

AdaD-FNN for chest CT-based COVID-19 diagnosis

Xujing Yao, Ziquan Zhu, Cheng Kang, Shui-Hua Wang, *SMIEEE*, Juan Manuel Gorriz, Yu-Dong Zhang, *SMIEEE*

Abstract—Coronavirus disease 2019 (COVID-19) generated a global public health emergency since December 2019, causing huge economic losses. To help radiologists strengthen their recognition of COVID-19 cases, we developed a computer-aided diagnosis system based on deep learning to automatically classify chest computed tomography-based COVID-19, Tuberculosis, and healthy control subjects. Our novel classification model AdaD-FNN sequentially transfers the trained knowledge of an FNN estimator to the next FNN estimator while updating the weights of the samples in the training set with a decaying learning rate. This model inhibits the network from remembering the noisy information and improves the learning of complex patterns in the hard-to-identify samples. Moreover, we designed a novel image preprocessing model F-U2MNet-C by enhancing the image features using fuzzy stacking and eliminating the interference factors using U2MNet segmentation. Extensive experiments are conducted on four publicly available datasets namely, TLDCA, UCSD-AI4H, SARS-CoV-2, TCIA, and the obtained classification accuracies are 99.52%, 92.96%, 97.86%, 91.97%. Our novel system gives out compelling performance for assisting COVID-19 detection when compared with 22 state-of-the-art methods. We hope to help link together biomedical research and artificial intelligence and to assist the diagnosis of doctors, radiologists, and inspectors at each epidemic prevention site in the real world.

Index Terms—deep learning; ensemble models; convolutional neural network; COVID-19; fractional pooling; transfer learning

I. INTRODUCTION

CORONAVIRUS disease 2019 (COVID-19) is an ongoing pandemic disease caused by the SARS-CoV-2 virus. It rapidly swept worldwide, generating a global public health emergency within merely one month [1, 2]. According to the statistical data from World Health Organization, as of December 23, 2021, a total of 276,436,619 COVID-19 cases have been confirmed, resulting in 5,374,744 deaths. As the outbreak spread, many countries were affected, and a great deal of effort was devoted to the anti-epidemic project. However, many cases have been spread in communities with no history of travel to the outbreak area or contact with infected people. In such cases, health care workers need the highly sensitive COVID-19 diagnostic tool to ensure that every case is not missed, especially in those with false-negative

reverse transcriptase-polymerase chain reaction (RT-PCR) [3].

Chest computed tomography (CCT) examination is regarded as a useful mainstream auxiliary technique for the diagnosis of viral pneumonia cases associated with COVID-19 and was included in the national diagnostic treatment protocol (7th edition) of China [4]. When compared to those biomedical-based methods such as nucleic acid testing using real-time RT-PCR, CCT owns many outstanding advantages: (i) Normal nucleic acid testing process can take days to get results [5], while CCT approaches can quickly report back in minutes. (ii) Although nucleic acid testing is considered the ‘gold standard’ for clinical diagnosis, the problem of false-negative has been reported [6]. This limitation can be supplemented with the reported high sensitivity of CCT in diagnosing COVID-19 [7, 8]. (iii) From an environmental point of view, CCT approaches reduce the consumption of materials, like swabs, paper boxes, plastic bags, etc., which to some degree relieve the possible pollution problems. In addition, CCT allows more precise visualization of extremely small nodules in the lung area compared with other chest imaging approaches such as chest X-ray and chest ultrasound approaches [9].

In the epidemic situation, the problem of labor intensity, work efficiency, and emotional fluctuation of radiologists are all significant factors that need consideration. A limitation of CCT is it may share certain similar imaging features between COVID-19 and other categories of chest diseases, thus making it difficult for radiologists to differentiate [8]. Besides, the presence of faintly ground glass in the early lung lesions may lead to missed diagnosis by radiologists [10], especially in the follow-up work of COVID-19 detection that requires a large number of slides reading. To help relieve this problem and strengthen the radiologists’ recognition of the lesion features, we proposed a diagnostic system based on deep learning, which has made remarkable progress for automated diagnosis in the real world to automatically screen lesions, analyze and generate reports [11]. Radiologists can utilize the rapidly generated information to make more credible judgments based on the overall screening, improving the detection rate of lesions, work efficiency, and reducing the possibility of missed diagnosis [12].

Our study intends to improve the recognition performance of

Manuscript received February; revised; accepted. Date of publication November; date of current version

This paper is partially supported by Hope Foundation for Cancer Research, UK (RM60G0680), Royal Society International Exchanges Cost Share Award, UK (RP202G0230), Medical Research Council Confidence in Concept Award, UK (MC_PC_17171), British Heart Foundation Accelerator Award, UK (AA/18/3/34220), Sino-UK Industrial Fund, UK (RP202G0289), Global Challenges Research Fund (GCRF), UK (P202PF11), LIAS Pioneering Partnerships award, UK (P202ED10), Data Science Enhancement Fund, UK (P202RE237), MCIN/AEI (10.13039/501100011033); Consejería de Economía, Innovación, Ciencia y Empleo; MINECO/JUNTA/FEDER, Spain/Regional/Europe (RTI2018-098913-B100, CV20-45250, A-TIC-080-

UGR18, B-TIC-586-UGR20, P20-00525). (Corresponding Author: Yu-Dong Zhang)

X. Yao, S. Wang, Y. Zhang are with the School of Computing and Mathematical Sciences, University of Leicester, Leicester, UK (e-mails: xy147@leicester.ac.uk, shuihuawang@ieee.org, yudongzhang@iee.org).

Z. Zhu is with Science in Civil Engineering, University of Florida, Gainesville, Florida, FL 32608, USA (e-mail: zhu.ziquan@ufl.edu)

C. Kang is with the Department of Cybernetics and Robotics, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic (e-mail: kangchen@fel.cvut.cz).

J. Gorriz is with the Department of Signal Theory, Networking and Communications, University of Granada, Granada, 52005, Spain (e-mail: gorriz@ugr.es)

COVID-19 infection in CCT images by developing a novel deep neural network ‘Adaboosting-Decay Fractional max-pooling Neural Network (AdaD-FNN)’ for classification and a novel image preprocessing model ‘Fuzzy stacking enhancement U2MNet for CCT images (F-U2MNet-C)’. Our contributions and findings entail the following five angles:

(i) Inspired by the architecture of U2-Net, an advanced version U2MNet is proposed. Although the U2-Net architecture provides rich information when the training set is sufficient, the boundary ambiguity problem about the regions of interest may exist when encountering a size-limited dataset. This paper demonstrates that the incorporation of maximum entropy threshold segmentation and erosion&dilation refining approach can effectively minimize this loss of boundary information.

(ii) A novel CCT image preprocessing model, F-U2MNet-C is designed by using the fuzzy stacking method for the feature enhancement and the redesigned U2MNet for the interference factors elimination. Compared with traditional preprocessing models, this model shows stronger universality by relieving the block effect at the boundary of sub-images, the difference in enhancement levels shown in adjacent areas, and the boundary ambiguity problem brought by the size-limited dataset.

(iii) A novel AdaD-FNN model is proposed for multi-class classification. The performance is evaluated on four public datasets with 22 state-of-the-art methods.

(iv) In the base estimator of AdaD-FNN, we used fractional max-pooling to replace max-pooling and average-pooling to further improve the network recognition capability.

(v) The proposed AdaD-FNN consists of the multi-learning-mode base estimators. Instead of simply introducing the decaying learning rate in a single estimator, we incorporated a *lrdecay* module inside the AdaBoosting structure to further help optimization and generalization.

These five improvements can help enrich the performance of our model specially built for COVID-19 detection. Sections II–V will introduce the related work, methodology, discussions of experimental results, and the conclusions of our study.

II. RELATED WORK

Modern machine learning and deep learning technology have achieved a plethora of contributions when dealing with COVID-19 classification tasks from analyzing CCT images. Among these studies, research direction can be divided into single model-based classification models, ensemble learning-based classification models and image preprocessing models.

