A fuzzy fine-tuned model for COVID-19 diagnosis

Esmi, Nima; Golshan, Yasaman; Asadi, Sara; Shahbahrami, Asadollah; Gaydadjiev, Georgi

DOI
10.1016/j.compbiomed.2022.106483

Publication date
2023

Document Version
Final published version

Published in
Computers in Biology and Medicine

Citation (APA)
Esmi, N., Golshan, Y., Asadi, S., Shahbahrami, A., & Gaydadjiev, G. (2023). A fuzzy fine-tuned model for COVID-19 diagnosis. *Computers in Biology and Medicine*, *153*, Article 106483. https://doi.org/10.1016/j.compbiomed.2022.106483

Computers in Biology and Medicine 153 (2023) 106483

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

A fuzzy fine-tuned model for COVID-19 diagnosis

Nima Esmi [a], Yasaman Golshan [b], Sara Asadi [b], Asadollah Shahbahrami [a,b,*], Georgi Gaydadjiev [a]

[a] Faculty of Science and Engineering, University of Groningen, Netherlands
[b] Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran

ARTICLE INFO

Keywords:
COVID-19
Deep learning
Blind/Referenceless image spatial quality evaluator
Fuzzy pooling
Weighted multi-class cross-entropy

ABSTRACT

The COVID-19 disease pandemic spread rapidly worldwide and caused extensive human death and financial losses. Therefore, finding accurate, accessible, and inexpensive methods for diagnosing the disease has challenged researchers. To automate the process of diagnosing COVID-19 disease through images, several strategies based on deep learning, such as transfer learning and ensemble learning, have been presented. However, these techniques cannot deal with noises and their propagation in different layers. In addition, many of the datasets already being used are imbalanced, and most techniques have used binary classification, COVID-19, from normal cases. To address these issues, we use the blind/referenceless image spatial quality evaluator to filter out inappropriate data in the dataset. In order to increase the volume and diversity of the data, we merge two datasets. This combination of two datasets allows multi-class classification between the three states of normal, COVID-19, and types of pneumonia, including bacterial and viral types. A weighted multi-class cross-entropy is used to reduce the effect of data imbalance. In addition, a fuzzy fine-tuned Xception model is applied to reduce the noise propagation in different layers. Quantitative analysis shows that our proposed model achieves 96.60% accuracy on the merged test set, which is more accurate than previously mentioned state-of-the-art methods.

1. Introduction

COVID-19 is an infectious disease caused by acute respiratory syndrome generated by the SARS-CoV-2 virus. The first report of such a virus dates back to December 2019. The virus is easily transmitted from person to person through airborne particles or tiny droplets caused by coughing or sneezing. Symptoms of this disease appear after a few days and include cough, fever, and severe respiratory involvement [1]. As of February 2022, about 400 million people have been infected, and 5.7 million have died worldwide [2,3]. Because no definitive cure has yet been found for the disease, ways to identify and isolate the affected person and begin treatment immediately are of considerable importance. There are currently several ways to diagnose COVID-19 disease. Reverse Transcription-Polymerase Chain Reaction (RT-PCR) testing [4] and methods based on radiographic images [5] are examples of these ways. Problems such as errors in results, high cost of test kits, and lack of advanced medical imaging tools and related specialists [4,6] have led researchers to consider the use of low-cost, affordable Chest X-ray (CXR) images [7–12].

Diagnosis of COVID-19 disease through CXR images is a matter of classification in machine learning. Deep learning methods that use Convolutional Neural Networks (CNNs) have received much attention [13]. Many recent studies [1,9,14] have either reused or modified pre-trained models. Because not much time has passed since the emergence of COVID-19, existing research has encountered obstacles to the poor quality of datasets, such as noise in images, including salt and pepper, Poisson, speckle, and Gaussian noises. [15–19], and severe limitations on the volume of training data. Also, many datasets are highly imbalanced. In general, as stated in [20], many existing datasets related to COVID-19 do not have the necessary quality to be used in learning models. This lack of quality can lead to bias in the models and false accuracy in the output. In addition, many available datasets only provide binary classification between COVID-19 and normal cases. Due to these limitations, the results obtained from the models have some problems, such as generalization errors and overfitting, which reduce the accuracy of models in the actual operating environment [21]. Another problem occurs when a lung disease, such as pneumonia, affects the lungs similarly to COVID-19. Many studies have not distinguished COVID-19 positive and positive pneumonia images [8,12,13,21–36]. When multi-class classification is performed, the severity of the problems related to the small dataset volume and the presence of noise is more apparent. In addition, there are some obstacles when using pre-trained CNN models in relation to noisy images. The main parts of a CNN include four

* Corresponding author at: Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran.
E-mail addresses: nima.esmi@gmail.com (N. Esmi), ysmngolshan@gmail.com (Y. Golshan), sara.asadi.khomami@gmail.com (S. Asadi), shahbahrami@guilan.ac.ir (A. Shahbahrami), g.gaydadjiev@rug.nl (G. Gaydadjiev).

https://doi.org/10.1016/j.compbiomed.2022.106483
Received 28 September 2022; Received in revised form 16 December 2022; Accepted 25 December 2022
Available online 4 January 2023

basics: convolution, pooling, fully connected layer, and softmax. The pooling layer, specifically max pooling, cannot prevent the propagation of incoming noises to other layers, and sometimes the intensity of the noises increases [37].

To address these challenges, we focus on reducing the effect of noise propagation and improving the data quality. First, to deal with the different types of noise that may be received [33], a Blind/Referenceless Image Spatial QUality Evaluator (BRISQUE) is applied, which measures image quality. Second, two standard datasets are combined to improve the dataset size and diversity. These two items can also help reduce the bias problem. Combining two datasets made it possible to classify into three categories: COVID-19, normal, and various types of pneumonia, including bacterial and viral types. Third, the weighted cross-entropy technique is used to deal with data imbalance. With the help of weighted multi-class cross-entropy, the problem of overfitting is addressed. Numerous pre-trained models have already been proposed to classify images in the realm of deep learning. One of the most popular and common examples is Xception, provided by Google. To address the error propagation problem in different layers of the pre-trained CNN model, we replace the max pooling operation with the fuzzy pooling in the Xception model. In general, our contributions are as follows:

- By combining two datasets, each consisting of two classes, multi-class classification of normal, COVID-19 and types of pneumonia is provided. In addition, in order to remove noisy images, preprocessing BRISQUE operations are applied.
- By applying fuzzy pooling instead of max pooling in the model, the model's accuracy is improved.
- Weighted multi-class cross-entropy is used in the classification stage to reduce the effect of data imbalance. By using weighted multi-class cross-entropy, the learning process occurs faster, and the possibility of overfitting is reduced. Experimental results show that our proposed model archives an accuracy of 96.6%.

The rest of this article is organized as follows. In Section 2, background and related work are presented. The proposed approach is described in detail in Section 3. The experimental results and evaluation are discussed in Section 4. Finally, concluding remarks have been made in Section 5.

2. Background and related work

In this section, we first discuss the diagnosis of diseases with the help of computer technology and specifically through medical images. Various problems related to the use of datasets, such as imbalance and noisy images, binary and multi-class classification, and also the problem of noise propagation in CNN models, are investigated. The latest works of literature in the multi-class classification between normal, COVID-19 disease, and types of pneumonia through CXR images are considered.

