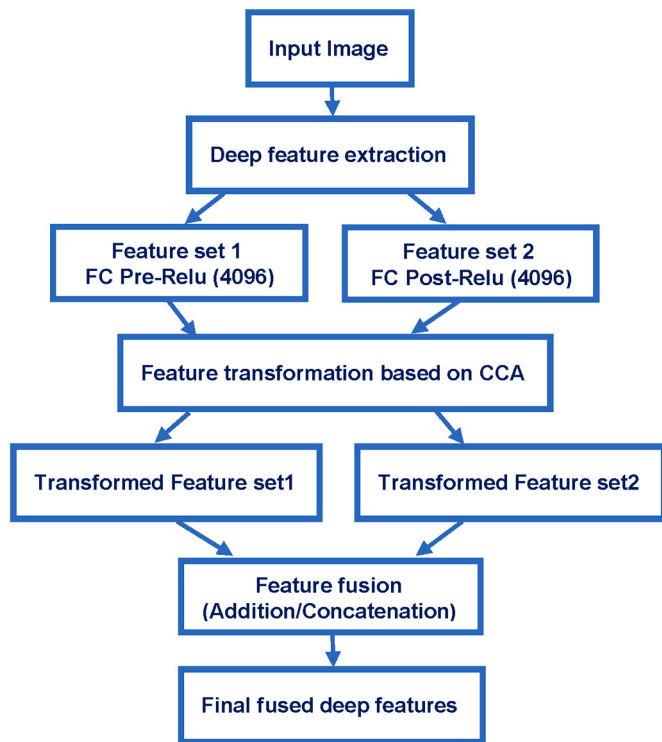

Fig. 4. VGG-Net architecture.

Fig. 5. Canonical correlation analysis (CCA) feature fusion.

hidden layer are extracted as the first deep feature vector (pre-ReLU features, 4096), while the post-ReLU feature set (post-ReLU features, 4096) is extracted as the second feature vector. For datasets not large enough, transfer learning can be used. In this paper, fine-tuning is applied with transfer learning. This is done by retraining VGG-Net (the weights of the model trained using the ImageNet dataset) on the COVID-19 x-ray images and transferring the deep features that are extracted to achieve better performance. When class labels are used to fine-tune the weights of the network, the extracted features are more discriminative in classifying the problem to distinguish COVID-19 from other viral pneumonia.

3.4. Deep feature fusion

The obtained deep features are combined to acquire more informative fusion features. This can help to minimize the impact of inadequate features obtained from a single CNN. Indeed, feature fusion efficiently

produces features that comprise rich information describing the image well. Appropriately combining two or more feature sets is not a trivial task. In this study, a well-known parallel feature fusion technique, i.e., CCA, is used. Feature fusion based on CCA creates two sets of transformations based on the correlation between two feature vectors resulting in transformed features with a higher correlation than the original feature sets. Suppose that $X \in R^{m \times n}$ and $Y \in R^{k \times n}$ are two feature sets, where n indicates the number of samples, and m and k denote the dimensions of X and Y, respectively. Let $S_{xx} \in R^{m \times m}$ and $S_{yy} \in R^{k \times k}$ represent the covariance matrices of X and Y, respectively, and $S_{xy} \in R^{m \times k}$ be the covariance matrix between sets, in which $S_{yx} = S_{xy}^T$. The overall covariance matrix $S = R^{(m+k) \times (m+k)}$ can be presented by (1):

$$S = \begin{pmatrix} cov(X) & cov(X,Y) \\ cov(Y,X) & cov(Y) \end{pmatrix} = \begin{pmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{pmatrix} \tag{1}$$

The goal of CCA is to define a linear combination of these feature sets, as shown in (2).

$$X^* = W_x^T X, \ Y^* = W_y^T Y \tag{2}$$

where W_x, W_y are associated coefficients of X and Y, respectively, that maximize the pair-wise correlation across the two feature sets. This objective is presented in (3):

$$corr(X^*, Y^*) = \frac{cov(X^*, Y^*)}{var(X^*).var(Y^*)} \tag{3}$$

where, $cov(X^*, Y^*) = W_x^T S_{xy} W_y$, $var(X^*) = W_x^T S_{xx} W_x$, and $var(Y^*) = W_y^T S_{yy} W_{xy}$. The combination of the transformed features is achieved by addition (summation) or concatenation, as follows in (4) and (5).

$$Z = X^* + Y^* = \begin{pmatrix} W_X \\ W_Y \end{pmatrix}^T \begin{pmatrix} X \\ Y \end{pmatrix} \tag{4}$$

$$Z = \begin{pmatrix} X^* \\ Y^* \end{pmatrix} = \begin{pmatrix} W_X & 0 \\ 0 & W_Y \end{pmatrix}^T \begin{pmatrix} X \\ Y \end{pmatrix} \tag{5}$$

where the resulting parallel fused feature matrix Z contains the canonical correlation discriminant features. Fig. 5 describes the CCA deep feature fusion procedure, where deep features are extracted by VGG-net from the input x-ray images.

3.5. Large margin piecewise linear (LMPL) classifier

In the last stage, fused features are used for the classification by the proposed LMPL classifier. A cellular structure is developed in this study such that space is partitioned by more than one hyper-plane, unlike in
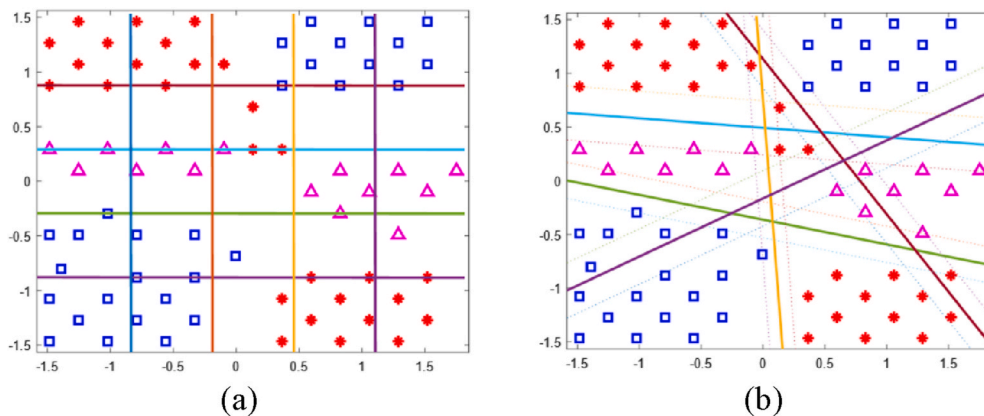