A. single model-based classification models: Yu, et al. [13] gave the first attempt to integrate graph convolutional neural (GCN) network into COVID-19 detection. They constructed a graph based on the Euclidean distance between features extracted by the proposed ResNet101-C and then encoded the graphs with the features to output the final predicted results. The Laplacian smoothing in graph convolution can aggregate the features of nearest neighbour nodes and make the features of nodes from the same class similar, which is suitable for classification problems. However, when encountering a size-limited dataset, deep GCNs are prone to the phenomenon of over-smoothing, which makes the output features of nodes excessively smooth and difficult to be distinguished. To make better use of the information in a size-limited dataset, Scarpiniti,

et al. [14] trained a deep denoising convolutional autoencoder and created a robust statistical representation by evaluating the histogram of the hidden features. Transfer learning-based approaches [15, 16] are also good choices for small datasets. [17-21] fine-tuned the pre-trained models to extract distinct features from images, and further fed the resulting feature maps into appropriate classifiers. These rapid frameworks usually achieved promising results. But when the source task and target task are not sufficiently correlated, or the transfer learning does not make good use of the relationship between the source task and target task, a negative transfer phenomenon will occur, resulting in performance degradation. Thus transfer learning-based works perform not well on the heterogeneous dataset. To develop a network tailored for COVID-19 detection, Wang, et al. [22] presented COVID-Net by introducing a lightweight projection-expansion-projection-extension design, which enables enhanced representation capacity while reducing computational complexity. In its redesign study [23], a joint learning framework was proposed through conducting separate feature normalization and constructing the contrastive objective. However, previous works on other joint learning applications [24, 25] have observed that straight-forward joint learning brings limited improvement with heterogeneous datasets and may underperform when trained on a single dataset.

B. ensemble learning-based classification models: Ensemble learning is a training concept of constructing multiple basic classifiers and ensembling them into a more powerful classifier to make the final decision [26, 27], which is widely used in classification tasks. One mode of ensemble learning is to generate the prediction function in parallel. Abdar, et al. [28] proposed a two-branches ensemble learning model. The first branch has five convolutional blocks, and the second branch is a transfer learning network based on VGG16. A fusion layer then concatenates the third, fourth, and fifth convolutional layers’ output with the VGG16’s output. Lu, et al. [29] employed ResNet-18 and ResNet-50 as backbone networks and fused the output by discriminant correlation analysis to obtain the refined features. Three randomized neural networks were trained using these refined features, and the predictions were ensembled. Khan, et al. [30] extracted features from AlexNet and VGG16 models and implemented an entropy-controlled Firefly optimization algorithm for the robust feature selection. Akram, et al. [31] first applied discrete wavelet transform and extended segmentation-based fractal texture analysis methods for feature extraction. Then an entropy controlled genetic algorithm was applied for feature selection. Another mode of ensembling is to generate each prediction function in order, as the latter model needs the weight of the previous model. Taherkhani, et al. [32] utilized the Adaboosting to construct a strong classifier by combining a group of weak classifiers. The final prediction result was obtained by multiplying the predicted value and weight of the weak classifiers. In all, ensemble learning-based models usually outperform single models when trained on a single dataset and are not easy to overfit. But they have high computational costs and redundant features may exist after feature selection and feature fusion.

C. image preprocessing models: Many studies incorporate the image preprocessing session in their classification system to improve the follow-up feature learning and extraction process.

Khan, et al. [30] proposed a new hybrid contrast enhancement approach by sequentially employing linear filters to improve the visual quality of images. The framework in [33] generates lung masks using maximum entropy segmentation to let the subsequent training of the model focus more on the lung regions.

Through summarizing the previous studies, we conclude an overview: (i) The ensemble learning models usually consume high computational costs. (ii) Most of these previous works are sensitive to noise and may have a performance drop in the absence of high-quality and abundant datasets. To overcome these limitations, (i) we adopted the strategy of training using the multi-learning-mode base estimators, which can reduce the number of iterations as much as possible while maintaining high performance. (ii) F-U2MNet-C preprocessing model was proposed to enhance the image's visual quality and eliminate the interference factors. In the experimental section, the related works [34] and the state-of-the-art methods in the field of medical image multi-classification [35-37] will be further explored and compared with our framework.

III. METHODOLOGY

Section A, B, C give the basics of the novel F-U2MNet-C. Section D, E, F give the basics of the novel AdaD-FNN. Section G shows the implementation and measure indicators.

A. Preprocessing using proposed F-U2MNet-C

Preprocessing has already shown its success in many COVID-19 applications. Assume the raw dataset T_R contained n 2D slice images be set as: $T_R = \{t_R(1), \dots, t_R(i), \dots, t_R(n)\}$. The size of each raw image was $size[t_R(i)] = height \times width \times channel$. Fig 1 displays the schematic graph of F-U2MNet-C.

First, the three-channel color CCT images were converted into grayscale images by retaining the luminance channel. The newly created grayscale dataset is: $T_G = F_{Grayscale}(T_R) = \{t_G(1), \dots, t_G(i), \dots, t_G(n)\}$.

Fuzzy enhancement technique was then employed on $t_G(i)$ to enhance the features in the grayscale image, which is discussed in Section B. The newly reconstructed fuzzy enhanced dataset: $T_F = F_{Fuzzy}(T_G) = \{t_F(1), \dots, t_F(i), \dots, t_F(n)\}$, where F_{Fuzzy} represents the fuzzy enhancement process.

Third, to minimize the interference in $t_F(i)$, we stacked the raw image $t_R(i)$ on $t_F(i)$ in terms of the enhanced correlation coefficient (ECC) criterion [38]. The fuzzy stacked dataset T_{ST} is set as: $T_{ST} = F_{Stack}(T_F) = \{t_{ST}(1), \dots, t_{ST}(i), \dots, t_{ST}(n)\}$, where F_{Stack} represents the image stacking process.

Fourth, to suit the input dimension of deep neural networks, each image in T_{ST} was resized to a smaller size of $[a, b]$, obtaining a downsampled dataset T_D as: $T_D = F_{DS}(T_{ST}) = \{t_D(1), t_D(2), t_D(3) \dots, t_D(i), \dots, t_D(n)\}$, where F_{DS} represents the downsampling (DS) process. $a=b=224$ for AdaD-FNN.

After the above preprocessing procedures, assume the input image size as $1024 \times 1024 \times 3$, each image will only cost about 1.60% of its original storage according to the byte compression ratio calculation: $\frac{224 \times 224 \times 3}{1024 \times 1024 \times 3} = 0.01595$.

Fifth, to remove the background interference factors such as the regions of the heart, ribs, and thoracic vertebrae, we implemented U2MNet segmentation on T_D . An element-wise

multiplication was conducted between the obtained finishing segmentation mask set M_{ED} and T_D to output T_{SE} : $T_{SE} = T_D \odot M_{ED} = \{t_{SE}(1), \dots, t_{SE}(i), \dots, t_{SE}(n)\}$, where \odot represents element-wise multiplication. Details are discussed in Section C.

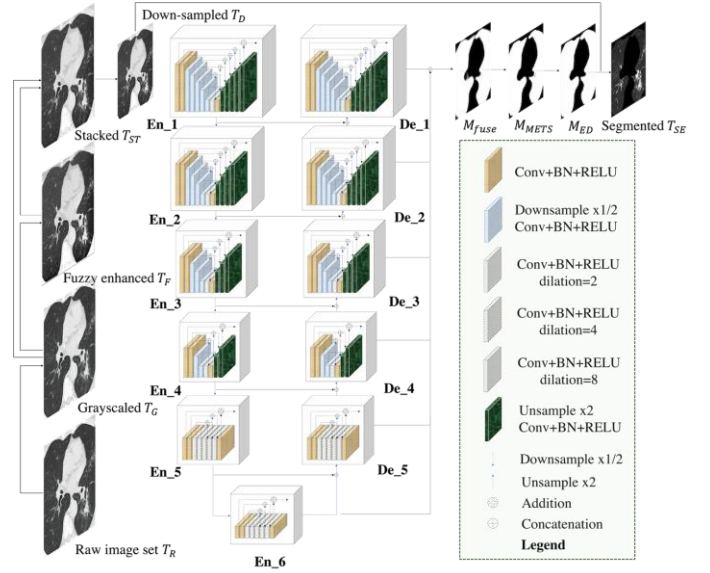


Fig 1. The schematic graph of F-U2MNet-C, which enhances the image features using fuzzy stacking and eliminates the interference factors using the U2MNet segmentation.

