

A Large Margin Piecewise Linear Classifier with Fusion of Deep Features in the Diagnosis of COVID-19

Neda Azouji¹, Ashkan Sami², Mohammad Taheri³, and Henning Müller⁴

Abstract— The world has experienced epidemics of coronavirus infections several times over the last two decades. Recent studies showed that using medical imaging techniques can be useful to develop an automatic computer-aided diagnosis system to detect pandemic diseases fast and early with high accuracy. In this study, a large margin piecewise linear classifier is 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 is employed, then deep pre- and post-rectified linear unit (ReLU) features are extracted using the well-known VGG-Net19, which is fine-tuned to optimize transfer learning. After this, the canonical correlation analysis is performed for feature fusion and fused deep features are 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.

Index Terms— Computer-aided diagnosis (CAD), COVID-19, Deep feature extraction, Large margin classifier, MERS, SARS, x-ray

I. INTRODUCTION

The COVID-19 virus is very contagious and spreads 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 in 2002-2003 originating in Guangdong, China;
- The Middle East Respiratory Syndrome (MERS) epidemic in 2011 originating in Jeddah, Saudi Arabia;
- 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 it tends to overfit the training data. We introduce a large margin piecewise linear (LMPL) classifier as a proper learning method to gain the best profit 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 (k NN). The proposed LMPL addresses the following challenges that are important in machine learning:

- *small sample size*: medical datasets are difficult to find;
- *challenging to collect* due to privacy constraints that limit data sharing;
- *overfitting to training data*: this problem limits the generality of models to unseen data. It is essential, particularly in transfer learning tasks in which the distribution of source and target can be so different;
- *nonlinearity*: most classes or clusters in real-world problems are not linearly separable;
- *multi-class classification*: some of the classifiers e.g., SVMs, have a binary structure and need extensions to solve this problem such as One versus One (OvO) and One versus All (OvA) approaches;
- *multi-modality*: when a class has several modalities, it means 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] is used for feature extraction. Indeed, the deep VGG-Net19 model is retrained by transfer learning to learn deep features of x-ray images. The deep features are obtained from the seventh fully connected layer that is the last layer before the output. We mark the activation values of both pre- and post-rectified linear units (ReLU) as deep features with feature vectors of dimension 4096. The study develops a preprocessing pipeline for quality improvement of x-ray images. For feature fusion, we used canonical correlation analysis (CCA) [4] and the fused features are passed to the LMPL classifier.

We aim to apply our introduced approach for two classification tasks in this study.

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

A. Contribution

The contributions of this paper are outlined as follows:

- To learn discriminant and robust feature representations, a novel COVID-19 diagnostic method is introduced that

¹ Neda Azouji, Department of Computer Science and Engineering and IT; School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran. Tel: +98-71-36133569 E-mail: azouji@shirazu.ac.ir.

² Ashkan Sami, Department of Computer Science and Engineering and IT; School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran. E-mail: sami@shirazu.ac.ir.

³ Mohammad Taheri, Department of Computer Science and Engineering and IT; School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran. E-mail: motaheri@shirazu.ac.ir.

⁴ Henning Müller, Department of Business Information Systems University of Applied Sciences Western Switzerland, Sierre (HES SO), Switzerland. E-mail: henning.mueller@hevs.ch

integrates a deep feature extraction and fusion methodology.

- To learn the most informative knowledge from limited labeled data, we introduced a LMPL classifier using a cellular structure to exploit the limited training data better.