Fig. 6. An example of nonlinear multi-class classification with the proposed LMPL classifier. (a) The initial grid structure is constructed by eight initial hyperplanes, and (b) The final piecewise linear decision boundary is obtained by the LMPL classifier.
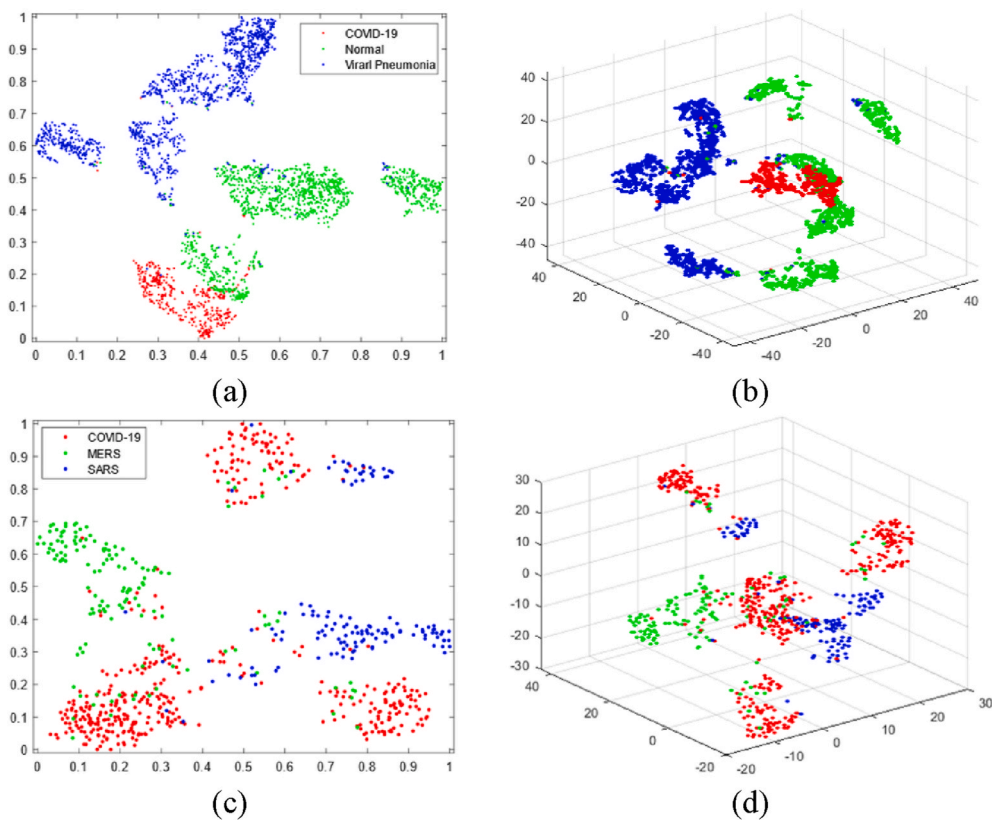
Fig. 7. Representation of classes by t-SNE Embedding. (a) Scatter plot in the two-dimensional space of Experiment1: Normal, COVID-19, and typical viral pneumonia, (b) 3D scatter plot of (a), (c) Scatter plot in the two-dimensional space of Expriment2: COVID-19, SARS, and MERS. and (d) 3D scatter plot of (c). according to its cell label. Cell labels are stored in the obtained cell table in the training phase.
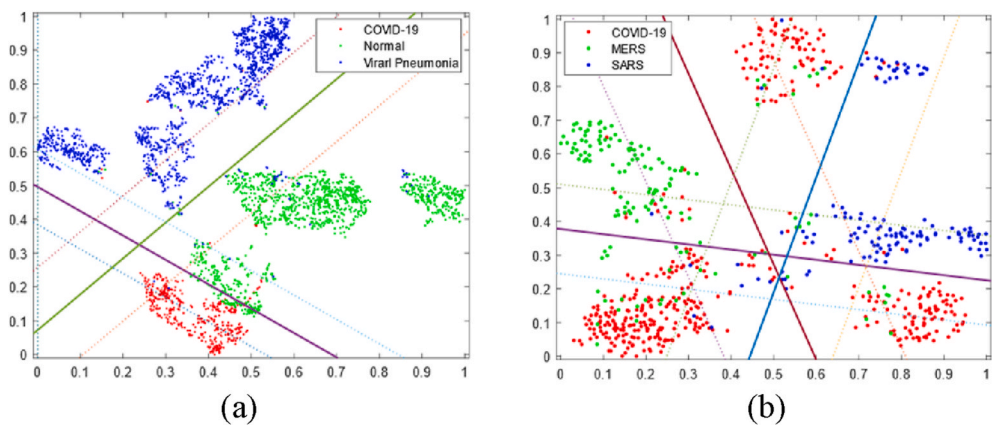
Fig. 8. Representation of the LMPL decision boundary on t-SNE embedding features with four initial hyperplanes (a) two remaining hyperplanes of Expriment1: Normal, COVID-19, and Typical Viral Pneumonia, and (b) three remaining hyperplanes in Expriment2: COVID-19, SARS, and MERS.

traditional models, e.g., SVMs, into some regions called cells. At first, a grid of cells is generated by predefined hyper-planes, as shown in Fig. 6 (a).

In Fig. 6(a), there are three classes: star (red), square (blue), and diamond (violet). Then, a cell table is created to store the information of each cell, such as the number of samples and its class label. The class labels are assigned to cells based on majority voting in each cell. Afterward, the parameters of the hyperplanes are tuned one by one in iterations. In other words, one of the hyper-planes is adjusted, considering that the others are fixed. Changing this hyperplane changes the region of cells bound to it from one side. Hence, the covering cell of a few instances may change. According to the actual label of each sample and

the associated label of the cells, samples may prefer to be on a specific side of the hyper-plane. This preference is considered their virtual labels. For example, if an instance prefers the right side of the hyperplane, its virtual label is 1, and if it chooses the left side, its virtual label is -1. After virtual labeling, there is a binary classification problem for the hyperplane under tuning. There are three sample groups in virtual labeling, as follows:

1. *Normal:* These samples are classified correctly on one side of the hyperplane. Their loss function is defined by the famous hinge loss, as shown in (6):

Table 2

Comparison of deep CNNs as an end-2-end network and as a feature extractor along with the proposed LMPL classifier (CNN$^+$) in Experiment 1: Normal, COVID-19, and typical viral pneumonia.