B. Improvement I: Using the fuzzy stacking enhancement

A partition is often built to divide the image into statistically uniform sub-images for image enhancement. However, classic partition often meets a block effect at the border of sub-images. To avoid this, we employed fuzzy partition in the enhancement stage of F-U2MNet-C. The logic behind the fuzzy partition is 1) Separate the input image into fuzzy windows. 2) Calculate the membership degree of each pixel according to the distance between the window and the pixel. 3) Obtain the fuzzy mean and the fuzzy variance within the window. 4) Summarize the weights of the images of each fuzzy window in a weight way to generate the output image [39, 40]. Here, the weight values used are membership degrees, which define the fuzzy partition.

Without loss of generality, the rectangle $\mathcal{S} = [x_0, x_1] \times [y_0, y_1]$ can be considered as image support. Let a fuzzy partition P of the image support \mathcal{S} be: $P = \{\mathbb{W}_{kl} | (k, l) \in [0, m] \times [0, n]\}$, where \mathbb{W}_{kl} represents the fuzzy window.

The space of gray levels is set as $E = (-1, 1)$. For each $t_F(i)$ heaving the support \mathcal{S} , the membership degrees ω_{kl} of a point $(x, y) \in \mathcal{S}$ to the fuzzy window \mathbb{W}_{kl} is defined according to:

$$\begin{cases} p_{kl}(x, y) = \frac{m!}{k!(m-k)!} \left(\frac{(x-x_0)^k (x_1-x)^{m-k}}{(x_1-x_0)^m} \right) \\ \cdot \frac{n!}{l!(n-l)!} \left(\frac{(y-y_0)^l (y_1-y)^{n-l}}{(y_1-y_0)^n} \right) \\ \omega_{kl}(x, y) = \frac{(p_{kl}(x, y))^\delta}{\sum_{l=0}^n \sum_{k=0}^m (p_{kl}(x, y))^\delta} \\ \text{s.t. } (k, l) \in [0, m] \times [0, n]. \delta \in (0, \infty), \mathcal{S} \rightarrow [0, 1] \end{cases}, \quad (1)$$

where (x, y) refers to the coordinates of a pixel within the support \mathcal{S} ; δ works as a tuning parameter that controls the fuzzification-defuzzification degree of the partition P .

For each window defined by Eq. (1) the fuzzy cardinality can be computed according to: $f_{card}(\mathbb{W}_{kl}) = \sum_{(x,y) \in \mathcal{S}} \omega_{kl}(x, y)$.

Further on, fuzzy statistics of a gray level image $t_G(i)$ are employed in relation to the fuzzy window \mathbb{W}_{kl} . The fuzzy mean $\mu(t_G(i), \mathbb{W}_{kl})$ and the fuzzy variance $\sigma^2(t_G(i), \mathbb{W}_{kl})$ within the window \mathbb{W}_{kl} are thus defined as:

$$\begin{cases} \mu(t_G(i), \mathbb{W}_{kl}) = \langle + \rangle_{(x,y) \in \mathcal{S}} \left(\frac{\omega_{kl}(x,y)}{f_{card}(\mathbb{W}_{kl})} \langle \times \rangle t_G(i)(x, y) \right) \\ \sigma^2(t_G(i), \mathbb{W}_{kl}) = \sum_{(x,y) \in \mathcal{S}} \frac{\omega_{kl}(x,y) \| t_G(i)(x,y) - \mu(t_G(i), \mathbb{W}_{kl}) \|_E^2}{f_{card}(\mathbb{W}_{kl})} \end{cases}, \quad (2)$$

where $\langle + \rangle$ is the addition; $\langle \times \rangle$ is the scalar multiplication; $\langle - \rangle$ is the subtraction; $\| \cdot \|_E$ is the norm; $f_{card}(\mathbb{W}_{kl})$ is the fuzzy cardinality for each window \mathbb{W}_{kl} .

On support fuzzification, the function that transforms the pixels belonging to the fuzzy window \mathbb{W}_{kl} is defined as:

$$\mathcal{T}_{kl}(t_G(i)) = \frac{\sigma_u}{\sigma(t_G(i), \mathbb{W}_{kl})} \langle \times \rangle (t_G(i) \langle - \rangle \mu(t_G(i), \mathbb{W}_{kl})), \quad (3)$$

where \mathcal{T}_{kl} represents the affine transform; σ_u^2 is set to $\frac{1}{3}$ in the experiment design.

Finally, the transform \mathcal{T}_{enh} is built as a sum of the affine transforms \mathcal{T}_{kl} , weighted according to the membership degree ω_{kl} , to achieve the enhanced image $t_F(i)$:

$$t_F(i) = \mathcal{T}_{enh}(t_G(i)) = \sum_{l=0}^n \sum_{k=0}^m \omega_{kl} \langle \times \rangle \mathcal{T}_{kl}(t_G(i)). \quad (4)$$

Significantly, as the transform is different depending on the brightness and contrast of each fuzzy window, the enhancement level shown in adjacent areas might be slightly different. To further improve the fuzzy enhancement, inspired by the work of [41], we stacked the raw image $t_R(i)$ on $t_F(i)$ in terms of the ECC criterion, which is a similarity measure for estimating the parameters of motion, and successfully solved this problem. Reasons for using ECC staking are: (i) Unlike the traditional similarity measure, ECC is variant to photometric distortions in brightness and contrast. (ii) ECC solves the optimization problem in a simple linear iterative strategy, which has a light computational cost. In all, compared with other enhancement methods, fuzzy stacking efficiently avoids the block effect and the adjacent area difference problem by supplying fuzzy partitions and enhanced correlation coefficient stacking.

C. Improvement II: A novel U2MNet

Due to the limited size of the dataset, the preliminary masks obtained from traditional U2Net will possibly encounter a boundary ambiguity problem as revealed in Fig 1, which hinders the ideal segmentation effect of completely removing the interference factors. Inspired by the work of U2Net [42], we presented an advanced version named U2MNet that can deal with relatively small datasets through applying a Maximum entropy threshold segmentation-based [43, 44] approach to the preliminary masks M_{fuse} .

Given an estimated probability density function $\rho(g)$ in the digital image, the entropy in the downsampled image $t_D(i)$ can be defined as:

$$\begin{aligned} H(t_D(i)) &= \sum_{u,v} \rho(t_D(i)(u, v)) \cdot \log \left\{ \frac{1}{\rho(t_D(i)(u,v))} \right\} \\ &= - \sum_{u,v} \rho(g) \cdot \log(\rho(g)), \end{aligned} \quad (5)$$

where H represents the cumulative histogram corresponding to the cumulative probability. g is the grayscale level (with the abscissa and ordinate value of $t_D(i)$ as u and v).

Given a specific threshold $\theta \in [0, C - 1]$, C stands for the limited range of pixel value (e.g., 256). For the two image regions Y_0 and Y_1 segmented by this threshold, the estimated probability density function can be expressed as:

$$\begin{cases} Y_0: \left(\frac{\rho(0)}{P_0(\theta)} \frac{\rho(1)}{P_0(\theta)} \dots \frac{\rho(\theta)}{P_0(\theta)} \ 0 \ 0 \dots 0 \right) \\ Y_1: \left(0 \ 0 \dots 0 \ \frac{\rho(\theta+1)}{P_1(\theta)} \ \frac{\rho(\theta+2)}{P_1(\theta)} \dots \frac{\rho(C-1)}{P_1(\theta)} \right), \\ \text{s.t.} \begin{cases} P_0(\theta) = \sum_{r=0}^{\theta} \rho(r) = P(\theta) \\ P_1(\theta) = \sum_{r=\theta+1}^{C-1} \rho(r) \end{cases} \end{cases}, \quad (6)$$

where $P_0(\theta)$ and $P_1(\theta)$ represent the cumulative probability of background and foreground pixels segmented by the threshold θ respectively, and the sum of them is 1.

The corresponding entropy of background $H_0(\theta)$, foreground $H_1(\theta)$ and the total entropy of the image $t_D(i)$ are defined as:

$$\begin{cases} H_0(t_D(i)|\theta) = - \sum_{r=0}^{\theta} \frac{\rho(r)}{P_0(\theta)} \log \left(\frac{\rho(r)}{P_0(\theta)} \right) \\ H_1(t_D(i)|\theta) = - \sum_{r=\theta+1}^{C-1} \frac{\rho(r)}{P_1(\theta)} \log \left(\frac{\rho(r)}{P_1(\theta)} \right) \\ H_{01}(t_D(i)|\theta) = H_0(\theta) + H_1(\theta) \end{cases}. \quad (7)$$

Through calculating the total entropy of the image under all the segmentation thresholds, we determined the final threshold by the segmentation threshold corresponding to the maximum entropy. The segmentation mask set M_{METS} was obtained by considering the pixel in the image whose gray value is larger than this threshold as the foreground, and the pixel whose gray value is smaller than the threshold as the background.