II. RELATED WORK

Since the spread of COVID-19, there has been an increasing effort to advance deep models to COVID-19 detection using medical images such as CT scans and x-rays. In a review of the literature, we mention some of the recent developments in deep networks for COVID-19 diagnosis. Ozturk et al. [5] developed DarkCovidNet to diagnose COVID-19 using raw chest x-ray images. Their network reached an accuracy of 98.08% in a binary and 87.02% in a multi-class scheme. Taik et al. [6] introduced the COVIDGR-1.0 dataset and proposed the COVID-SDNet method that combines segmentation, data-augmentation and data transformation and 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) are used to train the proposed CovXNet. Then, Transfer learning is used to fine-tune and retrain the net by a smaller number of COVID-19 and other pneumonia cases. CovXNet is designed and trained by various image resolutions and a stacking algorithm is employed for optimization. CovXNet obtained an accuracy of 90.2% for multiclass classification 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. The author investigated the effect of different image enhancement techniques on COVID-19 detection. Various deep CNNs are trained on plain and segmented lung images. The method reached the best performance of 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: Inception V3, Xception, and ResNeXt where the Xception model had the best performance with the highest accuracy of 97.97% in classifying COVID-19 using chest x-ray images. Togaçar et al. [10] restructured the COVID-19 dataset by employing the Fuzzy Color technique as a preprocessing method and the images that were organized together with the original images were stacked. SqueezeNet and MobileNetV2 are 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 use 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 and also, 98.3% accuracy in a three-class model with normal, COVID-19, and bacterial pneumonia. Wang and Wong [13] developed an accurate deep network namely 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 98.75% in accuracy for the two-class problem and 93.48% in 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 classification accuracy in multi-class classification, 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 reached promising results in several tasks. Moreover, there are also various recent studies on COVID-19 detection that applied several deep learning models with CT images [2,19–23] and Lung Ultrasound (LUS) [24]

A few studies focused on the diagnosis of SARS and MERS in X-ray chest images: In the work of Hamimi [25] about MERS, it was shown 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 identify SARS from typical pneumonia from 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 pre-processing technique and deep learning algorithms and reported outperforming four CNN methods: SqueezeNet, ResNet18, Inceptionv3, and DenseNet201. They achieved the best accuracy of 98.22% using InceptionV3. Abbas et al. [28] proposed a method based on DeTraC deep CNN to classify COVID-19 using chest x-ray images. The DeTraC model consists of three phases, Decomposition, Transfer, and Compose. 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 includes 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, where the regression trees achieved the highest detection accuracy. However, for pneumonia classification, NN-based models produce reasonably good results.

As presented in this section, researchers worldwide introduced several approaches to COVID-19 detection and also techniques applied to identify SARS or MERS pneumonia. Since features of the lung infections are the same in these diseases, it is problematic for the expert to distinguish them, so

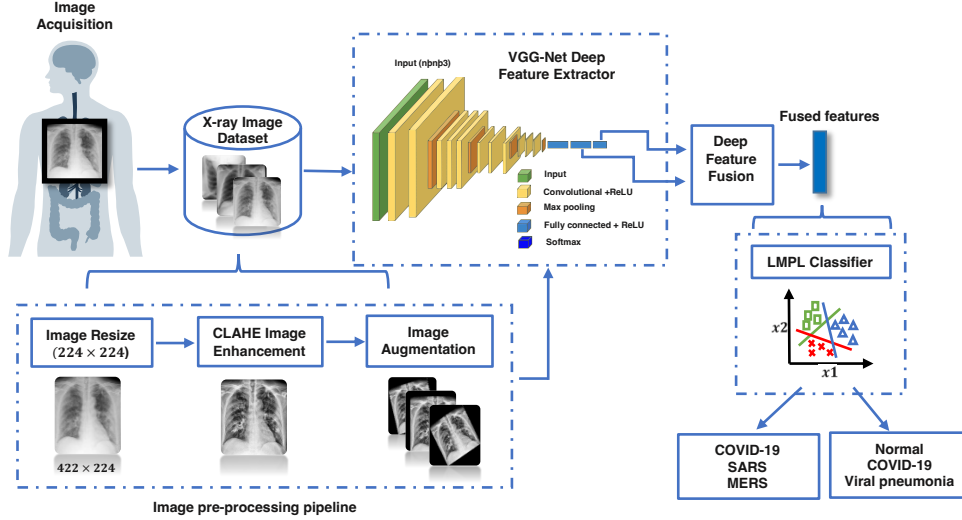


Fig. 1. Schematic of the overall proposed diagnostic system

well-trained models are needed to develop an accurate and useful method in the treatment and decision-making. In this study, we advanced an effective method in the diagnosis of current COVID-19 from a wide range of viral pneumonia including, the same CoV family members SARS and MERS.

III. MATERIAL AND METHODS

We present the details of our proposed COVID-19 diagnostic method. After pre-processing 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. We introduce LMPL classifier for coronavirus detection. Fig. 1 illustrates a schema of the proposed COVID-19 diagnostic model.

A. 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. The details of the dataset can be seen in Table I. The total number of images of the dataset is 3387.

TABLE I
DETAILS OF 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]

The dataset was balanced by image augmentation to reach 6179 images, as mentioned in *Sec. III. 3) Image Augmentation*.