Method	Performance Metrics (%)					
	Feature ExtractorLayer	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
ResNet18		96.40 (4)	96.35 (5)	96.37 (4)	96.17 (4)	4.3
ResNet18$^+$		97.60 (7)	98.06 (6)	97.82 (6)	97.52 (6)	6.3
ResNet50	Avg_pool	96.20 (7)	96.16 (6)	96.17 (6)	95.91 (6)	6.3
ResNet50$^+$		98.43 (3)	98.49 (4)	98.45 (4)	98.10 (4)	3.8
ResNetV2	Avg_pool	94.04 (10)	92.61 (10)	93.28 (11)	93.18 (10)	10.3
ResNetV2$^+$		97.48 (8)	97.07 (10)	97.26 (10)	97.01 (9)	9.3
Inception	pool5	96.40 (4)	96.40 (4)	96.35 (5)	96.01 (5)	4.5
Inception$^+$		98.41 (4)	98.69 (3)	98.54 (3)	98.36 (3)	3.3
InceptionV3	Avg_pool	95.80 (8)	94.95 (9)	95.35 (8)	95.11 (8)	8.3
InceptionV3$^+$		98.26 (5)	98.10 (5)	98.18 (5)	97.75 (5)	5.0
Xception	Avg_pool	92.00 (11)	92.11 (11)	92.05 (10)	91.73 (11)	10.8
Xception$^+$		97.24 (9)	97.34 (9)	97.29 (9)	96.98 (10)	9.3
DenseNet201	Avg_pool	97.48 (2)	97.66 (2)	97.56 (2)	97.30 (3)	2.3
DenseNet201$^+$		98.85 (2)	98.86 (2)	98.85 (2)	98.58 (2)	2.0
SqueezeNet	Pool10	95.14 (9)	95.13 (8)	95.06 (9)	94.76 (9)	8.8
SqueezeNet$^+$		95.39 (11)	94.75 (11)	95.03 (11)	95.08 (11)	11.0
ShuffleNet	Node_200	96.34 (6)	95.68 (7)	96.00 (7)	95.82 (7)	6.8
ShuffleNet$^+$		97.20 (10)	97.56 (8)	97.36 (8)	97.07 (8)	8.5
AlexNet	fc7	97.24 (3)	97.57 (3)	97.40 (3)	97.33 (2)	2.8
AlexNet$^+$		97.67 (6)	97.84 (7)	97.75 (7)	97.39 (7)	6.8
VGGNet19	fc7	**98.21(1)**	**98.43 (1)**	**98.32 (1)**	**98.10 (1)**	**1.0**
VGGNet19$^+$		**98.87 (1)**	**99.08 (1)**	**98.97 (1)**	**98.81 (1)**	**1.0**
Avg. on CNN Avg on CNN$^+$		95.93	95.73	95.81	95.58	
		97.76	**97.80**	**97.77**	**97.51**	

*Bold numbers indicate the best performance.

Table 3

Comparison of deep CNNs as an end-2-end Network and as a feature extractor along with the proposed LMPL classifier (CNN$^+$) in Experiment 2: COVID-19, SARS, and MERS pneumonia.

Method	Performance Metrics (%)					
	Feature ExtractorLayer	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
ResNet18		87.47 (5)	84.52 (7)	85.86 (5)	88.02 (4)	5.3
ResNet18$^+$		91.97 (8)	90.66 (7)	91.22 (7)	92.72 (7)	7.3
ResNet50	Avg_pool	88.11 (4)	84.90 (6)	86.35 (4)	87.87 (5)	4.8
ResNet50$^+$		94.25 (6)	91.26 (5)	92.64 (5)	93.58 (5)	5.3
ResNetV2	Avg_pool	73.10 (11)	76.76 (9)	74.56 (11)	78.74 (10)	10.3
ResNetV2$^+$		91.45 (10)	88.99 (9)	90.12 (8)	91.58 (9)	9.0
Inception	pool5	83.76 (8)	85.38 (5)	84.38 (8)	86.59 (8)	7.3
Inception$^+$		95.68 (3)	94.35 (3)	94.95 (3)	95.72 (3)	3.0
InceptionV3	Avg_pool	80.70 (9)	76.38 (10)	77.95 (9)	80.46 (9)	9.3
InceptionV3$^+$		92.60 (7)	90.90 (6)	91.70 (6)	93.15 (6)	6.3
Xception	Avg_pool	75.04 (10)	74.49 (11)	74.75 (10)	78.60 (11)	10.5
Xception$^+$		95.46 (4)	89.00 (8)	90.10 (9)	91.73 (8)	7.3
DenseNet201	Avg_pool	91.83 (3)	91.60 (2)	91.57 (2)	92.87 (2)	2.3
DenseNet201$^+$		96.59 (2)	96.20 (2)	96.39 (2)	96.72 (2)	2.0
SqueezeNet	Pool10	85.64 (6)	84.51 (8)	85.02 (7)	87.30 (7)	7.0
SqueezeNet$^+$		85.39 (11)	86.12 (11)	85.45 (11)	87.87 (11)	11.0
ShuffleNet	Node_200	84.68 (7)	85.95 (4)	85.27 (6)	87.73 (6)	5.8
ShuffleNet$^+$		91.80 (9)	88.71 (10)	90.10 (9)	91.16 (10)	9.5
AlexNet	fc7	92.08 (2)	91.07 (3)	91.39 (3)	92.72 (3)	2.8
AlexNet$^+$		94.87 (5)	91.99 (4)	93.33 (4)	94.15 (4)	4.3
VGGNet19	fc7	**92.32 (1)**	**92.58 (1)**	**92.45 (1)**	**93.30 (1)**	**1.0**
VGGNet19$^+$		**96.64 (1)**	**96.29 (1)**	**96.46 (1)**	**96.86 (1)**	**1.0**
Avg. on CNN Avg. on CNN$^+$		85.88	84.38	84.50	86.75	
		93.34	**91.32**	**92.04**	**93.20**	

*Bold numbers indicate the best performance.

Table 4
Comparison of commom classifiers with the proposed LMPL classifier in Experiment 1: Normal, COVID-19, and typical viral pneumonia.

Method	Performance Metrics (%)				
	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
NaiveBayes	97.94 (12)	97.63 (12)	97.78 (12)	97.75 (12)	12.0
kNN (k = 3)	98.61 (6)	98.72 (6)	98.67 (6)	98.55 (6)	6.0
kNN (k= 5)	98.53 (7)	98.75 (5)	98.64 (7)	98.52 (8)	6.8
kNN (k = 7)	98.44 (8)	98.60 (8)	98.52 (8)	98.39 (9)	8.3
OvO SVM	98.42 (10)	98.52 (10)	98.47 (10)	98.30 (10)	10.0
OvA SVM	98.82 (3)	98.61 (7)	98.71 (5)	98.68 (4)	4.8
Decision Tree	98.45 (9)	98.56 (9)	98.50 (9)	98.55 (6)	8.3
AdaBoostM2	98.63 (5)	98.96 (4)	98.80 (4)	98.65 (5)	4.5
TotalBoost	98.68 (4)	99.06 (3)	98.87 (3)	98.78 (3)	3.3
Random Forrest	98.84 (2)	99.00 (2)	98.92 (2)	**98.84 (1)**	1.8
SoftMax	98.21 (11)	98.43 (11)	98.32 (11)	98.10 (11)	11.0
Proposed LMPL	**98.87 (1)**	**99.08 (1)**	**98.97 (1)**	98.81 (2)	**1.3**

*Bold numbers indicate the best performance.

Table 5
Comparison of common classifiers with the proposed LMPL classifier in Experiment 2: COVID-19, SARS, and MERS pneumonia.