To avoid a circumstance that some lesion regions be eliminated together with the interference factors, a refining module ED was added into U2Mnet to create a finishing mask M_{ED} by utilizing the erosion&dilation processing approach [45-47]. This approach depends solely on relative ordering, rather than the numerical values of the pixel values, thus is especially suited to the processing of the binary image.

D. Classification using proposed AdaD-FNN

An overview of our novel high-sensitivity deep learning framework AdaD-FNN for COVID 19 diagnosis is illustrated in part A of Fig 2. AdaD is an ensembled algorithm where a sequence of base estimators E is trained by a data weight vector $W = \{w_i\}$, $i = 1, 2, \dots, n$ to construct a strong classifier with higher classification ability. In our study, each estimator is set as a deep fractional max-pooling neural network (FNN). Suppose the training dataset is set as: $\{(\kappa_1, e_1), \dots, (\kappa_n, e_n)\}$, s.t. $e_i \in (1, 2, \dots, L)$, where κ_i is an input vector, e_i is the

corresponding output of κ_i , n is the number of training samples, L is the total number of classes.

The first estimator $E^{k=1}(\kappa)$, where k is the index of estimators, is trained on all the training samples with the same weight of an initialized $W_1 = \frac{1}{n}$ and the learning rate η_1 .

The output vector for an input sample κ_i is defined as: $OP(\kappa_i) = [o_l(\kappa_i)]$, s. t. $l \in (1, 2, \dots, L)$, where OP stands for the probabilities that the applied input belongs to the L classes.

$OP^k(\kappa_i)$ is then used to update the data weights $W = \{w_i\}$ by:

$$w_i^{k+1} = w_i^k \exp\left(-\eta_k \frac{L-1}{L} Z_i^T \log(OP^k(\kappa_i))\right), i = 1, \dots, n \quad (8)$$

where w_i^{k+1} is the weight of the i^{th} training sample utilized by the k^{th} estimator, $OP^k(\kappa_i)$ refers to the output vector of the k^{th}

estimator in response to the i^{th} training sample, η_k stands for the learning rate of the k^{th} estimator, Z_i refers to the label vector corresponding to the i^{th} training sample.

This weight updating concept helps the training sample focus more on the hard-to-identify samples by increasing the weights for the misclassified samples in further learning. The construction of a strong classifier is completed when the expected error rate is reached. After the training of K base estimators, the output class $E(\kappa)$ is predicted through:

$$\begin{cases} E(\kappa) = \operatorname{argmax}_l \sum_{k=1}^K q_l^k(\kappa) \\ \text{s.t. } q_l^k(\kappa) = (L-1) \left(\log(o_l^k(\kappa)) - \frac{1}{L} \sum_{l=1}^L \log(o_l^k(\kappa)) \right) \end{cases} \quad (9)$$

where $o_l^k(\kappa)$ stands for the l^{th} element of the output vector of the k^{th} estimator when κ is applied as its input.

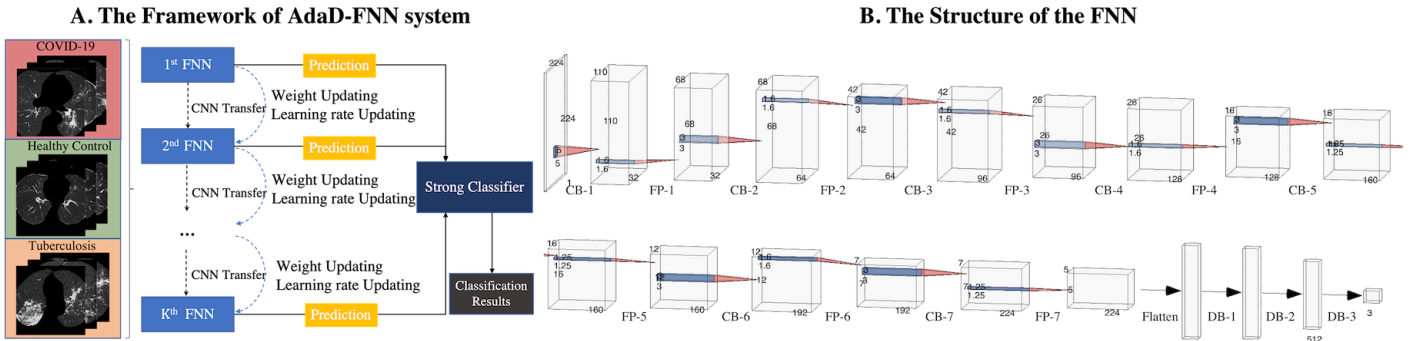


Fig 2. The overview of proposed AdaD-FNN framework, which sequentially transfers the trained knowledge of an FNN estimator to the next one while updating the weights of the samples in the training set with a decaying learning rate. This framework inhibits the network from remembering the noisy information and improves the learning of complex patterns in the hard-to-identify samples.

The structure of the proposed 10-layer FNN is demonstrated in part B of Fig 2, within which 2D Conv Blocks (CBs), Fractional max-pooling Layers (FPs), and Dense Blocks (DBs) are utilized as the main building elements. In each CB block, the convolutional layer (CL) outputs are fed into a batch normalization layer to avoid the covariate shift, which will probably cause gradient divergency during backpropagation.

E. Improvement III: Using FP to replace MP and AP

To avoid the rapid loss of surrounding information, Graham [48] formulated a fractional version of max-pooling (MP) named fractional max-pooling (FP) by giving allowance for the multiplicative factor α to be a non-integer value. For instance, if $N_{in}/N_{out} \approx \sqrt[n]{2}$, then the reduction rate of feature information will be n times slower. From another perspective, a non-integer α gives chances for more pooling layers to be used in the backbone of the neural network. For instance, if pooling in the neural network structure is to reduce the size of FM by a factor of $\sqrt{2}$, then twice as many layers of pooling could be used when compared with a factor of 2. Every time of pooling can be considered as an opportunity to view the input image at a different scale, and viewing the image at the ‘correct’ scale would help recognise the ‘distinct’ features that could identify subjects belonging to a particular class, thus we

replaced MP and average pooling (AP) with FP to improve the recognition capability of our proposed network [48, 49].

In this study, we conducted comparative experiments on the disjointed FP and the overlapped FP, which will be discussed in Section IV. For each FP operation {FP-1, FP-2, FP-3, FP-4, FP-5, FP-6, FP-7}, α is set as {1.6, 1.6, 1.6, 1.6, 1.25, 1.6, 1.25} based on [48] and the TensorFlow documentation. An illustration graph for the FP results with $\alpha = 1.25, 1.6$ and the traditional MP results with $\alpha = 2$ is shown in Fig 3.

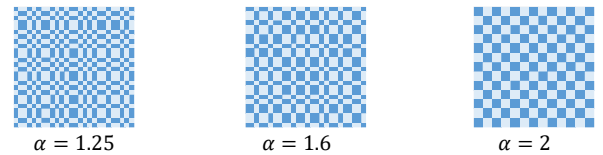


Fig 3. An illustration of the FP and MP results.

F. Improvement IV: Incorporate a IrDecay inside the AdaD

In the traditional AdaBoosting (Ada) structure, when the prescribed number of iterations is reached, the training of samples will stop without finding the expected error rate. However, simply setting a large iteration number for searching the expected error rate would bring huge computation costs and make the whole system long time lasting. Therefore, we

adopted the idea of incorporating a learning rate decay (*lrDecay*) module into the structure to help every classifier find the global minimum of error function of neural networks and achieve better performance. This new version of Ada structure substitutes the traditional format of base estimator with multi-learning-mode base estimator and is named as AdaD structure. A toy illustration is shown in Fig 4, where $E^k(\kappa)$ represents the K^{th} estimator when κ is applied as its input; $W = \{w_i\}$ represents the data weights.