B. Image pre-processing pipeline

Medical images often contain a low dynamic range. This affects the overall performance of image processing approaches. So, it may become harder to evaluate them visually

[8]. We perform pre-processing to enhance the image information and visual quality through implementing contrast enhancement. Another critical issue in medical imaging is imbalanced data, where the number of positive cases is lower than the negative cases.

1) Resizing X-ray images

To feed our images into CNNs, we need to resize the images to the input size dimensions of the deep network. For VGG-Net, images were resized to 224×224 pixels.

2) Contrast limited adaptive histogram equalization

Histogram equalization (HE) is an ordinary method and widely used to 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 the image by spreading out the most common intensity values of pixels in the range of 0 (black) to 255 (white). 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 CLAHE method. It can be seen that the histogram is stretched around the entire spectrum and over all pixels of the image, this should facilitate the extraction of

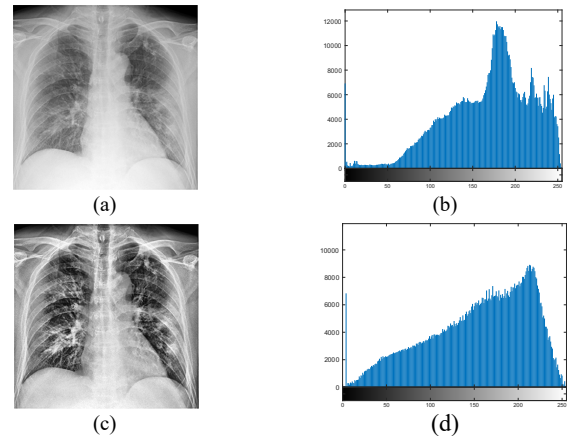


Fig. 2. CLAHE image enhancement. (a) COVID-19 X-ray image, (b) Histogram of (a), (c) CLAHE enhanced COVID-19 image, and (d) Histogram of (b).

homogeneous features across all datasets.

3) Image Augmentation

To extend the number of training samples and avoid the risk of overfitting, we augment our data by applying different types of transformations. This makes the dataset suitable for deep learning. Deep learning algorithms such as CNNs generally outperform on a larger dataset rather than a smaller one. Besides this, 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) in order to balance the class instances with augmentation. Images were rotated 5, 10, 15, and 30 degrees and instances were also translated in a horizontal or vertical direction by 5%, 10%, 15% in the SARS and MERS classes that have few instances, 134 and 144 images, respectively. Finally, 1072 and 1152 images were obtained in the SARS and MERS classes, respectively. In the COVID-19 class with 423 instances, 15-degree rotation and 10% translation were performed to reach 1269 images in this class. No augmentation was 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>

C. Deep Feature Extraction

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

The input image size in VGG networks is $224 \times 224 \times 3$. The architectural overview of VGG-Net is illustrated in Fig. 4. When applying VGG-Net as a deep feature extraction network, we remove the classification layer, which means the last fully connected layer. We can extract features as raw values or after transforming by a ReLU, where an output x is mapped to $\max(0, x)$. The activation values from the last hidden layer are extracted as the first deep feature vector (preReLU features, 4096) as well as, the post-ReLU feature set (postReLU features, 4096) as the second feature vector. For datasets that are not large enough, we can use transfer learning. In this paper, fine-

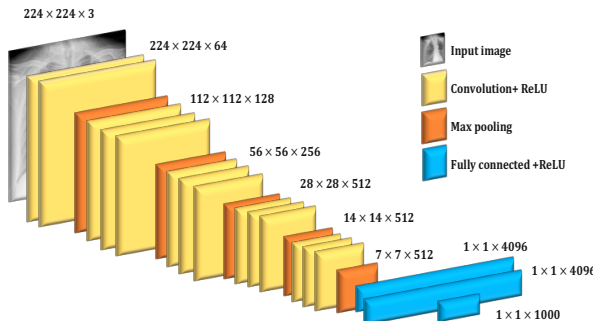


Fig. 4. VGG-Net architecture

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 image and by 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 diagnose COVID-19 vs. other viral pneumonia.