2.1. Background and motivation

The use of images in the medical field has a wide range of applications. These applications can include the diagnosis of various cancers or types of cardiovascular, liver, respiratory diseases, etc. [38–44]. The development of artificial intelligence-based solutions such as deep learning has led to the use of medical images in the accurate and rapid diagnosis of various diseases. Examples of these efforts to find COVID-19 are presented in [33]. Fig. 1 depicts examples of attempts to diagnose COVID-19 disease. The continuous lines indicate the techniques which are used in this paper, while the dotted lines are related to other methods. Jawahar, M., et al. [5] reviews the different methods of predicting and diagnosing COVID-19. In this study, data related to single voice [7], lung ultrasound, Computerized Tomography (C.T.), and X-ray images were used. With the emergence of the coronavirus pandemic, researchers turned their attention to the rapid and accurate diagnosis of this disease through medical images such as C.T. and X-rays. Many studies on the diagnosis of COVID-19 disease through imaging had two limitations. First, some studies were only able to classify the two states of COVID-19 and normal [8,12,13,21–36]. Second, some studies have used C.T. images, which have problems such as high cost compared to CXR, limited access in less privileged areas, the need for a specialist to examine the images, and so on [45–48]. As the disease spread rapidly around the world, there was not enough opportunity for researchers to create a dataset with sufficient data volumes and standard images. Since the presence of unusable data in datasets can reduce the accuracy of the models that use those datasets, the quality of the data must be ensured before entering the data into the model. There are several ways to assess image quality. Some of them need a reference image for evaluation. Some also work without a reference image [49]. To assess the quality of medical images in the real world, particularly in relation to the evaluation of images related to COVID-19 datasets, due to the lack of a reference image, only non-reference image quality evaluation methods can be used [50,51]. Dubbed Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) is one of the methods of evaluating images without reference, which evaluates the quality of images only using mathematical equations such as Gaussian [52]. After filtering the images, there is a need for a model to teach the classification of images into categories such as COVID-19, normal, or pneumonia images. There are many pre-trained models for image classification [53]. Some of them have been widely used in research related to the classification of COVID-19 images due to their simplicity, small size, and acceptable accuracy [54–58]. One of the most common options for the classification of COVID-19 images is the Xception model, which is also supported by Google and Keras. A simple view of this model is shown in Fig. 2. The data first passes through the entry flow. After that, they pass through the middle flow (in this intermediate stage, it repeats itself 8 times) and finally through the exit flow.

As mentioned earlier, convolution and pooling layers are two main parts of a CNN model. Convolution and pooling are used for feature extraction and spatial dimension reduction, respectively. Making minor acceptable changes to different model parts can improve model performance for specific applications. As can be seen in Fig. 2, the Xception model uses four max pooling operators. If an error occurs for various reasons, such as noise in images, the uncertainty created cannot be filtered by max pooling. This uncertainty will be spread to the following layers. One way to reduce this problem is to make the pooling function more flexible. For example, fuzzy pooling can be flexibly used to reduce dimensions [59,60]. Another problem with deep learning in diagnosing COVID-19 disease is the imbalanced datasets. An imbalanced dataset can lead to problems such as overfitting and unreliability of results. Various solutions have been proposed to deal with such a problem. One of these methods is oversampling, which artificially increases the class elements of the dataset in the minority. One of the disadvantages of this method is that part of the processing required to learn the model is spent on analyzing artificially created data. Opposite to the oversampling method is the undersampling, in which some data related to the majority class are removed. Since the small size of the COVID-19 dataset itself is one of the deep learning problems, omitting some data can exacerbate the problem. Another method is to use weighted cross-entropy. In this method, instead of manipulating the dataset, the weights of the minor and major classes change with different coefficients during the training [61–63].

2.2. Related work

Multi-class classification has been applied in [1,6,9–11,28,64]. A summary of some studies performed for the multi-class diagnosis of COVID-19 disease using CXR images related to the two years ending in 2022 is shown in Table 1. In addition to being more technically complex than binary classifications, such classifications require more

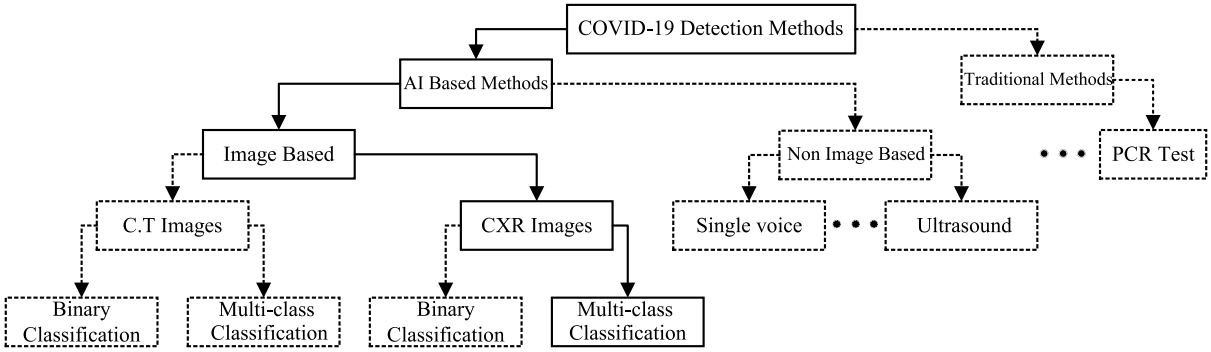

Fig. 1. Different methods for diagnosing COVID-19 disease. The continuous lines indicate the techniques which are considered in this paper, and the dotted lines are related to other methods.

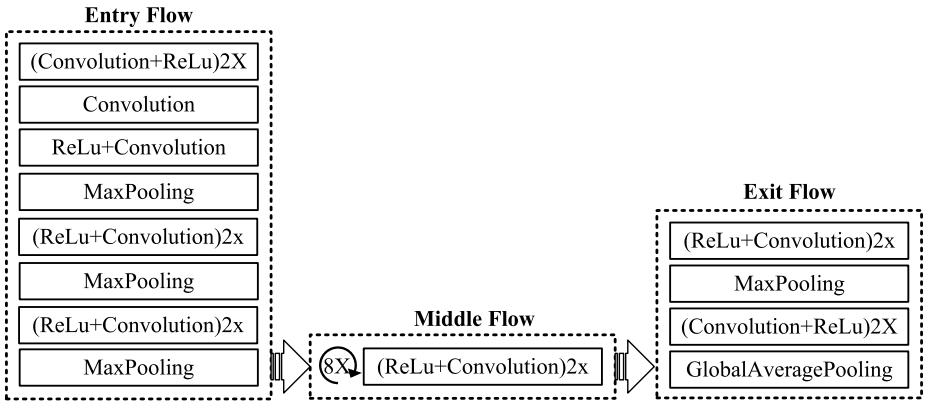

Fig. 2. A simplified design of the Xception model, which has three phases; entry, middle, and exit flows, respectively with four max pooling.

Table 1
Multi-class classification of COVID-19 detection using CXR images. The fourth column shows the total number of dataset elements and the distribution of the number of COVID-19, normal and pneumonia elements in three classes.