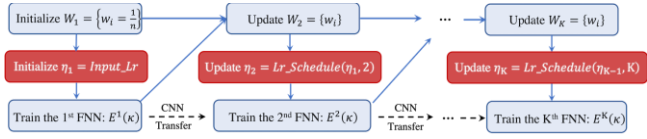


Fig 4. A toy illustration for AdaD structure.

In the *lrDecay* module, the learning rate η is scheduled to be reduced after the $\mathcal{D}_1^{\text{th}}$, $\mathcal{D}_2^{\text{th}}$, ..., $\mathcal{D}_h^{\text{th}}$ estimator according to

$$\eta_k = \begin{cases} \eta \cdot \varrho^0, & k < \mathcal{D}_1 \\ \eta \cdot \varrho^1, & \mathcal{D}_1 < k < \mathcal{D}_2 \\ \dots & \dots \\ \eta \cdot \varrho^{h-1}, & \mathcal{D}_{h-1} < k < \mathcal{D}_h \end{cases}, \quad (10)$$

where k represents the index of estimator, η_k represents the learning rate of the k^{th} estimator, \cdot stands for a multiplication process, ϱ is a drop factor belonging to $(0, 1)$.

Through introducing this module, the initially large η will inhibit the network from remembering the noisy information of the dataset, while the subsequent decaying η will gradually improve the learning of complex patterns in the image[50]. The final determination of parameter setting in our study is $\eta_1=0.05$, $\varrho=0.6$, $\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5, \mathcal{D}_6\} = \{2, 3, 4, 5, 6, 7\}$, $K=10$.

G. Implementation and measures

The pseudocode for the implementation of our proposed F-U2MNet-C and AdaD-FNN is listed in Algorithm 1. The learning rate was initialized with 0.05 and the drop factor was set as 0.6. For our proposed method and all the comparison methods, we totally trained 80 epochs with a batch size as 10. To make complete use of the dataset information, a five-fold cross-validation was implemented, and we would have five times of running. In each running, images were split into 80% for training and 20% for testing. The F-U2MNet-C was used on the training set of five-fold cross-validation. Four performance metrics (accuracy, precision, sensitivity and F1 score) are introduced to comprehensively evaluate our framework.

Algorithm 1. Pseudocode of our proposed framework

Input: Original Image set T_R and its ground truth label G_t

Phase I: Preprocessing

- Separate the input image into fuzzy windows.
 - Calculate the membership degree of each pixel, see Eq. (1).
 - Obtain the image mean and variance, see Eq. (2)
 - Generate T_F by summing up the images of every fuzzy window in a weight way, see Eqs. (3)(4).
 - Generate T_{ST} through stacking fuzzy enhanced dataset and raw dataset.
 - Segment the images utilizing U2MNet and get T_{SE} , see Eqs. (5)(6)(7).
-

Phase II: Five-fold cross-validation on the Training set

```

Split  $T_{SE}$  into training set and testing set:  $T_{SE} \rightarrow \{T_{SE}^{train}, T_{SE}^{test}\}$ 
for  $i = 1:5$  %  $T_{SE}^{train,i}$  is the training set,  $T_{SE}^{test,i}$  is the testing set.
  Initialize the data sample weight  $W_1 = 1/n$ ,  $n$  is the total number
  of training samples.
  for  $k=1$  to  $K$ :
    if  $k == 1$ :
      Train the first FNN using the initial sample weights.
    else:
      Transfer the learning parameters of the previous FNN
      to the  $k^{\text{th}}$  FNN.
      Train the  $k^{\text{th}}$  FNN using the sample weight vector  $W_k$ 
      and the learning rate  $\eta_k$ .
    end
    Obtain the output of the  $k^{\text{th}}$  FNN.
    Update the data sample weight  $W_k$ , see Eq. (8).
    Re-normalize the updated data sample weight  $W_k$ .
    Update the learning rate  $\eta_k$ , see Eq. (10).
    Save the  $k^{\text{th}}$  FNN.
  end
end

```

Phase III: Report the test performance of our proposed framework

```

The Training set is  $T_{SE}^{train}$  and its labels  $G_t(T_{SE}^{train})$ 
The Testing set is  $T_{SE}^{test}$  and its labels  $G_t(T_{SE}^{test})$ 
for  $i = 1:5$ 
  Prediction:  $Pred(i) = \text{predict}(\text{AdaDFNN}^i, T_{SE}^{test,i})$ , see Eq. (9).
  Calculate Indicators: accuracy, precision, sensitivity, F1 score.
end

```

Output: The test performances.

IV. EXPERIMENTS, RESULTS, AND DISCUSSIONS

The experiments were written using the language of Python 3.7.10 and the programming platform of Matlab R2021b. The programs ran with NVIDIA TESLA P100 GPUs. The performances are reported over the test sets with five runs.

A. Datasets and Statistical results

The proposed framework is evaluated on three public datasets with 2D slices of CT volumes *TLDCA* [29], *UCSD-A14H* [16], *SARS-CoV-2* [51], and one with entire CT volumes *TCIA* [52]. Among these datasets, *TLDCA* (denoted as Site A) consists of 1260 images, in which 420 are positive in COVID-19, 420 are healthy control (HC) and 420 are tuberculosis (TB). The resolution of each image is 1024×1024 . *UCSD-A14H* (denoted as Site B) includes 349 COVID-19 images and 190 HCs. The spatial sizes of images range from 102×137 to 1853×1485 . *SARS-CoV-2* (denoted as Site C) includes 2482 images from 120 patients, in which 1252 are positive in COVID-19 and 1230 are HCs. The spatial sizes of these images range from 119×104 to 416×512 . The NiftI dataset *TCIA* (denoted as Site D) consists of unenhanced chest CTs from 661 patients. We used 974 images, including 473 with COVID-19 infections and 501 HCs. The resolution of these slices is 512×512 .

Our experiment conducted five-fold cross-validation on each dataset. We report the results in form of average and standard deviation in TABLE I, and the results of ROC curves over four data sites in Fig 5. Through analyzing the misclassification cases in 10 runs, we made two-fold conclusions: (i) Many of these misclassifications are abnormal samples. The patterns of these samples are different from the same category samples input in the training set. They may obtain high weight in AdaD iteration and affect the prediction accuracy of the final strong classifier. (ii) There exists inconsistency in data. In other words,

images collected from different machines in hospitals may not have the exact same resolution. Besides, in a few cases, there exist foreign objects or texts in the images. As a consequence, these issues impacted the model and led to misclassifications.

TABLE I
STATISTICAL RESULTS ON PUBLIC DATASETS

Dataset	Acc (%)	Prc (%)	Sen (%)	F1 (%)
Site A	99.52±0.52	99.54±0.50	99.52±0.52	99.52±0.52
Site B	92.96±2.49	93.00±2.49	92.96±2.49	92.91±2.53
Site C	97.86±1.16	97.88±1.16	97.86±1.16	97.86±1.16
Site D	91.97±1.52	92.23±1.48	91.97±1.52	91.97±1.52

(Acc=Accuracy. Prc=Precision. Sen=Sensitivity. F1=F1 score.)

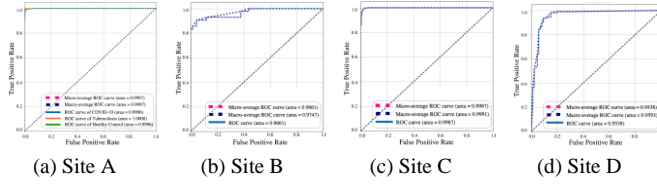


Fig 5. ROC curves over Site A, B, C, D.

To further explore the robustness of our framework, we artificially lower the image quality by separately introducing ‘Gaussian-distributed additive noise’ (var=0.01), ‘Poisson-distributed noise’ and ‘salt&pepper noise’ (amount=0.05) into four datasets. Through analyzing the accuracy drop in TABLE II, we concluded that all the drops are within 5%, thus our framework is robust to different categories of noise.

TABLE II
ACCURACY DROP (%) UNDER DIFFERENT NOISE SCENARIOS

Noise	Site A	Site B	Site C	Site D
gaussian	-0.23	-2.34	-2.25	-2.68
poisson	-0.15	-2.22	-2.09	-1.44
s&p	-1.27	-3.49	-2.98	-3.16

B. Ablation study

The experiments were conducted on site A to further explore the effectiveness of each component in the final results.