D. 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 that describe 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 is used, CCA. Feature fusion based on CCA creates two sets of transformations based on the correlation between two feature vectors that results 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 $S_{xy} \in R^{m \times k}$ is the between-sets covariance matrix where, $S_{yx} = S_{xy}^T$. The overall covariance matrix $S = R^{(m+k) \times (m+k)}$ can be presented by (1):

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

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

$$X^* = W_x^T X, Y^* = W_y^T Y \quad (2)$$

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

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

where, $cov(X^*, Y^*) = W_x^T S_{xy} W_y$, $var(X^*) = W_x^T S_{xx} W_x$ and $var(Y^*) = W_y^T S_{yy} W_y$. The combination of the transformed

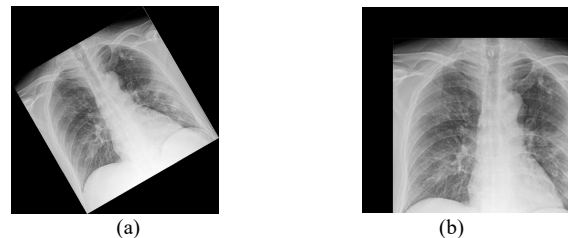


Fig. 3. Image augmentation. (a) Image rotation by 30 degrees counter-clockwise, and (b) Image translation in the horizontal and vertical direction by 15%.

features is achieved by addition (summation) or concatenation as follows in (4) and (5).

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

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

where, the resulted 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 image.

E. Large Margin Piecewise Linear (LMPL) Classifier

In the last stage, fused features are used for the classification task by the proposed LMPL classifier. A cellular structure is developed, in this study, such that space is partitioned by more than one hyper-plane, instead of just one as in 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 hyper-planes 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 which are bounded 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 as 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 will be 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, in (6):

$$l(\mathbf{x})_{Normal\tilde{y}} = \max(0, 1 - \tilde{y}(\mathbf{w}^T \cdot \mathbf{x} + b))$$

where: $\tilde{y} = \{-1, +1\}$ (6)

Where \tilde{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. So, it is not important on which side they are located. Nevertheless, it is tested to change the position of samples concerning 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 due to the fact that they are misclassified.

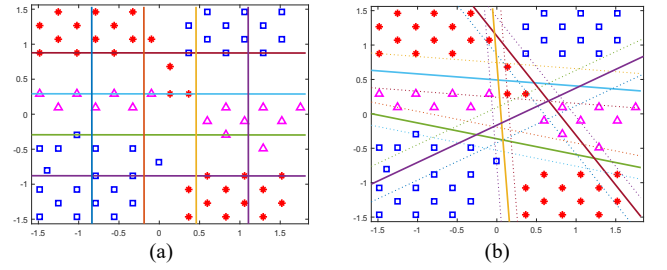


Fig. 6. An example of nonlinear multi-class classification with the proposed LMPL classifier. (a) The initial grid structure is constructed by eight initial hyperplanes, and (b) The final piecewise linear decision boundary is obtained by the LMPL classifier.

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

3. *Positive Don't care*: these samples are classified correctly on both sides of the hyperplane. Such as the second group (Negative Don't care), it is not important which side of the hyperplane they are on. But in this case, it is tested to maximize their distance to the hyper-plane. The total loss is defined by (8):

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

The *Positive Don't care group* is discarded here, since they 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):

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + \left(C_1 \sum_{x \in Normal} l(\mathbf{x})_{Normal(\tilde{y})} \right) + \left(C_2 \sum_{x \in DC^-} l(\mathbf{x})_{DontCare^-} \right) \quad (9)$$

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 will be removed.