Ref.	Year	Model	Cases/Cov./No./Pn.	COVID/Total	Accu.	F1
[1]	2020	Xception	1125/125/500/500	11.11%	97.40%	0.97
[2]	2020	Bayes-SqueezeNet	5949/76/1583/2839	1.28%	98.30%	0.98
[64]	2020	MobileNetV2 + SqueezeNet	458/295/65/98	64.41%	99.27%	0.97
[6]	2020	COVID-Net	13,962/358/8066/5538	2.56%	93.30%	–
[9]	2021	EfficientNet	16634/5634/6000/5000	33.87%	93.48%	0.93
[10]	2021	EDL-COVID	15477/573/8851/6053	3.69%	95.00%	–
[11]	2021	EfficientNetB4	2878/219/1341/1345	7.61%	98.04%	0.98
[28]	2021	ULNet	2878/219/1314/1345	7.61%	95.25%	0.93
[14]	2021	DenseNet	15498/590/8851/6057	3.81%	92.00%	–
[30]	2021	MFBCNNC-II	6939/2313/2313/2313	33.33%	94.37%	–
[65]	2021	Fuzzy based VGG16 +Xception+InceptionV3	3975/752/1639/1584	18.92%	93.81%	–
[66]	2021	The Exemplar model (fuzzy tree transform)	435/135/150/150	31.03%	97.01%	–
[67]	2021	xViTCOS	8214/2358/1583/4273	28.71%	96.00%	0.96
[68]	2021	Fuzzy model with 9 layers	1551/224/504/700	14.44%	93.53%	0.93
[31]	2022	SARS-Net	15254/358/6045/8851	2.34%	97.60%	–
[69]	2022	CMTNet	33951/2513/10270/21168	7.40%	98.79%	0.74

accurate and larger datasets to provide reliable results. In [1,6,14, 28,30,31], transfer learning has been used to deal with the lack of X-ray images in the dataset. In [2], SqueezeNet was used as a pre-trained model. Different offline augmentation was used to deal with the lack of images in the dataset. In [64], MobileNetV2 and SqueezeNet models were introduced. The Support Vector Machine (SVM) method was used for multi-class classification. Nigam, B., et al. [9] used VGG16, DenseNet121, Xception, NASNet, and EfficientNet models. NASNet performed with the highest accuracy, with 93.48%. In [14] three pre-trained models named DenseNet, InceptionV3, and Inception-ResNetV4 were used, with DenseNet providing the highest accuracy in multi-class classification with 92%. Similarly, [30] used the LeNet-5, VGG-16, and ResNet-18 models. Okolo, G.I., et al. in [11] examined eleven deep convolutional neural network architectures under three different approaches.

Imbalanced datasets and the presence of noise in images are two major challenges for classification [6,10,28,68,69]. In [6], a dataset with 13975 images was used. Nevertheless, this dataset was so imbalanced that only 2.56% of the images were related to COVID-19. Tang, S., et al. in [10,14] used a combination of multiple models as an ensemble learning architecture to overcome data-related problems. Malhotra, A., et al. in [69] used two different datasets with a total of 33,951 images. Although an accuracy of 98.7% was achieved, only 7% of dataset images were related to COVID-19, and the precision value was only 64.12%. In [70], on the one hand, the problems caused by the small number of dataset elements for learning were investigated.

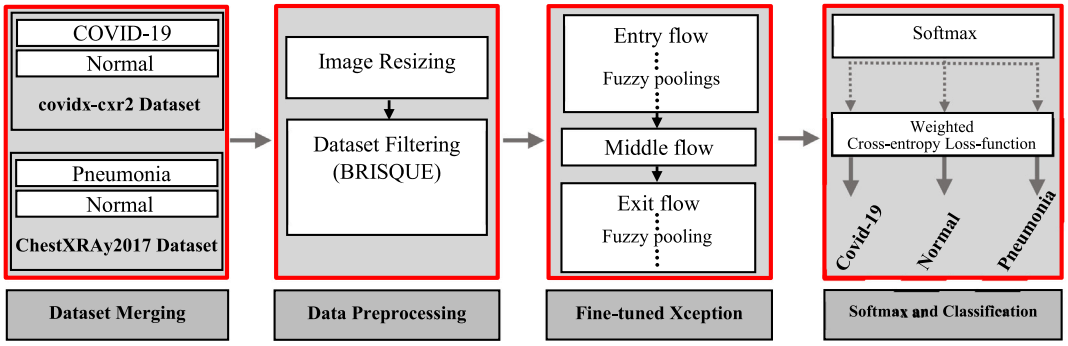

Fig. 3. The proposed model for multi-class classification of CXR images which has four steps, dataset merging, data preprocessing, fine-tuned Xception, and softmax and classification.

On the other hand, with the help of conditional generated adversarial networks, it was tried to reduce the effect of undesirable effects. It is shown that the lack of accurate data can lead to the extraction of uninformative features so that the model shows unrealistically high accuracy. In [68], fuzzy logic was used to improve the pixel intensity and reduce background noise. By doing so, two problems of deficiency and the quality of datasets were considered. In [11], a new model, which is a combination of CNNs and graph convolutional networks was proposed with an accuracy of 97.60%. Bhowal, P., et al. in [65] also used ensemble learning by use of three different sets of fuzzy measures to leverage the classification accuracy. In [66] a three-level fuzzy-tree is constructed, which acts like convolution. In [67], a multi-stage transfer learning technique was used to deal with the problem of data scarcity. Since uncertainty and imprecision are integral to image data in datasets, some literature proposes fuzzy classifiers. Some examples are presented in [71–74]. It is shown in [72] that similar images can be classified into similar classes using the fuzzy classifier.

3. Proposed approach

The proposed model is depicted in Fig. 3, which has four steps. In the first step, CXR images of two datasets are received, including images of COVID-19 disease, normal images, and images of various types of pneumonia. Then, some damaged and noisy data are removed in the preprocessing data stage. The BRISQUE plays the role of a preliminary stage to enhance the quality of care and reduce the dimensionality of the dataset, which can decrease the probability of overfitting and improve model accuracy [75–77]. In the third step, an Xception learning model is used, which is fine-tuned using fuzzy pooling. Among the various CNN models, Xception is chosen as a representative model in this paper. In addition, the Xception model provides some separable and optional parts that ease max pooling replacement by fuzzy pooling [58]. In the fourth stage, the usual softmax operation is applied to the output of the Xception model to prepare the model for classifying images into three categories. Then the selection of one of the categories entered from the softmax is performed. Afterward, in case of a false positive or false negative occurrence in any of the classes, based on the frequency of that class in the dataset, the weight of classification in cross-entropy is changed for the next stage of learning. In the following sections, each step is discussed in detail. In addition, Table 2 lists the all used notations throughout this section.

3.1. Data merging

In the data merging step, CXR images are obtained from two different datasets. One of the datasets includes normal images and images related to types of pneumonia [78]. The second dataset contains normal CXR and COVID-19 images [79]. By performing this integration, multi-class classification between normal images, COVID-19, and types of pneumonia becomes possible.

Table 2
Nomenclatures and their respective description.