(1) *Effectiveness of F-U2MNet-C*: Assume the datasets be preprocessed at four stages, including ‘Stage I: raw dataset’, ‘Stage II: Stage I + fuzzy enhanced’, ‘Stage III: Stage II + ECC stacking’, ‘Stage IV: Stage III + U2MNet segmentation’. After being trained using AdaD-FNN, the results in TABLE III demonstrate that each stage outperforms its previous stage, which represents the ‘enhanced’, ‘stacking’ and ‘segmentation’ components incorporated in F-U2MNet-C are all effective. The whole preprocessing model not only helps improve all the metrics with a nearly 2% increase but also enhances the stability of the system. These improvements are consistent with our expectation that the F-U2MNet-C can optimize the training process because it enhances the features and removes the interference factors in the grayscale image.

TABLE III
COMPARISON AGAINST DIFFERENT STAGES IN F-U2MNET-C

Stage	Acc (%)	Prc (%)	Sen (%)	F1 (%)
I	97.62±2.79	97.91±2.33	97.62±2.79	97.63±2.78
II	98.73±1.39	98.77±1.33	98.73±1.39	98.74±1.38
III	98.81±0.97	98.87±0.75	98.81±0.97	98.81±0.97
IV	99.52±0.52	99.54±0.50	99.52±0.52	99.52±0.52

(In TABLE III-VII, the best results are shown in bold.)

(2) *Effectiveness of AdaD*: Assume ‘FNN’ as ‘single FNN without prediction ensemble technique’, ‘Ada-FNN’ as ‘FNN trained using traditional AdaBoosting ensemble technique’, and ‘AdaD-FNN’ as ‘FNN trained using proposed AdaD ensemble technique’. As can be observed in TABLE IV, AdaD-FNN outperforms FNN on all the evaluated metrics with 3.96% in accuracy, 3.67% in precision, 3.96% in sensitivity, 4.09% in F1 score and around 3% in stability. When compared with the Ada-FNN, AdaD-FNN also has nearly 2% improvements on all metrics. There are two reasons for this improvement. First, the weight updating concept in the AdaD structure could help the training sample focus more on the hard-to-identify samples by increasing the weights for the misclassified samples in further learning. Second, the *lrDecay* module inside the AdaD structure can offer help in inhibiting the network from remembering the noisy information in the dataset and improving the learning of complex patterns in the hard-to-identify samples.

TABLE IV
COMPARISON AGAINST DIFFERENT STRUCTURES

Model	Acc (%)	Prc (%)	Sen (%)	F1 (%)
FNN	95.56±3.90	95.86±3.48	95.56±3.90	95.43±4.12
Ada-FNN	97.62±1.40	97.72±1.25	97.62±1.40	97.61±1.43
AdaD-FNN	99.52±0.65	99.53±0.64	99.52±0.65	99.52±0.66

(3) *Drop factor comparison*: We then tried different settings of drop factor ρ in our *lrDecay* module inside the AdaD structure to determine the optimal value of ρ . As displayed in TABLE V, our framework gives the best classification results with $\rho = 0.6$. Training with $\rho = 1$ gives the worst performance, further proving that incorporating the *lrDecay* module into AdaD structure could bring improvement in performance. In all, the proposed *lrDecay* in AdaD structure is effective. The optimal value for ρ can be varied for another dataset.

TABLE V
COMPARISON AGAINST DIFFERENT DROP FACTORS ρ

ρ	Acc (%)	Prc (%)	Sen (%)	F1 (%)
1.0	97.62±1.40	97.72±1.25	97.62±1.40	97.61±1.43
0.8	98.65±0.82	98.67±0.81	98.65±0.82	98.65±0.83
0.6	99.52±0.52	99.54±0.50	99.52±0.52	99.52±0.52
0.4	99.13±0.71	99.14±0.70	99.13±0.71	99.13±0.71
0.2	98.89±0.81	98.91±0.79	98.89±0.81	98.89±0.81

(4) *Effectiveness of FNN*: We compared the results between the disjointed FNN, overlapped FNN, standard CNN using MP and standard CNN using AP under the training of AdaD structure. The results in TABLE VI clearly reveal that the disjointed type of AdaD-FNN outperforms other pooling methods, which validates the effectiveness of our proposed FNN. This is because the utilization of FP could avoid the rapid loss of surrounding information and meanwhile allow more

times of viewing the input image at a different scale when compared with traditional MP and AP.

TABLE VI
COMPARISON AGAINST DIFFERENT POOLING METHODS

AdaD-	Acc (%)	Prc (%)	Sen (%)	F1 (%)
FNN-disjoint	99.52±0.52	99.54±0.50	99.52±0.52	99.52±0.52
FNN-overlap	98.89±0.81	98.91±0.79	98.89±0.81	98.89±0.81
CNN (MP)	98.97±1.47	99.00±1.41	98.97±1.47	98.97±1.47
CNN (AP)	98.33±1.30	98.39±1.28	98.33±1.30	98.33±1.30

C. Comparison to the state-of-the-art approaches

We compared our approach with state-of-the-art COVID-19 classification methods, as shown in TABLE VII. To conclude, our framework outperforms the most of methods among four public datasets, indicating the superiority of our framework to exploit more robust features from heterogeneous datasets. On Site C, our framework shows a lower precision when compared with xDNN [34], but the sensitivity of ours is 2.33% higher than theirs. In real life, a high-sensitivity diagnosis system would offer significant help for the management of COVID-19 due to the false-negative problem of RT-PCR.

TABLE VII
COMPARISON AGAINST STATE-OF-THE-ART APPROACHES

Site	Approach	Acc (%)	Prc (%)	Sen (%)	F1 (%)	
A	DNS [17]	89.21±11.65	91.71±8.67	89.21±11.65	88.43±12.59	
	RDE [29]	96.35±1.97	97.38±3.98	97.86±1.50	97.62±2.18	
	6L-CNN [35]	93.97±1.74	94.33±1.49	93.97±1.74	93.99±1.66	
	COVNet [18]	97.94±2.42	98.07±2.18	97.94±2.42	97.90±2.51	
	CovidAID [15]	96.83±2.00	97.04±1.84	96.83±2.00	96.80±2.03	
	CLCnet [36]	96.11±1.72	96.44±1.45	96.11±1.72	96.13±1.70	
	BSN [19]	97.22±1.51	97.27±1.51	97.22±1.51	97.22±1.52	
	CSGBBNet [33]	98.01±1.12	98.02±1.13	98.01±1.12	98.01±1.13	
	AdaBoost-CNN [32]	98.97±1.47	99.00±1.41	98.97±1.47	98.97±1.47	
	Ensemble DCNN [27]	97.14±2.05	97.32±1.78	97.14±2.05	97.14±2.07	
	UFN [28]	96.59±0.72	96.67±0.63	96.59±0.72	96.57±0.73	
	CUIEM-Covid [26]	98.49±0.76	98.54±0.74	98.49±0.76	98.49±0.76	
	Ours	99.52±0.52	99.54±0.50	99.52±0.52	99.52±0.52	
	COVID-Net [22]	63.12±2.09	64.03±3.91	57.73±2.94	61.09±1.28	
	COVID-Net-R [23]	77.07±1.92	79.48±0.96	74.69±3.91	77.04±2.17	
Series Adapter [24]	70.01±3.82	63.04±4.87	74.91±1.89	67.08±3.09		
Parallel Adapter [25]	74.93±1.83	79.84±1.75	71.81±2.47	73.46±1.68		
MS-Net [37]	76.23±1.81	79.29±1.48	74.07±1.29	76.54±1.73		
SNC [23]	78.69±1.54	78.02±1.34	79.71±1.42	78.83±1.43		
B	ResNet-50 [21]	82.57±5.81	82.79±5.77	82.57±5.81	82.62±5.79	
	DNC [16]	89.1	-	-	89.6	
	AdaBoost-CNN [32]	92.03±1.53	92.37±1.52	92.03±1.53	91.95±1.48	
	CUIEM-Covid [26]	88.86±4.47	89.22±4.55	88.86±4.47	88.90±4.45	
	UFN [28]	89.60±2.86	89.69±2.81	89.60±2.86	89.58±2.80	
	Ensemble DCNN [27]	84.97±2.30	85.54±2.52	84.97±2.30	85.03±2.35	
	Ours	92.96±2.49	93.00±2.49	92.96±2.49	92.91±2.53	
	COVID-Net [22]	77.12±0.98	80.04±2.87	70.97±2.37	76.03±1.13	
	COVID-Net-R [23]	89.09±1.08	94.58±2.07	83.78±0.62	88.97±0.91	
	Series Adapter [24]	85.73±0.71	90.08±0.79	81.91±2.61	86.19±1.65	
	Parallel Adapter [25]	82.13±1.91	83.51±1.87	80.02±2.47	82.39±1.78	
	MS-Net [37]	87.98±1.31	93.78±2.76	84.91±2.83	88.73±1.20	
	SNC [23]	90.83±0.93	95.75±0.43	85.89±1.05	90.87±1.29	
	C	xDNN [34]	97.38	99.16	95.53	97.31
		AdaBoost-CNN [32]	95.16	93.63	96.71	95.14
CUIEM-Covid [26]		93.59±1.19	93.62±1.16	93.59±1.19	93.59±1.18	
UFN [28]		95.81±0.22	95.85±0.23	95.81±0.22	95.81±0.22	
Ensemble DCNN [27]		95.57±0.51	95.59±0.52	95.57±0.51	95.57±0.51	
Ours		97.86±1.16	97.88±1.16	97.86±1.16	97.86±1.16	