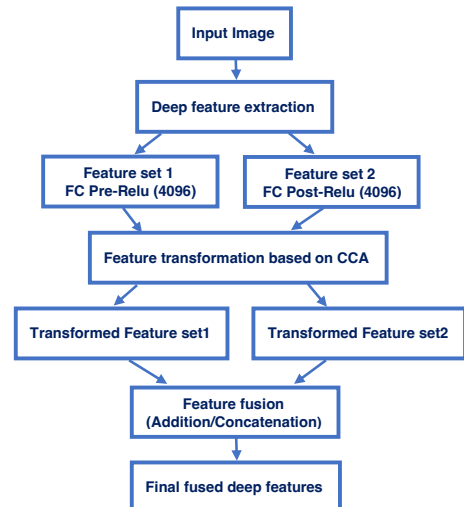


Fig. 5. canonical correlation analysis (CCA) feature fusion

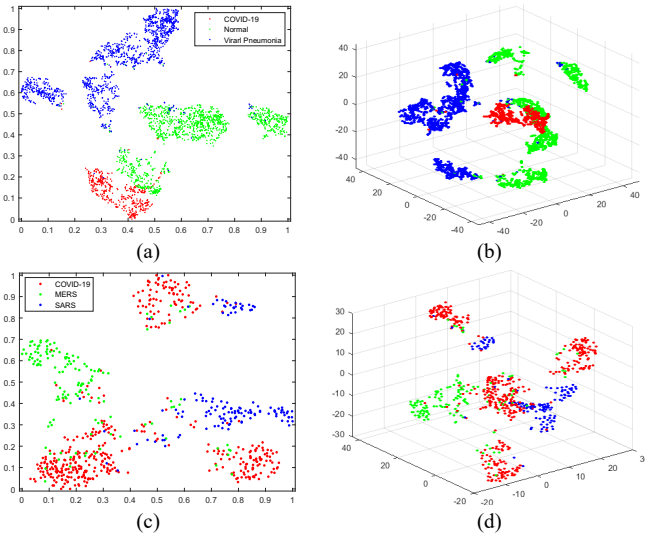


Fig. 7. Representation of classes by t-SNE Embedding. (a) Scatter plot in the two-dimensional space of Experiment1: Normal, COVID-19, and typical viral pneumonia, (b) 3D scatter plot of (a), (c) Scatter plot in the two-dimensional space of Experiment2: COVID-19, SARS, and MERS. and (d) 3D scatter plot of (c).

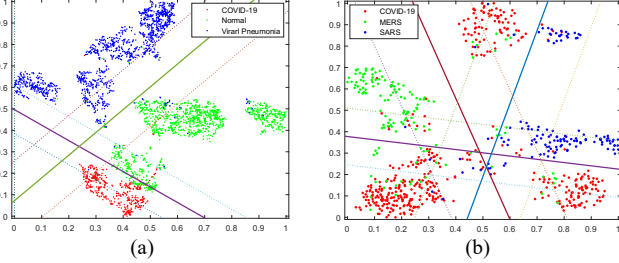


Fig. 8. Representation of the LMPL decision boundary on t-SNE embedding features with four initial hyperplanes (a) two remaining hyperplanes of Experiment1: Normal, COVID-19, and Typical Viral Pneumonia, and (b) three remaining hyperplanes in Experiment2: COVID-19, SARS, and MERS.

After several iterations, the final piecewise linear decision boundary is constructed. In the test step, an unseen sample x is classified according to its cell label. Cell labels were stored in the obtained cell-table in the training phase.

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 *multi-class* 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 in order 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. Both using global hyper-planes and the large margin objective function lead to consider *generalization* and preventing the *overfitting* on the training data. 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 not

useful in the classification will be 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 it also prevents overfitting on the training data by use of all the hyperplanes with a large-margin approach.

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

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 Algorithm 2
16. **Testing:**
17. Classify Test Image x' by LMPL Model
18. **Output:**
19. y' : Predicted Class Label of Test Image

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-label based on majority voting
5. **Input:** 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' which is the predicted class label for the test instance x'

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

5-fold cross-validation (CV) is utilized. 80% of the original labeled data are used as the training set and the remaining samples (20%) are employed as the test set to evaluate the model (unseen instances). Results are 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© i7-core @4.0GHz processor, 24GB RAM, 128 GB SSD, and 64-bit Windows 10 operating system. We use mini-batch gradient descent optimization with momentum update, $\beta = 0.9$, and learning rate, $\alpha = 1e^{-4}$ 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>

B. 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*

Several deep CNNs were compared as an End-to-End network and as a feature extractor with the proposed LMPL (named CNN⁺) in Table II and Table III. Experiments show the proposed LMPL classifier performed better than the conventional SoftMax classifier in all CNNs. VGGNet and VGGNet⁺ were best among the networks in both experiments. As illustrated in Table IV and Table V, the proposed LMPL outperformed the conventional classifiers in terms of performance metrics in both Experiment 1 and Experiment 2 compared to ensemble methods including AdaboostM2, Total Boost, and Random Forests consisting of 100 trees as weak learners. Moreover, the comparison of different feature fusion methods and deep features without fusion means preReLU and postReLU demonstrated in Table VI and Table VII. Concatenation and Summation techniques were used in Classical, CCA, and DCA feature fusion methods. The *sum* and *concat* operations, generate 4096 and 4096*2=8192 final fused features, respectively.