Symbol	Description
$\overline{f}_v$	Fuzzy sets
U	All output values for activation functions of the convolution layer
x	Replaces softmax values
V	Number of fuzzy sets ($\overline{f}_v$)
me_v	Membership functions for fuzzy sets
T	Volume of feature maps
T^n	A spatial feature map
z	The total number of feature maps
n	Depth of a certain feature map
k	Pooling window size (k×k)
d_c	The number of elements (CXR images) in class c
p	Volume of patch
p^n	A spatial patch
$p^n_{i,j}$	A set of volume patches at depth n and location i and j
fp^n_v	Fuzzy patch
$s^n_{fp_v}$	Fuzzy algebraic sum
fp'	The highest value of $s^n_{fp_v}$.
p'^n	Defuzzification
C	The number of classes (in this case, 3)
a_c	cth element of the estimated vector for the class c(1,2,3)
b_c	cth element of the normalized ground truth vector
c	Is an integer $c \in [1, C]$
$\vec{z}_{in}$	The elements of the input vector to the softmax function
$L(a, b)$	Cross-entropy loss function
$LW(a, b)$	Weighted cross-entropy loss function
W_C	Cross-entropy loss function weight for class c
$Inp(i, j)$	Image intensity in pixel (i, j)
$\widehat{Inp}(i, j)$	Image luminance in pixel (i, j)
N	Width and Height of an image
μ	Local mean field of original image
σ	Local variance field of original image
$Hor(i, j)$	MSCN horizontal coefficient
$Ver(i, j)$	MSCN vertical coefficient
$Dia1(i, j)$	MSCN left-diagonal coefficient
$Dia2(i, j)$	MSCN right-diagonal coefficient

3.2. Data preprocessing

A summary of the steps in the preprocessing stage is presented in Algorithm 1.

Algorithm 1 Steps related to the preprocessing stage

```
Receive image from Dataset
if image format= .png then
    Convert to .jpg
end if
Change image dimensions to 299 × 299 pixel
Call BRISQUE (Figure 6)
if BRISQUE index ≥ 38 then
    Delete image
end if
Send the image to Fine-tuned Xception stage
```

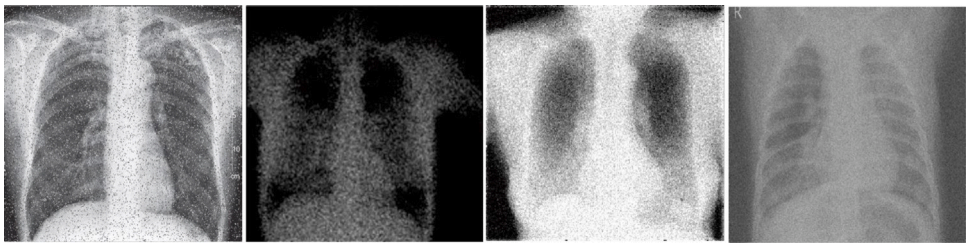

Fig. 4. Different types of noise in CXR images. From left to right, salt and pepper, Poisson, speckle, and Gaussian noises, respectively.

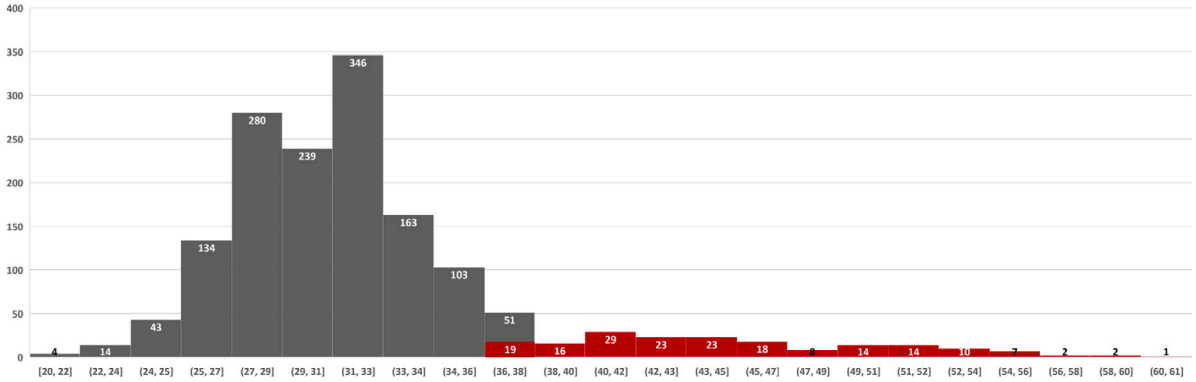

Fig. 5. Distribution of BRISQUE scores on a sample of 1544 CXR images. Values greater than 38 means that those images have much more noises in comparison to others, which are shown in red color.

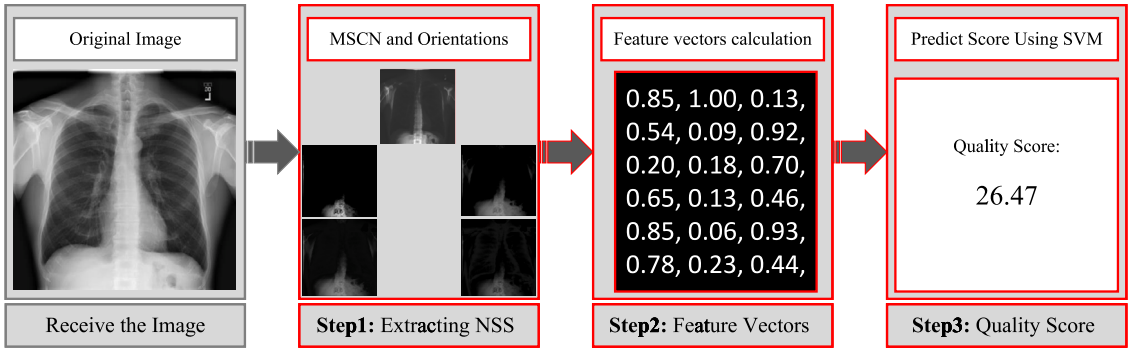

Fig. 6. Different steps of the BRISQUE algorithm, extracting Natural Scene Statistics (NSS), feature vector calculation, and predicting the quality score.

In the preprocessing stage, images are first received from the dataset. Different formats were converted to .jpg. The size of the images is then reduced to 299 by 299 pixels. The next issue is dealing with noisy images. As shown in Fig. 4, four common noises in CXR images include salt and pepper, Poisson, speckle, and Gaussian noise are deleted. In order to remove noises, the BRISQUE filter which is a noise detection and quality measurement tool without a reference image has been applied. The evaluation index between zero for images without noise and up to one hundred means the strongest possible amount of noise. Fig. 5 shows the distribution of BRISQUE scores on a sample of 1544 images used in this article. Images with a BRISQUE index above 38, which were obtained with trial and error that accounted for about 10% of all images, are removed. By doing this, the scattering of noisy images is reduced. Fig. 6, shows the three main steps of the BRISQUE algorithm, which are: extracting Natural Scene Statistics (NSS), feature vectors, and quality scores, respectively [52]. In Fig. 7, pseudocode of the BRISQUE algorithm is depicted, which includes steps and computational complexity. Images that pass through the BRISQUE filter are taken to the next (Fine-tuned Xception) stage. In the following sections, each of these steps, extracting NSS, feature vector calculation, and quality score prediction is explained.

Step 1: Extracting Natural Scene Statistics The intensity distribution of pixels for a noiseless image is different from a noisy or distorted image. This difference in distributions becomes more apparent when pixel intensities are normalized. According to Fig. 8, after normalization, the noiseless image pixel intensity distribution follows the Gaussian distribution. In the BRISQUE algorithm, Mean Subtracted Contrast Normalization (MSCN) is used for normalization. As shown in Eq. (1), in order to calculate the MSCN coefficients, the intensity of the image $Inp(i,j)$ in pixel (i,j) must be converted to the luminance of the image $\widehat{Inp}(i,j)$ in pixel (i,j) [52].

$$\widehat{Inp}(i,j) = \frac{Inp(i,j) - \mu(i,j)}{\sigma(i,j) + 1} \tag{1}$$

where i=1, 2, …, N and j=1, 2, …, N. N is the height and width of an image.