Method	Performance Metrics (%)				
	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
NaiveBayes	89.20 (12)	94.23 (10)	91.25 (12)	92.58 (12)	11.5
kNN (k = 3)	96.08 (2)	95.43 (4)	95.72 (3)	96.29 (3)	3.0
kNN (k = 5)	95.31 (5)	94.95 (6)	95.08 (6)	95.72 (5)	5.3
kNN (k = 7)	94.35 (8)	94.57 (9)	94.42 (8)	95.15 (8)	8.3
OvO SVM	94.64 (7)	93.10 (11)	93.84 (9)	95.01 (9)	9.0
OvA SVM	95.86 (4)	94.73 (7)	95.28 (4)	96.01 (4)	4.8
Decision Tree	94.95 (6)	95.20 (5)	95.06 (7)	95.72 (5)	5.8
AdaBoostM2	93.07 (10)	94.66 (8)	93.82 (10)	94.58 (10)	9.5
TotalBoost	93.63 (9)	**96.83 (1)**	95.13 (5)	95.72 (5)	5.0
Random Forrest	96.03 (3)	95.86 (3)	95.93 (2)	96.58 (2)	2.5
SoftMax	92.32 (11)	92.58 (12)	92.45 (11)	93.30 (11)	11.3
Proposed LMPL	**96.64 (1)**	96.29 (2)	**96.46 (1)**	**96.86 (1)**	**1.3**

*Bold numbers indicate the best performance.

$$l(\boldsymbol{x})_{Normal^{\widetilde{y}}} = \max\left(0,\ 1 - \widetilde{y}\left(\boldsymbol{w}^T.\boldsymbol{x} + b\right)\right)$$
$$where: \quad \widetilde{y} = \{-1, +1\} \tag{6}$$

where $\widetilde{y}$ is the virtual label of sample x, and the hinge loss function is convex.

2. *Negative don't care:* These samples are classified incorrectly on both sides of the hyperplane. Therefore, it is not important on which side they are located, unless it is tested and proved that changing the position of samples affects the hyperplane. If the sample is located on the positive side, it tries to be located on the negative side, and vice versa. In other words, they are forced to be near the hyper-plane.

This loss function is defined by (7). By this function, the samples always get a penalty since they are misclassified.

$$l(\boldsymbol{x})_{DontCare^-} = \max(l(\boldsymbol{x})_{Normal^{(+1)}}, l(\boldsymbol{x})_{Normal^{(-1)}}) \tag{7}$$

3. *Positive don't care:* These samples are classified correctly on both sides of the hyperplane. Like in the second gro (negative don't care), it is not important on which side of the hyperplane they are. However, in this case, their distance to the hyper-plane is tested to be maximized. The total loss is defined by (8):

$$l(\boldsymbol{x})_{DontCare^+} = \min(l(\boldsymbol{x})_{Normal^{(+1)}}, l(\boldsymbol{x})_{Normal^{(-1)}}) \tag{8}$$

The *positive don't care group* is discarded here since its samples are classified correctly and help us to preserve convexity in the final objective function. Therefore, according to the two first groups, the proposed LMPL classifier optimizes each hyperplane based on the following objective function (9):

$$\underset{w}{minimize} \frac{1}{2} \boldsymbol{w}^2 + \left(C_1 \sum_{x \in Normal} l(\boldsymbol{x})_{Normal^{(y)}}\right) + \left(C_2 \sum_{x \in DC^-} l(\boldsymbol{x})_{DontCare^-}\right) \tag{9}$$

The scalar values C_1 and C_2 control the balance between the structural and empirical error. In this paper, both C_1 and C_2 are set to 1000.

The defined objective is optimized by quadratic programming. After optimizing each hyperplane, the cell table is updated in each iteration. Additional hyperplanes that fall out of the search space and are not useful in the classification are removed. After several iterations, the final piecewise linear decision boundary is constructed. In the test step, an unseen sample x is classified.

As mentioned, the LMPL classifier utilizes the cellular structure (see Fig. 6. (a)). Each cell forms a rule with a decision space regarding its position with respect to the hyper-planes. By assigning a label to each cell, both *multiclass* classification and *multi-modality* are structurally supported (the same as a decision tree). However, contrary to decision trees, the separator hyper-planes are shared in many decision spaces to use both local and global attributes of the training sample for class separation. These cells are then adjusted by tuning the initial hyperplanes from Fig. 6 (a) to Fig. 6 (b) to optimize a large margin classifier inspired by SVMs. The use of global hyper-planes and the large margin objective function allows the method to be generalized and prevents the training data's overfitting. Fig. 6. (b) illustrates the obtained LMPL piecewise linear decision boundary where three classes are classified well by five remaining hyperplanes. As presented, the decision boundaries are in a *piecewise linear* form that makes the LMPL *non-linearly separable* samples in the original space instead of using kernel methods. After adjusting hyperplanes, additional hyperplanes that are

Table 6

Comparison of without fusion and fusion methods with the proposed LMPL classifier in Experiment 1: Normal, COVID-19, and typical viral.

Method		Performance Metrics (%)				
		Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
Without Fusion	preRelu	98.64 (8)	98.81 (8)	98.72 (8)	98.52 (8)	8.0
	postRelu	98.87 (6)	99.08 (6)	98.97 (7)	98.81 (7)	6.5
Concat	Classical	99.22 (4)	99.33 (4)	99.28 (4)	99.13 (4)	4.0
	DCA	98.86 (7)	99.08 (6)	98.99 (6)	98.87 (6)	6.3
	CCA	99.37 (2)	**99.48 (1)**	99.42 (2)	99.32 (2)	1.8
Sum	Classical	99.27 (3)	99.38 (3)	99.33 (3)	99.20 (3)	3.0
	DCA	98.91 (5)	99.18 (5)	99.04 (5)	99.00 (5)	5.0
	CCA	**99.42 (1)**	99.47 (2)	**99.45 (1)**	**99.39 (1)**	**1.3**

*Bold numbers indicate the best performance.

Table 7

Comparison of without fusion and fusion methods with the proposed LMPL classifier in Experiment 2: COVID-19, SARS, and MERS pneumonia.

Method		Performance Metrics (%)				
		Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
Without Fusion	preRelu	95.74 (8)	96.03 (8)	95.88 (8)	96.29 (8)	8.0
	postRelu	96.64 (5)	96.29 (7)	96.46 (7)	96.86 (7)	6.5
Concat	Classical	97.94 (4)	96.97 (4)	97.44 (4)	97.72 (4)	4.0
	DCA	96.44 (7)	96.88 (5)	96.66 (5)	97.00 (5)	5.5
	CCA	98.66 (2)	97.87 (2)	98.26 (2)	98.43 (2)	2.0
Sum	Classical	98.10 (3)	97.39 (3)	97.74 (3)	98.00 (3)	3.0
	DCA	96.48 (6)	96.58 (6)	96.52 (6)	97.00 (5)	5.8
	CCA	**98.89 (1)**	**98.56 (1)**	**98.73 (1)**	**98.86 (1)**	**1.0**

*Bold numbers indicate the best performance.

Table 8

Comparison of different preprocessing methods in Experiment1: Normal, COVID-19, and typical viral.