TABLE II

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

*Bold numbers indicate the best performance

TABLE III

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	Feature Extractor Layer	Performance Metrics (%)				Average Rank
		Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	
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		95.93	95.73	95.81	95.58	
Avg on CNN⁺		97.76	97.80	97.77	97.51	

Method	Feature Extractor Layer	Performance Metrics (%)				Average Rank
		Sensitivity (Recall)	Precision (PPV)	F1-score	Accuracy	
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		85.88	84.38	84.50	86.75	
Avg. on CNN⁺		93.34	91.32	92.04	93.20	

*Bold numbers indicate the best performance

TABLE IV

COMPARISON OF COMMON 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 V

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

TABLE VI

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	preRelu	98.64 (8)	98.81 (8)	98.72 (8)	98.52 (8)	8.0
Fusion	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 VII

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	preRelu	95.74 (8)	96.03 (8)	95.88 (8)	96.29 (8)	8.0
Fusion	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

TABLE VIII

COMPARISON OF DIFFERENT PREPROCESSING METHODS IN EXPERIMENT 1: 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 IX

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

TABLE X

COMPARISON OF DIFFERENT AUGMENTATION TECHNIQUES IN EXPERIMENT 1: 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 & Flipping	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 XI

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 & Flipping	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

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

We observed that regarding the performance metrics the CCA fusion method with summation achieved outstanding results in comparison with the other approaches with 99.42% and 98.89% averaged accuracies.

The final performance of the proposed method is compared with the other preprocessing techniques as shown in Table VIII and Table IX for both experiments. The effectiveness of different augmentation techniques on the final performance of the proposed method is compared as shown in

Table X and Table XI for both experiments. The comparing augmentation techniques are used with parameters as follow,

- **Brightness:** Brightness is the amount of hue. As brightness varies from 0 to 1, colors go 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]. The image appears brighter, as expected when the brightness increases.
- **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]. The contrast increases, such that 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 up to down direction (Y dimension).
- **Gaussian Noise:** A zero-mean, Gaussian white noise with 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 approximately 10% of pixels.

2) Statistical Analysis

To further analyze the experimental performance of the comparable methods statistically, the Friedman test was used. The Friedman test is a simple, nonparametric test and safe for comparing at least three related samples. It has no assumptions about the primary distribution of data. The Friedman test ranks the algorithms for each metric independently such that the algorithm with the highest performance reaches the rank 1, the second-best is ranked as 2, etc. R_j represents the average rank of the j^{th} method rely on different metrics, listed in Table II to XI.

In other words, R_j calculated as follow,

$$R_j = \frac{1}{n} \sum_{i=1}^n r_i^j \quad (10)$$

and r_i^j denotes the rank of j^{th} method on the i^{th} metric. In the case of ties, meaning two algorithms perform similarly, the same ranks are assigned.

It can be seen from Table II to XI that the proposed method improved the performance significantly and achieved the best average rank in all cases. These tables show a meaningful difference through the efficiency of algorithms. Based on the Friedman test, the performance of the proposed MLPL classifier could be significantly better. To determine the best fusion method, the averaged ranks of different fusion techniques are calculated, as shown in Table VI and Table VII. As presented in Table VI and Table VII, CCA fusion techniques with summation improved the performance considerably and achieved the best rank in both experiments.

Moreover, a comparison between preprocessing methods is conducted in Table VIII and Table IX where CLAHE obtained the first rank, also the proposed augmentation techniques (Rotation and Translation) show the best effectiveness and achieved the best rank in comparison to other augmentation techniques (see Table X and Table XI).

3) Visual Analysis

For a detailed visual analysis, we provided the occlusion sensitivity [39] that provides a detailed visual analysis visualization of the extracted deep features through the fine-tuned VGG-Net as shown in Fig. 8. Occlusion sensitivity is a technique to generate visual descriptions of the CNN model predictions. It helps to provide 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. 8. The standard jet color-map is used where red and yellow denote high contribution regions to the false-negative predictions, and blue indicates low contribution regions. 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.

Furthermore, the latest deep models using X-ray images (on not necessarily common datasets) are summarized in Table XII.