The μ and σ functions are the local mean field and the local variance field of the original image, respectively. "Local" means that calculations are performed at any moment for a specific pixel and its adjacent pixels. Through MSCN, the normalized intensity of pixels is measured. However, the comparison between noiseless and noisy images is not limited to the intensity distribution of pixels, and the relationship between a pixel and its neighboring pixels should also be considered.

$\mu(i,j)$:
for $(i = 0;\ i < n;\ i++)$
for$(j = 0;\ j < n;\ j++)$
Mu = conv2d(Inp)

$\sigma(i.j)$:
for $(i = 0;\ i < n;\ i++)$
for$(j = 0;\ j < n;\ j++)$
Sigma = $\sqrt{conv2d(Inp_{(i.j)} - Mu_{(i.j)})^2}$

$\hat{I}np(i.j)$:
$$\frac{Inp_{(i.j)} - Mu_{(i.j)}}{Sigma_{(i.j)} + 1}$$

Step1 $O(n^4)$

Pair-wise Products:

```
int shift = {(0,1),(1,0),(1,1),(-1,1)};
    for(int itr_shift = 1; itr_shift <= 4; itr_shift++){
      int* reqshift = shifts[itr_shift - 1]
      for (i = 0; i < n; i++)
        for(j = 0; j < n; j++)
          ShiftArr[i][j] = Original Image Arr[i + reqshift[0]][j + reqshift[1]];
      }
    }
}
```

Step1 $O(n^3)$

Generalized Gaussian Distribution
Return GGD fit to MSCN coefficients (Variance)

Asymmetric Generalized Gaussian Distribution:
Return AGGD fit to Horizontal/ Vertical/ Diagonal left-Diagonal Right Pairwise Products
(Mean, Left Variance, Right Variance)

Step2 $O(n)$

SVM:
Calculate SVM and Return BRISQUE Score

Step3 $O(n^3)$

Fig. 7. The pseudocode of the BRISQUE algorithm includes steps and their computational complexity related to extracting natural scene statistics (step1), feature vectors calculation (step2), and BRISQUE score calculation (step3).

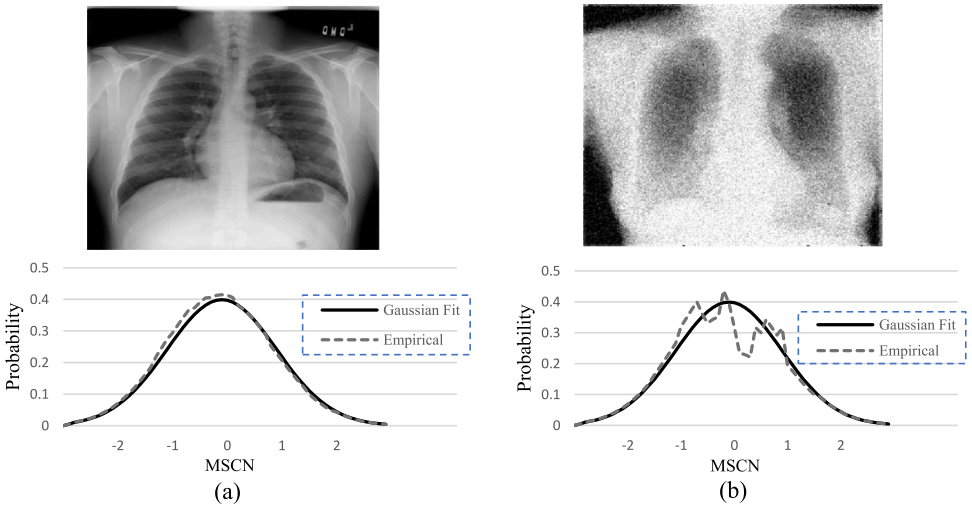

Fig. 8. Comparison between Mean Subtracted Contrast Normalization (MSCN) of an image and the Gaussian distribution. (a) example of CXR image with minimal noise; (b) with added noise.

The statistical relationships between neighboring pixels are computed using the empirical distributions of pairwise products of neighboring MSCN coefficients. For this purpose, four directions can be considered: horizontal, vertical, left diagonal, and right diagonal. In Eq. (2), the relevant equations of each of the orientations are presented [52].

$$\begin{aligned} Hor(i,j) &= \widehat{Inp}(i,j)\,\widehat{Inp}(i,j+1) \\ Ver(i,j) &= \widehat{Inp}(i,j)\,\widehat{Inp}(i+1,j) \\ Dia1(i,j) &= \widehat{Inp}(i,j)\,\widehat{Inp}(i+1,j+1) \\ Dia2(i,j) &= \widehat{Inp}(i,j)\,\widehat{Inp}(i+1,j-1) \end{aligned} \tag{2}$$