Method	Performance Metrics (%)				
	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
No Preprocessing	96.09 (5)	95.89 (5)	95.99 (5)	96.40 (5)	5.0
Image Adjustment	97.50 (2)	97.57 (2)	97.52 (2)	97.52 (2)	2.0
Histogram Matching	96.93 (4)	96.64 (4)	96.78 (4)	97.01 (4)	4.0
Histogram Equalization	97.42 (3)	97.39 (3)	97.39 (3)	97.43 (3)	3.0
CLAHE	**99.42 (1)**	**99.47 (1)**	**99.45 (1)**	**99.39 (1)**	**1.0**

*Bold numbers indicate the best performance.

Table 9

Comparison of different preprocessing methods in Experiment 2: COVID-19, SARS, and MERS pneumonia.

Method	Performance Metrics (%)				
	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
No Preprocessing	89.77 (5)	91.60 (4)	90.62 (5)	92.15 (5)	4.75
Image Adjustment	97.57 (3)	96.63 (2)	97.09 (3)	97.57 (2)	2.5
Histogram Matching	94.78 (4)	91.19 (5)	92.80 (4)	93.72 (4)	4.25
Histogram Equalization	97.94 (2)	96.47 (3)	97.18 (2)	97.43 (3)	2.5
CLAHE	**98.89 (1)**	**98.56 (1)**	**98.73 (1)**	**98.86 (1)**	**1.0**

*Bold numbers indicate the best performance.

not useful in the classification are removed. Therefore, regarding not only the sample size but also the distribution and complexity of the decision boundaries, the complexity of the model is tuned by removing redundant hyperplanes. The large-margin approach forces the redundant hyperplanes to be removed. This structure not only makes it feasible to work with *small sample-size* problems but also prevents overfitting of the training data by using a large-margin approach for all the hyperplanes.

Fig. 7 represents classes by t-distributed Stochastic Neighbor Embedding (t-SNE) in 2-D and 3-D for both experiments. An extended version of this figure is shown in Fig. 8. A multiclass problem is shown with multi-modal small samples in each class. Class samples are segmented into parts. Hence, the class pairs may not be linearly

separable or they would have a very small margin if can be separated, whereas others can be piecewise linearly separated with acceptable margins.

Fig. 8 illustrates the obtained decision boundary of the introduced LMPL classifier on t-SNE embedding features with four initial hyperplanes. After a few iterations, the LMPL adjusts hyperplanes and removes additional hyperplanes. Moreover, piecewise linear decision boundaries are obtained that can separate classes. The initial number of hyperplanes is random (possibly 8 or 16), and the number of iterations can be determined manually (for example, 100). Algorithm 1 indicates the whole process of the proposed diagnostic method, and the steps of the introduced LMPL classifier are shown in Algorithm 2.

Table 10

Comparison of different augmentation techniques in Experiment1: Normal, COVID-19, and typical viral pneumonia.

Method	Performance Metrics (%)				
	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
No Augmentation	96.12 (5)	96.88 (5)	96.48 (5)	96.37 (5)	5.0
Brightness & Contrast	98.48 (3)	98.22 (4)	98.35 (3)	98.10 (4)	3.5
Cropping & Fliping	98.50 (2)	98.40 (2)	98.45 (2)	98.20 (2)	2.0
Gaussian & Salt and pepper noise	98.34 (4)	98.24 (3)	98.29 (4)	98.20 (2)	3.25
Rotation and Translation	**99.42 (1)**	**99.47 (1)**	**99.45 (1)**	**99.39 (1)**	**1.0**

*Bold numbers indicate the best performance.

Table 11

Comparison of different augmentation techniques in Experiment 2: COVID-19, SARS, and MERS pneumonia.

Method	Performance Metrics (%)				
	Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	Average Rank
No Augmentation	95.12 (5)	96.44 (5)	95.74 (5)	95.56 (5)	5.0
Brightness & Contrast	97.84 (3)	97.98 (3)	97.89 (2)	97.62 (3)	2.75
Cropping & Fliping	97.52 (4)	98.05 (2)	97.76 (3)	97.56 (4)	3.25
Gaussian & Salt and pepper noise	97.89 (2)	97.66 (4)	97.76 (3)	97.68 (2)	2.75
Rotation and Translation	**98.89 (1)**	**98.56 (1)**	**98.73 (1)**	**98.86 (1)**	**1.0**

*Bold numbers indicate the best performance.

Algorithm 1. The Proposed Diagnostic Method.

Algorithm 1. The Proposed Diagnostic Method
1. **Initialization:** Experiment 1: COVID-19, normal, viral pneumonia
2. Experiment 2: COVID-19, SARS, MERS
3. **Input:** X-ray images
4. **Image Preprocessing:**
5. Resizing images to 224×224 pixels
6. Image enhancement by CLAHE histogram equalization
7. Image augmentation by rotation (5° to 30°) and translation (%5 to %20)
8. **Feature Extraction:**
9. Fine-tuning pre-trained VGG-Net by transfer learning
10. Deep feature extraction by VGG-Net
11. **Deep Feature Fusion:**
12. Pre-ReLU and post-ReLU deep feature fusion by CCA
13. **Classification:**
14. **Training:**
15. Train the LMPL classifier on the training Set x by **Algorithm2**
16. **Testing:**
17. Classify test image x' by LMPL model
18. **Output:**
19. y' : Predicted class label of the test image

Algorithm 2. Large Margin Piecewise Linear (LMPL) Classifier.

Algorithm 2. Large Margin Piecewise Linear (LMPL) Classifier
1. **Initialization:** f // Number of iterations
2.
3. H // Set of hyper-planes
4. Cell-table   // Number of samples and cell-labels based on majority voting
5. **Inputs:** Training set x, H Hyperplane set, Test instance x'
6. **Training Process:**
7. **for** itr =1: f
8. **for each** h_i in H
9. Fix other hyperplanes
10. Optimize the objective function:
11. $\min_h \frac{1}{2}\|w\|^2 + \sum_x C_1 l(x)_{Normal(\bar{y})} + C_2 l(x)_{DontCare^-}$
12. Update cell-table
13. **end for**
14. Remove extra hyperplanes
15. **end for**
16. **Testing Process:**
17. Find the cell of x'
18. Get the final label of the cell as y'
19. **Output:**
20. y': The predicted class label for the test instance x'

4. Experimental results

4.1. Experimental setup

The 5-fold cross-validation (CV) was utilized. Of the original labeled data, 80% were used as the training set, and the remaining samples (20%) were employed as the test set to evaluate the model (unseen instances). Results were averaged to produce the final confusion matrix. MATLAB (2020a) was utilized for training, evaluating, and testing different algorithms. The experiments were run on a computer with an Intel Core i7 @4.0 GHz processor, 24 GB RAM, 128 GB SSD, and a 64-bit Windows 10 operating system. We used mini-batch gradient descent optimization with momentum update of $\beta = 0.9$ and learning rates of $\alpha = 1e^{-4}$ and $\alpha = 3e^{-4}$. Besides, we selected 16 images as the size of mini-batch and five back-propagation epochs. The details of the training and test instances in each fold can be found in the Zenodo repository (https://zenodo.org/record/4691987).