Study	No. of cases	Method	Accuracy (%)		
Ozturk et al. [5]	125 COVID-19	DarkCovidNet	98.08		
	500 No-finding				
	125 COVID-19				
	500 Pneumonia				
Tabik et al. [6]	426 COVID-19	COVID-SD Net	97.72		
	426 Normal				
	3619 COVID-19			ChexNet	96.29
	8851 Normal				
6012 Pneumonia					
Togaçar 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	98.7		
	275 Normal				
	275 COVID-19			98.3	
	275 Normal				
275 Bacterial PNA					
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 Normal Pneumonia	Xception	97.97
Jin et al. [15]	543 COVID-19 600 Normal 600 Viral PNA	AlexNet	98.64
Hemdan et al. [11]	25 COVID-19 25 No-finding	COVIDX-Net	90.00
Narin et al. [16]	50 COVID-19 50 No-finding	Res-Net 50	98.00
Mahmud et al. [7]	305 COVID-19 305 Normal 305 COVID-19 305 Normal 305 Viral PNA 305 Bacterial PNA	CovXNet	97.4 90.3
Minaee et al. [18]	184 COVID-19 5000 Non-COVID 6054 Pneumonia	SqueezeNet	92.30
Abbaas et al. [28]	80 Covid-19 105 Normal 11 SARS	DeTraC	93.10
Tahir et al. [27]	423 COVID-19 134 SARS 144 MERS	InceptionV3	97.73
Proposed Method (Transferred deep features of VGG-Net, Deep feature fusion, The LMPL classifier)	423 COVID-19 1341 Normal 1345 Viral PNA 423 COVID-19 134 SARS 144 MERS	VGG-Net19	99.42 98.89

V. DISCUSSION

LMPL improved the performance of all CNNs significantly and achieved the best results with few iterations (5 in our experiments) to avoid overfitting with small sample sizes and leads to 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 is high and need space to train multiple models. The LMPL is simple and needs less 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 easy to apply and fast. The obtained deep feature vectors are fused to acquire more informative features that minimize the impact of insufficient features obtained from one CNN model. Feature fusion efficiently produces features that comprise rich information that describe 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 family SARS and MERS, the DCA feature fusion strategy could not find useful information to maximize between-class distances.

Although, x-rays are widely available and the most common approach as they are low-cost. CT scans are a more powerful method, so it might be needed in some cases to do a chest CT 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 can help to identify COVID-19, and we plan to extend our model for CT scans as well.

VI. CONCLUSION AND FUTURE WORK

Early COVID-19 detection can benefit in preparing an appropriate treatment plan and facilitate medical decision-making. In this study, a LMPL classifier was presented in the diagnosis of coronavirus from 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 show an outstanding average accuracy of the proposed method, as compared to state-of-the-art deep models, for both schemes with 99.4% and 98.9%, respectively.

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 to treat 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, object recognition, etc.

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. Haghigat, 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. Charre, 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. Informatics.* 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. Zughair, 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, *ArXiv Prepr. ArXiv2003.11055*. (2020).
- [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, *ArXiv Prepr. ArXiv2003.09871*. (2020).
- [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, *ArXiv Prepr. ArXiv2003.10849*. (2020).
- [17] P.K. Sethy, S.K. Behera, Detection of coronavirus disease (covid-19) based on deep features, *Preprints. 2020030300* (2020) 2020.
- [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. S³ Lung-Sys: A Deep Learning System for Multi-Class Lung Pneumonia Screening From CT Imaging, *IEEE J. Biomed. Heal. Informatics.* 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, *ArXiv Prepr. ArXiv2005.11524*. (2020).
- [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-radiology/> (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://github.com/ieee8023/covid-chestxray-dataset> (accessed June 30, 2020).
- [33] COVID-19 @ Radiopaedia.org, (n.d.). <https://radiopaedia.org/playlists/25975?lang=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/thread/1243928581983670272.html> (accessed June 30, 2020).
- [35] COVID-19 DATABASE | SIRM, (n.d.). <http://www.sirm.org/category/senza-categoria/covid-19/> (accessed June 30, 2020).
- [36] COVID-19, SARS, MERS X-ray Images Dataset | Kaggle, (n.d.). <https://www.kaggle.com/dataset/057e1b6dc41d9691e59dded4445fa8cc2f0b4b5cbcb49aef9583d95233799d5a/version/2> (accessed June 27, 2020).
- [37] Chest X-Ray Images (Pneumonia) | Kaggle, (n.d.). <https://www.kaggle.com/paultimothymooney/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.