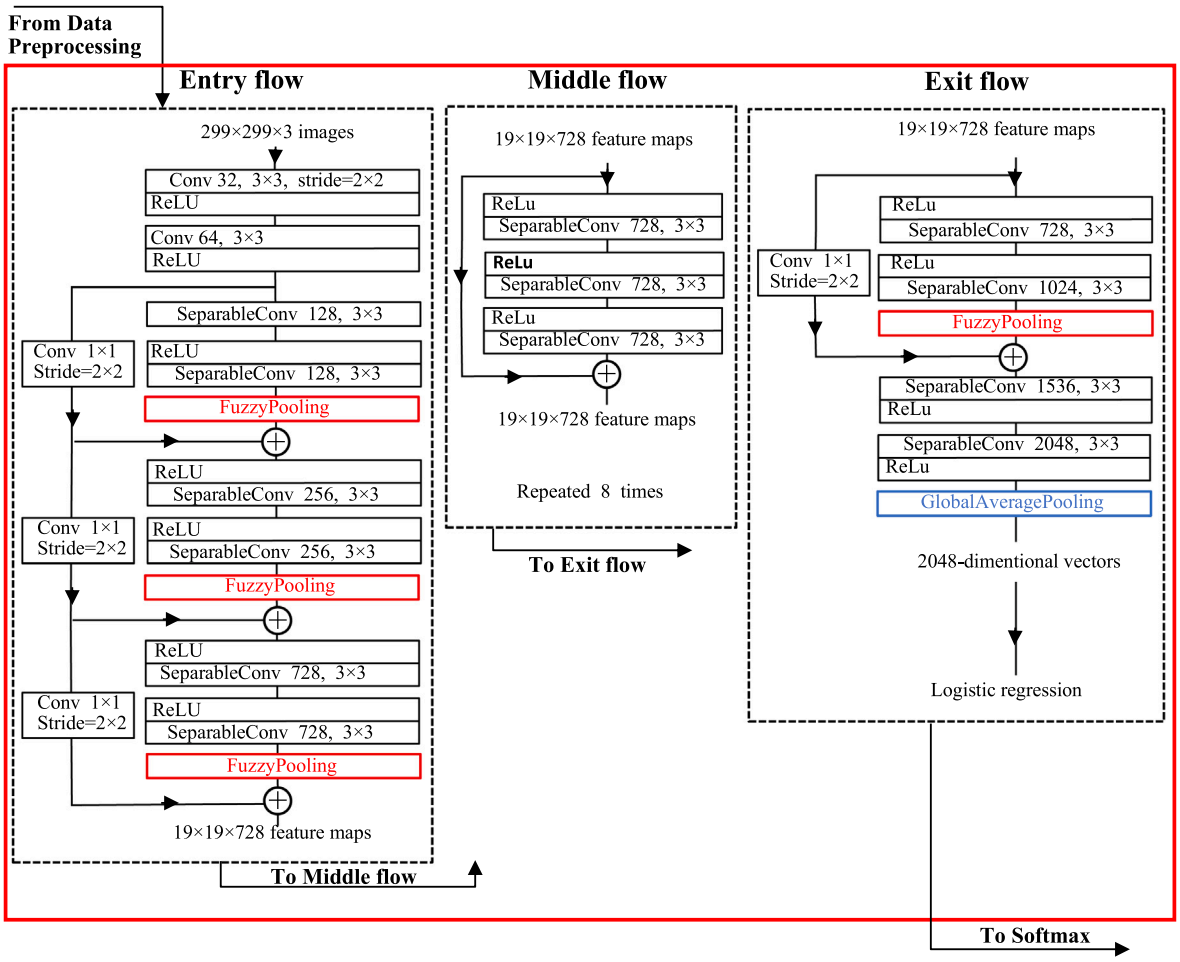

Fig. 9. The extended version of the Xception model which already mentioned in Fig. 2, where four max poolings are replaced by fuzzy poolings.

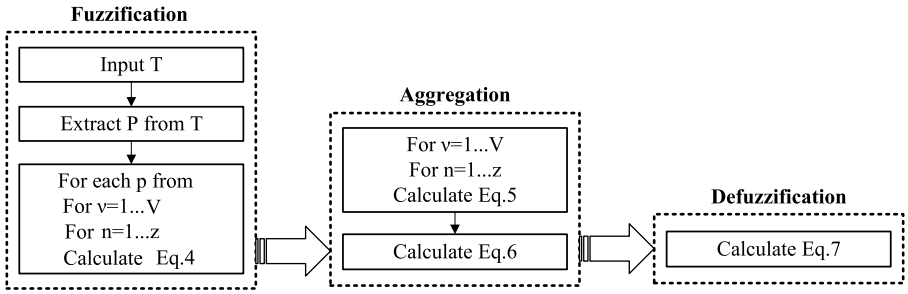

Fig. 10. Three steps of fuzzy pooling in which fuzzification decomposes inputs into fuzzy sets, aggregation starts pooling operations, and defuzzification assigns a fixed value to the aggregated fuzzy set.

Step2: Feature vectors calculation

In this step, five images that have been obtained using Eqs. (1) and (2) are received and a feature vector is produced, which contains an array of elements. The first element is obtained by matching the MSCN image with the Generalized Gaussian Distribution (GGD), for the variance. The other elements include matching each of the images related to orientations with Asymmetric Generalized Gaussian Distribution (AGGD). Each orientation consists of four parameters: shape, mean, left variance, and right variance.

Step3: Predict image quality score using SVM

In the third step, the trained LIBSVM package which is an integrated software for support vector classification, [80] is used to predict the score. In this case, feature vectors are first scaled to values between −1 and 1. Then these values are mapped to numbers in the range of 0 to 100. Zero represents images without noise, and as the numbers go up, the amount of noise also increases.

3.3. Fine-tuned Xception

Fig. 9 shows a fine-tuned model of the original Xception model. A simple design of this model has already been shown in Fig. 2. In the fine-tuned Xception stage, the data is entered into an Xception model. In this model, instead of max pooling, fuzzy pooling is used to prevent the propagation of uncertainty (due to the presence of various noises in the images) in different layers of the model. As shown in Fig. 10, the fuzzy pooling operation can be divided into three main parts, which include fuzzification, aggregation, and defuzzification. Fig. 11 shows the fuzzy pooling architecture. A fuzzy set is a class of objects. This

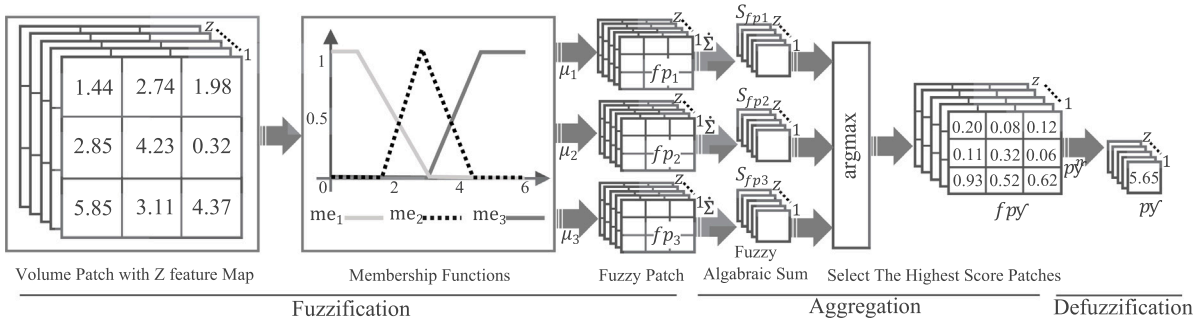

Fig. 11. Fuzzy pooling architecture includes volume patch with Z feature map, membership functions, fuzzy patches, algebraic sum, and select the highest score patches.

class consists of a sequence of degrees of membership. A fuzzy set is characterized by a membership function (characteristic) so that each element in a patch is assigned a membership degree in the range of 0–1. To extract a set of patches from the input volume, a sliding window with dimensions K by K (here K = 3 and stride 2) is applied to the features. Fuzzy Pooling function is based on fuzzy c-means idea. In fuzzy c-means, the dataset is divided into clusters and each data point in a cluster is assigned a certain degree. But in fuzzy Pooling, clusters are formed in patches, and a set of patches form a volume.

Operations are performed on feature maps extracted from a convolution layer. Uncertainty in the values of these feature maps can be modeled with a fuzzy set. This model is presented in Eq. (3) [60].

$$\overline{f}_v = \left(\langle x, me_v(x) | x \in U \rangle\right), \quad v = 1, \ldots, V, \tag{3}$$

In Eq. (3), U is equal to all output values for activation functions of the convolution layer and $x \in \mathbb{R}$ replaces softmax values. Variable V represents the number of fuzzy sets ($\overline{f}_v$) and me_v is equal to membership functions for fuzzy sets, that is used to represent small, medium, and large values of fuzzy patches. The calculation method of me_v is presented in [60]. T represents a set of z-features, $T = \{T^n | n = 1, 2, \ldots, z\}$. Each T input volume can contain a set of specific p^n patch volumes extracted from T^n feature maps, so that: $p = \{p^n | n = 1, 2, \ldots, z\}$. Now, if we consider $p^n_{i,j}$ to be a set of volume patches at depth n and location i and j, $i = 1...k$, $j = 1...k$, $n = 1...z$. In this case, for each p^n patch, the fuzzy patch fp^n_v is defined as Eq. (4) [60].

$$fp^n_v = me_v(p^n) = \begin{pmatrix} me_v\left(p^n_{1,1}\right) & \cdots & me_v\left(p^n_{1,k}\right) \\ \cdots & \ddots & \cdots \\ me_v\left(p^n_{1,k}\right) & \cdots & me_v\left(p^n_{k,k}\right) \end{pmatrix} \tag{4}$$

The pooling operation begins with the aggregation of the fuzzy patch values. A fuzzy algebraic sum (Eq. (5)) is used for this purpose [60].