3D AI-based [53]	90.8	79.4	84.0	81.6
AdaBoost-CNN [32]	89.32±2.51	90.22±1.41	89.32±2.51	89.25±2.59
CUIEM-Covid [26]	81.98±1.18	82.42±0.90	81.98±1.18	81.83±1.37
UFN [28]	85.78±2.61	85.97±2.52	85.78±2.61	85.78±2.60
Ensemble DCNN [27]	83.738±2.2383	84.8±2.2183	83.738±2.2383	84.698±2.21
Ours	91.97±1.52	92.23±1.48	91.97±1.52	91.97±1.52

(‘DNS’ refers to ‘DenseNet-201-SVM’; ‘RDE’ refers to ‘ResNets-DCA-EN’; ‘UFN’ refers to ‘UncertaintyFuseNet’; ‘BSN’ refers to ‘Bayes-SqueezeNet’; ‘COVID-Net-R’ refers to ‘COVID-Net (Redesign)’; ‘SNC’ refers to ‘SepNorm + Contrastive’; ‘DNC’ refers to ‘DenseNet-169-CSSL’.)

The reasons behind the good results are: First, preprocessing using F-U2MNet-C can help AdaD-FNN better capture the semantic representations and facilitate a smooth training process by eliminating the interference factors. Second, the weight updating concept helps the training sample focus more on the hard-to-identify samples. Thirdly, with the incorporation of *lrDecay* module, the proposed network was inhibited from remembering the noisy information, and its learning ability of complex patterns was improved. Fourth, FP layers provide more opportunities to view the input image at a different scale, which improves the recognition capability of the network.

D. Explainability

To understand the manner of the framework, gradient-CAM [54] was applied on some randomly chosen sample images. For each category, the generated attention heatmaps and the corresponding manual delineations are shown in Fig 6.

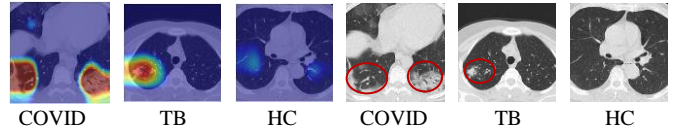


Fig 6. Heatmaps (left three) and delineations (right three) of COVID-19, TB and HC samples.

The results demonstrate that for COVID-19 and TB images, our framework focuses more on lesion areas (circled in red) and less on non-lesion areas. For HC images, the model's attention is not focused on any specific area, since there are no lesion areas in the HC group. In all, these heatmaps show how our framework predicts COVID-19, TB, and HC images in a clear and understandable manner. The concerns of our framework are highly consistent with the standard already approved in the medical community, which adds confidence that it can assist the diagnosis of radiologists in the real world.

V. CONCLUSION

In this study, we proposed a novel image preprocessing model F-U2MNet-C and a novel classification model AdaD-FNN, which entails five improvements: (i) proposed F-U2MNet-C with the usage of fuzzy partitions and ECC stacking, (ii) proposed U2MNet (iii) novel AdaD-FNN model (iv) usage of fractional max-pooling in proposed FNN, (v) novel *lrDecay* module. Through the help of these improvements, we created an explainable system for COVID-19 detection that shows advantages in diagnostic rate and stability not only on the public datasets but also when compared with other state-of-the-art deep learning approaches. It achieved an accuracy of 99.52% on TLDCA, 92.96% on UCSD-A14H, 97.86% on SARS-CoV-

2 and 91.97% on TCIA. Although the good performance is achieved, the limitation of our framework still exists. In our framework, the network is sensitive to abnormal samples, which may obtain high weight in iteration and affect the prediction accuracy of the final strong classifier. Future works will mainly focus on the incorporation of more advanced, rapid deep learning techniques into our system. Besides, we are interested in pretraining our framework on more large-scale datasets, which contain more classes of chest diseases, such as community-acquired pneumonia.