4.2. Experimental results and analysis

In this study, two experiments were carried out for two different classification problems:

- *Experiment1:* Normal, COVID-19, and typical viral pneumonia
- *Experiment2*: COVID-19, SARS, and MERS pneumonia
1) Performance analysis

Tables 2 and 3 compare several deep CNNs as an end-to-end network and as a feature extractor with the proposed LMPL (named CNN⁺). Experiments show that the proposed LMPL classifier performed better than the conventional SoftMax classifier in all CNNs. VGGNet and VGGNet⁺ were the best among the networks in both experiments. As illustrated in Tables 4 and 5, the proposed LMPL outperformed the conventional classifiers in terms of performance metrics in both
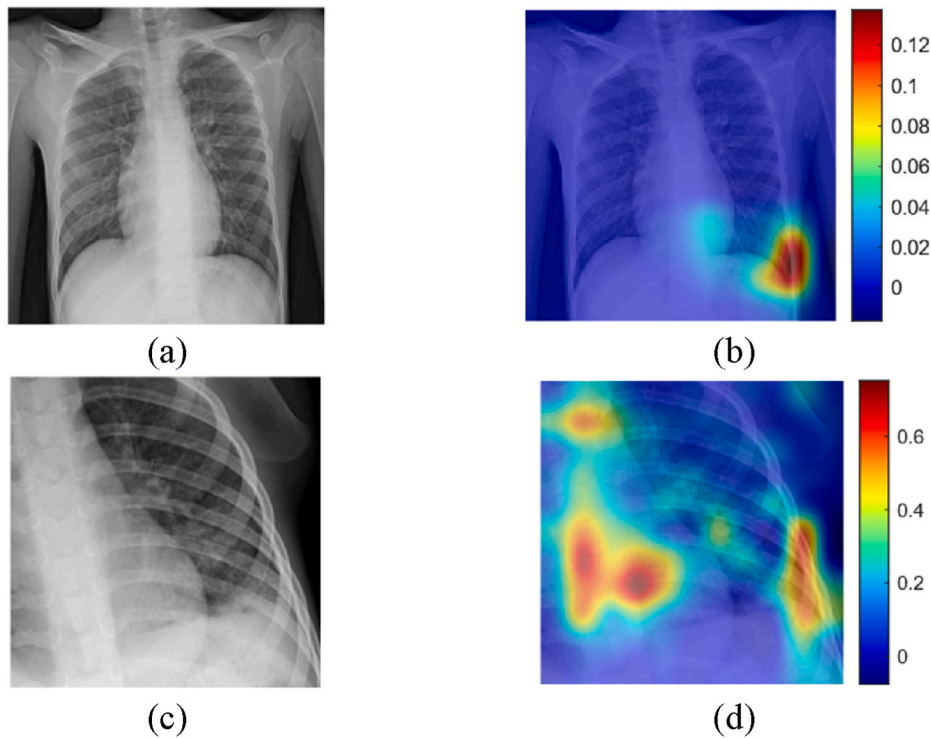
Fig. 9. False negative predictions in COVID-19 image classification (predicted as Viral pneumonia) and the corresponding heat maps: (a) first COVID-19 image (b) Heat map visualization of (a), (c) Second COVID-19 image, and (d) Heat map of (c).

Experiments 1 and 2, compared to ensemble methods, including Ada-boostM2, Total Boost, and Random Forests consisting of 100 trees as weak learners. Moreover, Tables 6 and 7 demonstrate the comparison of different feature fusion methods and pre-ReLU and post-ReLU deep features without fusion. Concatenation and summation techniques were used in Classical, CCA, and DCA feature fusion methods. The *sum* and *concat* operations generate 4096 and $4096 \times 2 = 8192$ final fused features, respectively.

The classical method is to group two sets of feature vectors which simply concatenates the two feature sets into a single vector. Discriminant Correlation Analysis (DCA) [38] maximizes the variance between classes, as well as the correlation among features through several feature sets.

We observed that regarding the performance metrics, the CCA fusion method with summation achieved outstanding results with 99.39% and 98.86% average accuracies in comparison with the other approaches.

Tables 8 and 9 compare the final performance of the proposed method with the other preprocessing techniques for both experiments. The effectiveness of different augmentation techniques in the final performance of the proposed method is compared in Tables 10 and 11 for both experiments using the following parameters.

- **Brightness:** Brightness is the amount of hue. As brightness varies from 0 to 1, colors change from black to white. Brightness jitter shifts the darkness and lightness of an input image. The brightness of the input image is adjusted by an offset selected randomly from the range $[0.1, 0.3]$. As expected, the image appears brighter with the rise in brightness.
- **Contrast:** The contrast jitter randomly adjusts the difference between the darkest and brightest regions in an input image. The contrast of the input image is adjusted by a scale factor selected randomly from the range $[1.2, 1.4]$. As the contrast increases, shadows become darker, and highlights become brighter.
- **Cropping:** The image is cropped to the target size 800×800 from the center of the image.

- **Flipping:** A reflection transformation is created that flips the input image in the left-right direction (X dimension) and top-bottom direction (Y dimension).
- **Gaussian noise:** A zero-mean, Gaussian white noise with a variance of 0.01 is added to the image.
- **Salt & pepper noise:** Salt and pepper noise is added with a noise density of 0.1. This affects almost 10% of the pixels.

2) Statistical analysis

The Friedman test was used to further analyze the experimental performance of the comparable methods statistically. It is a simple, nonparametric test and safe for comparing at least three related samples. The test has no assumptions about the primary distribution of data. It ranks the algorithms for each metric independently such that the algorithm with the highest performance is ranked first, the second-best is ranked second, and this procedure continues for the next ranks. R_j represents the average rank of the jth method depending on different metrics listed in Tables 2–11

In other words, R_j is calculated as follows

$$R_j = \frac{1}{n} \sum_{i=1}^{n} r_i^j \tag{10}$$

and r_i^j denotes the rank of the jth method on the ith metric. In the case of ties, i.e., when two algorithms perform similarly, the same ranks are assigned.

It can be seen from Tables 2–11 that the proposed method improved the performance significantly and achieved the best average rank in all cases. These tables show a significant difference in the effectiveness of the algorithms. Based on the Friedman test, the performance of the proposed MLPL classifier could be significantly better. In order to determine the best fusion method, the averaged ranks of different fusion techniques were calculated, as shown in Tables 6 and 7 As presented in the tables, CCA fusion techniques with summation improved the performance considerably and achieved the best rank in both experiments.

Table 12
Literature review of the state–of–the-art deep models using X-ray images on associated datasets (PNA stands for pneumonia).