$$s^n_{fp_v} = \sum_{i=1}^{k} \sum_{j=1}^{k} fp^n_{v_{i,j}}, \quad n = 1, 2..., z. \tag{5}$$

fp' (Eq. (6)) is used to select patches with the highest degree of confidence that obtains the highest value of $s^n_{fp_v}$ [60].

$$fp' = \left\{ fp'^n_v = fp^n_v | v = argmax\left(s^n_{fp_v}\right), \; n = 1, 2, \ldots, z \right\}. \tag{6}$$

In the last step, the dimensions must be reduced. Dimensions on each patch are reduced by defuzzification (Eq. (7)) [60].

$$p'^n = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} \left(fp'^n_{i,j} \cdot p^n_{i,j}\right)}{\sum_{i=1}^{k} \sum_{j=1}^{k} fp'^n_{i,j}}, \; n = 1, 2, \ldots, z. \quad where \; p' = \left\{ p'^n | \; n = 1, 2, \ldots, z \right\}. \tag{7}$$

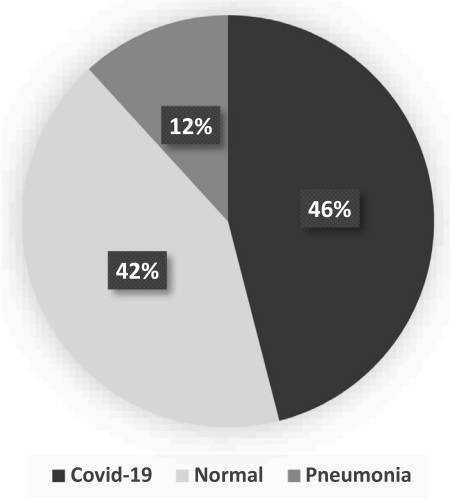

Fig. 12. Distribution of COVID-19, normal and pneumonia classes in the non-filtered dataset.

3.4. Softmax and classification

As mentioned earlier, one of the problems with COVID-19 image datasets is data imbalance, which can lead to overfitting, and reducing model performance and accuracy. As shown in Fig. 12, the dataset is used to learn the diagnosis of COVID-19 disease from normal conditions or pneumonia samples including imbalance classes. The imbalance in data classes can make a negative impact on the learning process. Although two datasets are combined to solve the imbalanced data problem, it is still not sufficiently balanced. Meanwhile, fuzzy pooling increases the complexity of the model [60], which can lead to an increase in the probability of overfitting. Weighted multi-class cross-entropy can reduce the effect of overfitting [73]. To deal with the data imbalance problem, we use weighted multi-class cross-entropy. In softmax, a set of values are generated for classes (Eq. (8)). In this case, the loss value is calculated as Eq. (9). In these equations, C represents the number of classes, b_c is equal to the cth element of the normalized ground truth vector, and a_c is equal to the cth element of the estimated vector for class c(1,2,3). Since learning means weight change and can be done in the cross-entropy loss function, a value of W is added to the equation of cross-entropy loss function Eq. (10) [63]. As shown in Eq. (11), the value of W is directly related to the inverse of the ratio of the number of elements in a class to the total data elements.

$$Softmax\vec{z}_{in} = \frac{exp^{\vec{z}_{in}}}{\sum_{c=1}^{3} exp^{z_c}} \tag{8}$$

$$L(a, b) = -\sum_{c=1}^{3} b_c \times \log\left(a_c\right) \tag{9}$$

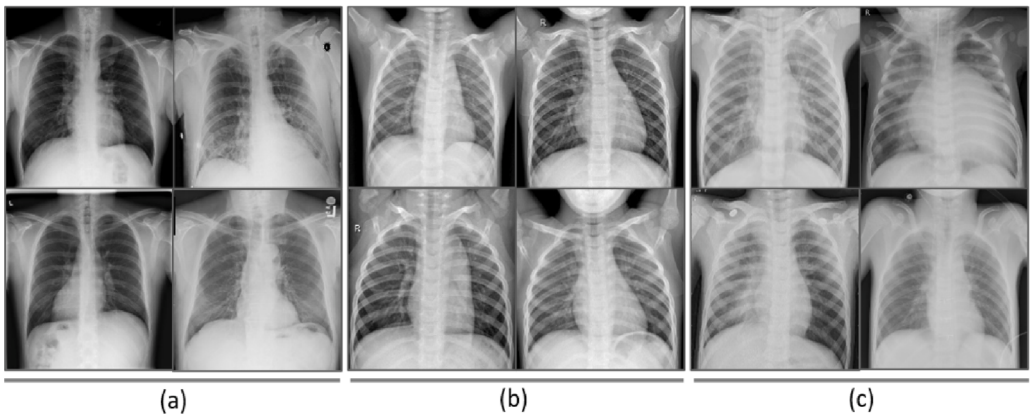

Fig. 13. Samples of three different image data categories from CXR-related datasets. (a) COVID-19 samples, (b) Normal samples, and (c) Pneumonia chest images.

Table 3
The number of COVID-19, normal, and pneumonia samples in two datasets, covidx-cxr2 and chestXRAy2017, before and after their combination.

covidx-cxr2 dataset			
	COVID-19	Normal	Pneumonia
Non-Filter	16690	13793	0
Filtered	14270	12796	0
ChestXRAy2017 dataset			
	COVID-19	Normal	Pneumonia
Non-Filter	0	1534	4273
Filtered	0	1412	3788
Overall			
	COVID-19	Normal	Pneumonia
Non-Filter	16690	15327	4273
Filtered	14270	14208	3788

$$LW\,(a,b) = -\sum_{c=1}^{3} W_c \times b_c \times \log\left(a_c\right) \tag{10}$$

$$W_c = \frac{\sqrt{d_c}+1}{2d_c} \tag{11}$$

4. Experimental results and evaluation

4.1. Datasets

As shown in Fig. 13, our experiment involves the separation of three different classes of CXR images. Since there was no single dataset that had a significant volume of all three COVID-19, normal and pneumatic classes at the time of writing, we used the combination of two datasets [78,79]. By combining two datasets and applying filtering operations we improved our dataset with three classes. In Fig. 14 our dataset is compared with some other datasets in terms of the number of samples in each category that have been used in various literature. The BRISQUE operations have been applied to our dataset for removing noisy images. Table 3 shows the details of two datasets, covidx-cxr2 and ChestXRAy2017 and their merging before and after applying filtering operations. As can be seen in this table, some noisy images have been removed in each category.

4.2. Environmental setup and criteria

Experiments have been performed on the Keras API and the Google colab-notebook. The number of image samples in the training, validation, and test phases for both non-filtered and filtered combined datasets are depicted in Table 4. We use the confusion matrix to evaluate the proposed techniques using criteria of precision, recall, F1-Score, and accuracy which their equations are represented in (12), (13), (14), and (15).