REFERENCES

- [1] Z. Ye, Y. Zhang, Y. Wang, Z. Huang, and B. Song, "Chest CT manifestations of new coronavirus disease 2019 (COVID-19): a pictorial review," *European radiology*, vol. 30, no. 8, pp. 4381-4389, 2020.
- [2] E. Ortiz-Prado *et al.*, "Clinical, molecular, and epidemiological characterization of the SARS-CoV-2 virus and the Coronavirus Disease 2019 (COVID-19), a comprehensive literature review," *Diagnostic microbiology and infectious disease*, vol. 98, no. 1, p. 115094, 2020.
- [3] L. M. Kucirka, S. A. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, "Variation in false-negative rate of reverse transcriptase polymerase chain reaction-based SARS-CoV-2 tests by time since exposure," *Annals of internal medicine*, vol. 173, no. 4, pp. 262-267, 2020.
- [4] B. Wang *et al.*, "AI-assisted CT imaging analysis for COVID-19 screening: Building and deploying a medical AI system," *Applied Soft Computing*, vol. 98, p. 106897, 2021.
- [5] S. Cao *et al.*, "Post-lockdown SARS-CoV-2 nucleic acid screening in nearly ten million residents of Wuhan, China," *Nature communications*, vol. 11, no. 1, pp. 1-7, 2020.
- [6] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, and J. Liu, "Chest CT for typical coronavirus disease 2019 (COVID-19) pneumonia: relationship to negative RT-PCR testing," *Radiology*, vol. 296, no. 2, pp. E41-E45, 2020.
- [7] Y. Fang *et al.*, "Sensitivity of chest CT for COVID-19: comparison to RT-PCR," *Radiology*, vol. 296, no. 2, pp. E115-E117, 2020.
- [8] L. Li *et al.*, "Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT," *Radiology*, vol. 296(2), pp. E65-E71, 2020.
- [9] S.-H. Wang, Y. Zhang, X. Cheng, X. Zhang, and Y.-D. Zhang, "PSSPNN: PatchShuffle Stochastic Pooling Neural Network for an explainable diagnosis of COVID-19 with multiple-way data augmentation," *Computational and Mathematical Methods in Medicine*, vol. 2021, 2021.
- [10] B. Li, J. Shen, L. Li, and C. Yu, "Radiographic and clinical features of children with coronavirus disease (COVID-19) pneumonia," *Indian pediatrics*, vol. 57, pp. 423-426, 2020.
- [11] R. Sánchez-Oro, J. T. Nuez, and G. Martínez-Sanz, "Radiological findings for diagnosis of SARS-CoV-2 pneumonia (COVID-19)," *Medicina clinica (English ed.)*, vol. 155, no. 1, p. 36, 2020.
- [12] D. Caruso *et al.*, "Chest CT features of COVID-19 in Rome, Italy," *Radiology*, vol. 296, no. 2, pp. E79-E85, 2020.
- [13] X. Yu, S. Lu, L. Guo, S.-H. Wang, and Y.-D. Zhang, "ResGNet-C: A graph convolutional neural network for detection of COVID-19," *Neurocomputing*, vol. 452, pp. 592-605, 2021.
- [14] M. Scarpiniti, S. S. Ahrabi, E. Baccarelli, L. Piazza, and A. Momenzadeh, "A novel unsupervised approach based on the hidden features of Deep Denoising Autoencoders for COVID-19 disease detection," *Expert Systems with Applications*, vol. 192, p. 116366, 2022.
- [15] A. Mangal *et al.*, "CovidAID: COVID-19 detection using chest X-ray," *arXiv preprint arXiv:2004.09803*, 2020.
- [16] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie, "COVID-CT-dataset: a CT scan dataset about COVID-19," *arXiv preprint arXiv:2003.13865*, 2020.
- [17] Z. Yu *et al.*, "Rapid identification of COVID-19 severity in CT scans through classification of deep features," *BioMedical Engineering OnLine*, vol. 19, no. 1, pp. 1-13, 2020.
- [18] L. Li *et al.*, "Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: evaluation of the diagnostic accuracy," *Radiology*, vol. 296, no. 2, pp. E65-E71, 2020.
- [19] F. Ucar and D. Korkmaz, "COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images," *Medical hypotheses*, vol. 140, p. 109761, 2020.
- [20] N.-u. Rehman *et al.*, "A self-activated cnn approach for multi-class chest-related COVID-19 detection," *Applied Sciences*, vol. 11, no. 19, p. 9023, 2021.
- [21] S. Walvekar and D. Shinde, "Detection of COVID-19 from CT images using resnet50," in *2nd International Conference on Communication & Information Processing (ICCIP)*, 2020.
- [22] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific Reports*, vol. 10, no. 1, pp. 1-12, 2020.
- [23] Z. Wang, Q. Liu, and Q. Dou, "Contrastive cross-site learning with redesigned net for covid-19 ct classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 2806-2813, 2020.
- [24] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," *Advances in neural information processing systems*, vol. 30, 2017.
- [25] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8119-8127.
- [26] A. K. Das, S. Ghosh, S. Thunder, R. Dutta, S. Agarwal, and A. Chakrabarti, "Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1111-1124, 2021.
- [27] S. D. Deb, R. K. Jha, K. Jha, and P. S. Tripathi, "A multi model ensemble based deep convolution neural network structure for detection of COVID19," *Biomedical Signal Processing and Control*, vol. 71, p. 103126, 2022.
- [28] M. Abdar *et al.*, "UncertaintyFuseNet: robust uncertainty-aware hierarchical feature fusion model with ensemble Monte Carlo dropout for COVID-19 detection," *Image Video Process*, 2022.
- [29] S. Lu, D. Wu, Z. Zhang, and S.-H. Wang, "An Explainable Framework for Diagnosis of COVID-19 Pneumonia via Transfer Learning and Discriminant Correlation Analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 3s, pp. 1-16, 2021.
- [30] M. A. Khan *et al.*, "COVID-19 case recognition from chest CT images by deep learning, entropy-controlled firefly optimization, and parallel feature fusion," *Sensors*, vol. 21, no. 21, p. 7286, 2021.
- [31] T. Akram *et al.*, "A novel framework for rapid diagnosis of COVID-19 on computed tomography scans," *Pattern analysis and applications*, vol. 24, no. 3, pp. 951-964, 2021.
- [32] A. Taherkhani, G. Cosma, and T. M. McGinnity, "AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning," *Neurocomputing*, vol. 404, pp. 351-366, 2020.
- [33] Z.-Q. Zhu, Y.-D. Zhang, X.-J. Yao, and S.-H. Wang, "CSGNet: An Explainable Deep Learning Framework for COVID-19 Detection," *Diagnostics*, vol. 11, no. 9, p. 1712, 2021.
- [34] E. Soares, P. Angelov, S. Biaso, M. Froes, and D. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. medRxiv 2020," *Google Scholar*.
- [35] X. Jiang and Y.-D. Zhang, "Chinese sign language fingerspelling via six-layer convolutional neural network with leaky rectified linear units for therapy and rehabilitation," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 9, pp. 2031-2090, 2019.
- [36] L. Tong *et al.*, "An Automated Cell Line Authentication Method for AstraZeneca Global Cell Bank using Deep Neural Networks on Brightfield Images," 2021.
- [37] Q. Liu, Q. Dou, L. Yu, and P. A. Heng, "MS-Net: multi-site network for improving prostate segmentation with heterogeneous MRI data," *IEEE transactions on medical imaging*, vol. 39, no. 9, pp. 2713-2724, 2020.
- [38] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858-1865, 2008.
- [39] V. Pătrașcu, "Color image enhancement using the support fuzzification," in *International Fuzzy Systems Association World Congress*, 2003: Springer, pp. 412-419.
- [40] L. Hildebrand and B. Reusch, "Fuzzy color processing," in *Fuzzy Techniques in Image Processing*: Springer, 2000, pp. 267-286.
- [41] M. Toğaçar, B. Ergen, and 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," *Computers in biology and medicine*, vol. 121, p. 103805, 2020.

[42] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-Net: Going deeper with nested U-structure for salient object detection," *Pattern Recognition*, vol. 106, p. 107404, 2020.

[43] A. K. Wong and P. K. Sahoo, "A gray-level threshold selection method based on maximum entropy principle," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 4, pp. 866-871, 1989.

[44] B. Wang, L. Chen, and J. Cheng, "New result on maximum entropy threshold image segmentation based on P system," *Optik*, vol. 163, pp. 81-85, 2018.

[45] S. B. Tambe, D. Kulhare, M. Nirmal, and G. Prajapati, "Image processing (IP) through erosion and dilation methods," 2013.

[46] I. Ragnemalm, "Fast erosion and dilation by contour processing and thresholding of distance maps," *Pattern recognition letters*, vol. 13, no. 3, pp. 161-166, 1992.

[47] P. Soille, "Erosion and dilation," in *Morphological Image Analysis*: Springer, 2004, pp. 63-103.

[48] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[49] K. Yue, F. Xu, and J. Yu, "Shallow and wide fractional max-pooling network for image classification," *Neural Computing and Applications*, vol. 31, no. 2, pp. 409-419, 2019.

[50] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Applied Intelligence*, vol. 51, no. 2, pp. 854-864, 2021.

[51] P. Angelov and E. Almeida Soares, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification," *MedRxiv*, 2020.

[52] H. Roth *et al.*, "Rapid artificial intelligence solutions in a pandemic-the COVID-19-20 lung CT lesion segmentation challenge," 2021.

[53] S. A. Harmon *et al.*, "Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets," *Nature communications*, vol. 11, no. 1, pp. 1-7, 2020.

[54] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618-626.



Cheng Kang received the B.S. degree from Weifang University of Science and Technology, China in 2013, and the Master degree from Shenzhen University, China in 2016. He is currently a PhD student, researcher of RCI (Research Center for Informatics) for AI, and researcher of AID Lab, FEL, CTU. Personal Page: <https://chengkang520.github.io/about/>.



Prof. **J.M. Górriz** received the B.Sc. degree in physics, the B.Sc. degree in electronic engineering from the University of Granada, Spain, in 2000 and 2001, respectively, Ph.D. degree from the University of Cádiz, Spain, in 2003, and the Ph.D. degree from the University of Granada in 2006. He is currently a Full Professor with University of Granada.



Shui-Hua Wang (SMIEEE) received B.S. degree from Southeast University, China, M.S. degree from the City College of New York, USA, and Ph.D. degree from Nanjing University, China, in 2017. She was an Assistant Professor with Nanjing Normal University China, during 2013–2018. She is a Lecturer with the University of Leicester, U.K. Her research interests include deep learning and information fusion.



Xujing Yao received her B.S degree in Machinal Engineering from Nanjing Tech University, China and Technological University Dublin, Ireland, in 2018 and began her PhD study in the University of Leicester from January in 2019. Her research interests include deep learning and medical image analysis.



Prof. **Yu-Dong Zhang** received his BE in Information Sciences in 2004 and MPhil in Communication and Information Engineering in 2007, from Nanjing University of Aeronautics and Astronautics. He received a Ph.D. degree in Signal and Information Processing from Southeast University in 2010. Now he serves as a professor at University of Leicester, UK. His research interests include deep learning and medical image analysis.



Ziquan Zhu received his M.S degree in Civil and Coastal Engineering from the University of Florida in 2020, the United States, and began his Ph.D. study at the University of Leicester from November 2021. His research interests include deep learning and medical image analysis.