Study	No. of cases	Method	Accuracy (%)
Ozturk et al. [5]	125 COVID-19	DarkCovidNet	
	500 No-finding		98.08
	125 COVID-19		
	500 Pneumonia		
	500 No-finding		87.02
Tabik et al. [6]	426 COVID-19	COVID-SD Net	97.72
	426 Normal		
Rahman et al. [8]	3619 COVID-19	ChexNet	96.29
	8851 Normal		
	6012 Pneumonia		
Toğaçar et al. [10]	295 COVID-19	MobileNetV, SVM	96.28
	65 Normal		
	98 Pneumonia		
Ioannis et al. [14]	224 COVID-19	VGG-19	93.48
	700 Pneumonia		
	504 Healthy		
Karakanis et al. [12]	275 COVID-19	ResNet, CGAN	
	275 Normal		98.7
	275 COVID-19		
	275 Normal		
	275 Bacterial PNA		98.3
Wang and Wong [13]	53 COVID-19	COVID-Net	92.4
	8066 Healthy		
Sethy and Behra [17]	25 COVID-19	ResNet 50, SVM	95.38
	25 No-finding		
Jain et al. [9]	COVID-19	Xception	97.97
	Normal		
	Pneumonia		
Jin et al. [15]	543 COVID-19	AlexNet	98.64
	600 Normal		
	600 Viral PNA		
Hemdan et al. [11]	25 COVID-19	COVIDX-Net	90.00
	25 No-finding		
Narin et al. [16]	50 COVID-19	Res-Net 50	98.00
	50 No-finding		
Mahmud et al. [7]	305 COVID-19	CovXNet	
	305 Normal		97.4
	305 COVID-19		
	305 Normal		
	305 Viral PNA		
	305 Bacterial PNA		90.3
Minaee et al. [18]	184 COVID-19	SqueezeNet	92.30
	5000 Non-COVID		
	6054 Pneumonia		
Abbaas et al. [28]	80 Covid-19	DeTraC	93.10
	105 Normal		
	11 SARS		
Tahir et al. [27]	423 COVID-19	InceptionV3	97.73
	134 SARS		
	144 MERS		
Proposed method (Transferred deep features of VGG-Net, deep feature fusion, the LMPL classifier)	423 COVID-19	VGG-Net19	
	1341 Normal		
	1345 Viral PNA		**99.39**
	423 COVID-19		
	134 SARS		
	144 MERS		**98.86**

Moreover, a comparison between preprocessing methods is performed in Tables 8 and 9 As can be seen, CLAHE obtained the first rank. The proposed augmentation techniques (rotation and translation) also showed the best effectiveness and achieved the best rank in comparison to other augmentation techniques (see Tables 10 and 11).

3) Visual analysis

We provided the occlusion sensitivity [39] that provides a detailed visual analysis of the extracted deep features through the fine-tuned VGG-Net, as shown in Fig. 9. Occlusion sensitivity is a technique to generate visual descriptions of the CNN model predictions. It provides an insight into the internal workings of a classifier based on deep CNNs.

Misclassifications or unpredicted results can be based on reasonable clarifications. Using heat maps, we investigated the prediction regions. False-negative examples were examined to find out the reasons underlying the hidden classification problem. The spatial parts that contributed most to false-negative predictions in COVID-19 classification were identified by occlusion sensitivity heat mapping, as shown in Fig. 9. The standard jet color-map is used where red and yellow denote regions with high contribution to the false-negative predictions, and blue indicates regions with a low contribution. Manual inspection showed that in these false-negative examples, the network incorrectly attended on some edges and corners of images that are not relevant to COVID-19.

The latest deep models using x-ray images (on not necessarily common datasets) are summarized in Table 12.

5. Discussion

LMPL improved the performance of all CNNs significantly and achieved the best results with a few iterations (five ones in our experiments) to avoid overfitting with small sample sizes and achieve better generalization. The LMPL outperformed conventional classifiers and provided very competitive results compared to ensemble methods with 100 trees as weak learners. However, the complexity of ensembles was high and needed space to train multiple models. The LMPL is simple and needs smaller space to produce a good model.

We can conclude that the fusion of deep features is helpful and improves the results even with classical concatenation and summation, which are fast and easy to apply. The obtained deep feature vectors were fused to acquire more informative features that minimized the impact of insufficient features obtained from one CNN model. Feature fusion efficiently produces features that comprise rich information describing the image well. Appropriately combining two or more features is not a trivial task. Due to similar visual features in various viral pneumonia, especially severe coronavirus of the SARS and MERS family, the DCA feature fusion strategy could not find useful information to maximize distances between classes.

However, x-rays are widely available and the most common approach as they are low-cost. CT scans are a more powerful method, so a chest CT scan might be needed in some cases to get a better picture and a more detailed view. Indeed, x-rays can help as a first-line diagnostic tool in most cases if they are detectable by this technology. In this study, we studied how much x-ray images could help to identify COVID-19, and we plan to extend our model for CT scans as well.

6. Conclusions and future work

Early COVID-19 detection can benefit preparing an appropriate treatment plan and facilitate medical decision-making. In this study, an LMPL classifier was presented for the diagnosis of coronavirus among a wide range of other viral pneumonia using raw chest x-ray images. The introduced method was shown to solve two classification problems: normal, COVID-19, and typical viral pneumonia; COVID-19, SARS, and MERS pneumonia. The results showed the outstanding average accuracy of the proposed method, equal to 99.39% and 98.86%, respectively, for both schemes, compared to state-of-the-art deep models.

In the future, the objective function of the introduced learning model can be extended to obtain clustering and semi-supervised models. To improve the LMPL classifier, positive don't care samples can be considered in tuning hyperplanes to get larger margins and better generalization of the model. However, such loss functions are not convex. Therefore, the solutions to deal with non-convex objective functions like gradient descent should be investigated. Other binary classifiers could be developed by the proposed cellular model. Applying kernels in the proposed classifier may help to consider complicated nonlinear decision boundaries. Extracting more effective features to make a more accurate and robust model for treating other emerging diseases with insufficient data can be considered in the future. The proposed pipeline can be adapted for chest CT scans to diagnose COVID-19 pneumonia, as well as other image processing applications of industrial and healthcare systems such as industrial cameras, process control, industrial robotics, and object recognition.

Declaration of competing interest

None Declared.

References

[1] E. Mahase, Coronavirus: Covid-19 Has Killed More People than SARS and MERS Combined, Despite Lower Case Fatality Rate, 2020.
[2] D.M. Ibrahim, N.M. Elshennawy, A.M. Sarhan, Deep-chest: multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases, Comput. Biol. Med. 132 (2021) 104348.
[3] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ArXiv Prepr. ArXiv1409. 1556 (2014).
[4] M. Haghighat, M. Abdel-Mottaleb, W. Alhalabi, Fully automatic face normalization and single sample face recognition in unconstrained environments, Expert Syst. Appl. 47 (2016) 23–34.
[5] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. (2020), 103792.
[6] S. Tabik, A. Gómez-Ríos, J.L. Martín-Rodríguez, I. Sevillano-García, M. Rey-Area, D. Charte, E. Guirado, J.L. Suárez, J. Luengo, M.A. Valero-González, COVIDGR dataset and COVID-SDNet methodology for predicting COVID-19 based on Chest X-Ray images, IEEE J. Biomed. Heal. Inform. 24 (2020) 3595–3605.
[7] T. Mahmud, M.A. Rahman, S.A. Fattah, CovXNet: a multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization, Comput. Biol. Med. 122 (2020), 103869.
[8] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S.B.A. Kashem, M. T. Islam, S. Al Maadeed, S.M. Zughaier, M.S. Khan, Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images, Comput. Biol. Med. 132 (2021), 104319.
[9] R. Jain, M. Gupta, S. Taneja, D.J. Hemanth, Deep learning based detection and analysis of COVID-19 on chest X-ray images, Appl. Intell. 51 (2021) 1690–1700.
[10] M. Toğaçar, B. Ergen, Z. Cömert, COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches, Comput. Biol. Med. 121 (2020), 103805.
[11] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar, Covidx-net: A Framework of Deep Learning Classifiers to Diagnose Covid-19 in X-Ray Images vol. 11055, 2020. ArXiv Prepr. ArXiv2003.
[12] S. Karakanis, G. Leontidis, Lightweight deep learning models for detecting COVID-19 from chest X-ray images, Comput. Biol. Med. 130 (2021), 104181.
[13] L. Wang, A. Wong, COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images, 2020. ArXiv Prepr. ArXiv2003.09871.
[14] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. (2020) 1.
[15] W. Jin, S. Dong, C. Dong, X. Ye, Hybrid ensemble model for differential diagnosis between COVID-19 and common viral pneumonia by chest X-ray radiograph, Comput. Biol. Med. 131 (2021), 104252.
[16] A. Narin, C. Kaya, Z. Pamuk, Automatic Detection of Coronavirus Disease (Covid-19) Using X-Ray Images and Deep Convolutional Neural Networks, 2020. ArXiv Prepr. ArXiv2003.10849.
[17] P.K. Sethy, S.K. Behera, Detection of coronavirus disease (covid-19) based on deep features, 2020, p. 2020. Preprints. 2020030300.
[18] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, G.J. Soufi, Deep-covid: predicting covid-19 from chest x-ray images using deep transfer learning, Med. Image Anal. 65 (2020), 101794.
[19] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi, Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks, Comput. Biol. Med. 121 (2020), 103795.
[20] A. Amyar, R. Modzelewski, H. Li, S. Ruan, Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: classification and segmentation, Comput. Biol. Med. 126 (2020), 104037.
[21] X. Qian, H. Fu, W. Shi, T. Chen, Y. Fu, F. Shan, X. Xue, M $^{\wedge}$ 3$ lung-sys: a deep learning system for multi-class lung pneumonia screening from CT imaging, IEEE J. Biomed. Heal. Inform. 24 (2020) 3539–3550.
[22] S. Serte, H. Demirel, Deep learning for diagnosis of COVID-19 using 3D CT scans, Comput. Biol. Med. 132 (2021), 104306.
[23] I. Shiri, M. Sorouri, P. Geramifar, M. Nazari, M. Abdollahi, Y. Salimi, B. Khosravi, D. Askari, L. Aghaghazvini, G. Hajianfar, Machine learning-based prognostic modeling using clinical data and quantitative radiomic features from chest CT images in COVID-19 patients, Comput. Biol. Med. 132 (2021) 104304.
[24] A.G. Dastider, F. Sadik, S.A. Fattah, An integrated autoencoder-based hybrid CNN-LSTM model for COVID-19 severity prediction from lung ultrasound, Comput. Biol. Med. 132 (2021) 104296.
[25] A. Hamimi, MERS-CoV: Middle East respiratory syndrome corona virus: can radiology be of help? Initial single center experience, Egypt, J. Radiol. Nucl. Med. 47 (2016) 95–106.
[26] X. Xie, X. Li, S. Wan, Y. Gong, Mining x-ray images of SARS patients, in: Data Min., Springer, 2006, pp. 282–294.
[27] A. Tahir, Y. Qiblawey, A. Khandakar, T. Rahman, U. Khurshid, F. Musharavati, S. Kiranyaz, M.E.H. Chowdhury, Coronavirus: Comparing COVID-19, SARS and MERS in the Eyes of AI vol. 11524, 2020. ArXiv Prepr. ArXiv2005.
[28] A. Abbas, M.M. Abdelsamea, M.M. Gaber, Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network, Appl. Intell. 51 (2021) 854–864.

[29] X. Xuanyang, G. Yuchang, W. Shouhong, L. Xi, Computer aided detection of SARS based on radiographs data mining, in: 2005 IEEE Eng. Med. Biol. 27th Annu. Conf., IEEE, 2006: pp. 7459–7462.

[30] S.M. Pizer, E.P. Amburn, J.D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J.B. Zimmerman, K. Zuiderveld, Adaptive histogram equalization and its variations, Comput. Vision, Graph, Image Process. 39 (1987) 355–368.

[31] Italian Society of Medical and Interventional Radiology | SIRM, (n.d.). https://www.sirm.org/en/italian-society-of-medical-and-interventional-radiolo gy/(accessed June 30, 2020).

[32] J.P. Cohen, P. Morrison, L. Dao, K. Roth, T.Q. Duong, M. Ghassemi, COVID-19 Image Data Collection: Prospective Predictions Are the Future, 2020. https://gith ub.com/ieee8023/covid-chestxray-dataset (accessed June 30, 2020).

[33] COVID-19 @ Radiopaedia.Org, (n.d.). https://radiopaedia.org/playlists/25975?la ng=us (accessed June 30, 2020).

[34] Thread by @ChestImaging: This Is a Thread of COVID-19 CXR (All SARS-CoV-2 PCR+) from My Hospital (Spain)., (n.d.). https://threadreaderapp.com/thr ead/1243928581983670272.html (accessed June 30, 2020).

[35] COVID-19 DATABASE | SIRM, (n.d.). http://www.sirm.org/category/senza-categ oria/covid-19/(accessed June 30, 2020).

[36] Covid-19, Sars, MERS X-Ray Images Dataset | Kaggle, (n.d.). https://www.kaggle. com/dataset/057e1b6dc41d9691e59dded4445fa8cc2f0b4b5cbcb49aef9583d95 233799d5a/version/2 (accessed June 27, 2020).

[37] Chest X-Ray Images (Pneumonia) | Kaggle, (n.d.). https://www.kaggle.com/pault imothymooney/chest-xray-pneumonia? (accessed June 27, 2020).

[38] Y. Bi, M. Lv, Y. Wei, N. Guan, W. Yi, Multi-feature fusion for thermal face recognition, Infrared Phys. Technol. 77 (2016) 366–374.

[39] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Eur. Conf. Comput. Vis., Springer, 2014, pp. 818–833.