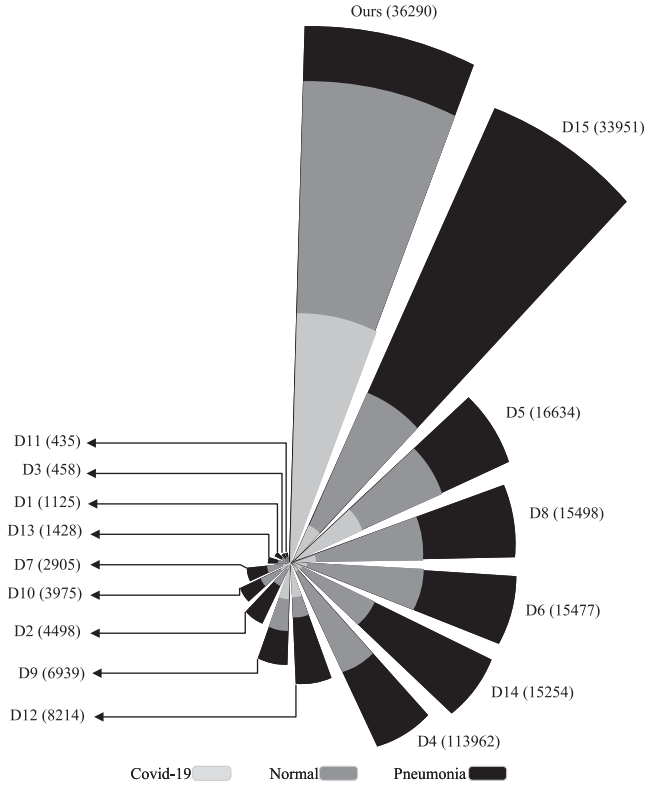

Fig. 14. A comparison between the used dataset in this paper with some other datasets which have been used for multi-class classification of COVID-19, from the number of all images and the ratio of classes point of view D1: [1], D2: [2], D3: [64], D4: [6], D5: [9], D6: [10], D7: [11], D8: [14], D9: [30], D10: [65], D11: [66], D12: [67], D13: [68], D14: [31], D15: [69].

$$Precision = \frac{TP}{TP+FP} \tag{12}$$

$$Recall = \frac{TP}{TP+FN} \tag{13}$$

$$F1-Score = \frac{2.Recall.Precision}{Recall+Precision} \tag{14}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{15}$$

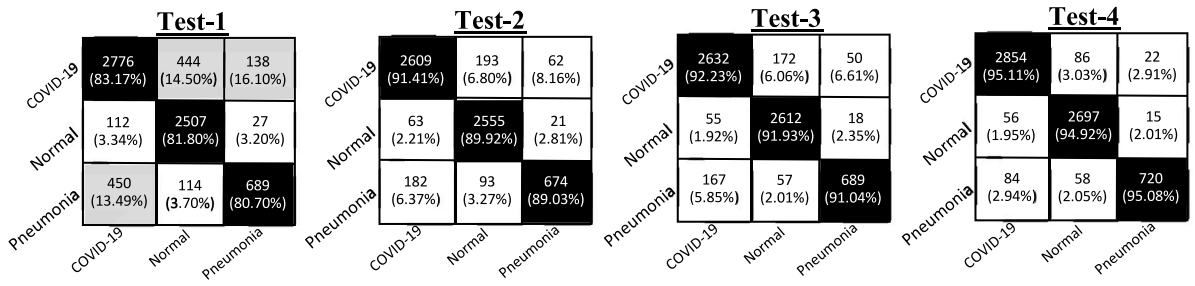

Fig. 15. Confusion matrix for four test experiments, non-filtered dataset, filtered dataset, fuzzy pooling, and weighted cross-entropy.

Table 4
Number of training, validation and test samples in each class, COVID-19, normal, and pneumonia in non-filtered and filtered dataset.

Non-filtered combined dataset				
Class	# of images	Training	Validation	Test
COVID-19	16690	11685	1667	3338
Normal	15327	10732	1530	3065
Pneumonia	4273	2994	425	854
Filtered combined dataset				
Class	# of images	Training	Validation	Test
COVID-19	14270	9991	1425	2854
Normal	14208	9947	1420	2841
Pneumonia	3788	2656	375	757

Table 5
Details of precision, recall, F1-Score and accuracy values for four experimental tests for three classes.

Steps	Class	Precision	Recall	F1-Score	Accuracy
Non-filtered dataset (1)	COVID-19	0.83	0.83	0.83	0.84
	Normal	0.95	0.82	0.88	0.90
	Pneumonia	0.55	0.81	0.65	0.89
Filtered dataset (2)	COVID-19	0.91	0.91	0.91	0.92
	Normal	0.97	0.90	0.93	0.94
	Pneumonia	0.71	0.89	0.79	0.94
Fuzzy pooling (3)	COVID-19	0.92	0.90	0.92	0.93
	Normal	0.97	0.92	0.95	0.95
	Pneumonia	0.76	0.91	0.83	0.95
Weighted cross-entropy (4)	COVID-19	0.96	0.90	0.96	0.96
	Normal	0.97	0.95	0.96	0.97
	Pneumonia	0.84	0.95	0.89	0.97

4.3. Evaluation results

In order to compare the performance of the proposed techniques with the baseline model and datasets, we have divided our experimental results into four test categories which are as follows.

1. **Non-filtered dataset:** Experimental results have been obtained for non-filtered combined datasets and classification has been performed by the softmax function.
2. **Filtered dataset:** Experimental results have been obtained for filtered combined datasets and classification has been performed by softmax function.
3. **Fuzzy pooling:** Experimental results have been obtained for filtered combined dataset and four max pooling in the Xception model are replaced with fuzzy pooling.
4. **Weighted cross-entropy:** All operations are the same as the third test and weighted cross-entropy is also used.

The confusion matrices for four test categories are depicted in Fig. 15. In the second test, by applying the BRISQUE filter to the data, the false positive value is reduced by half. A detailed comparison of the four mentioned tests using precision, recall, F1-Score, and accuracy criteria is depicted in Table 5. In addition, the overall accuracy of applying different proposed techniques is depicted in Fig. 16. The overall accuracy has been improved in each test in comparison with the first test, the baseline model. As can be seen in these results, using fuzzy pooling instead of max pooling improves the accuracy in the experimental test 3 in compared to tests 1 and 2. One of the main reasons for this behavior is that in the max pooling, most features are ignored and only one feature with the highest value is considered. As a result, the information provided by the feature map is distorted and uncertainties and noises will be able to propagate to other layers. While using fuzzy pooling we deal with uncertainties in feature values which lead to better preservation of the original feature map information. The best performance has been achieved in experimental test 4 where results have been obtained for filtered combined datasets and applying both fuzzy pooling and the weighted cross-entropy in the model.

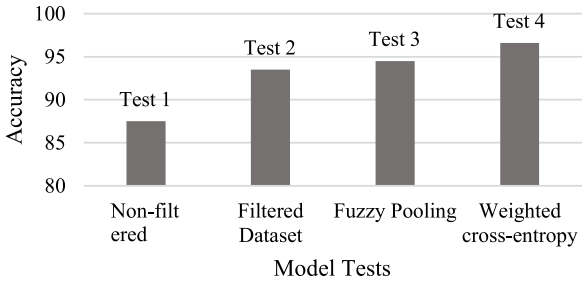

Fig. 16. Overall accuracy for four experimental tests, non-filtered dataset, filtered dataset, fuzzy pooling, and weighted cross-entropy.

The accuracy of training and validation as well as the amount of loss in training and validation for tests 3 and 4 which had the best results among other tests are shown in Fig. 17. After using weighted multi-class cross-entropy, the gap between training and validation is reduced. In addition, the probability of overfitting is also reduced. Moreover, by looking closely at both graphs, it can be seen that in the absence of weighted multi-class cross-entropy, the learning process requires more than 80 epochs to achieve stability. While by using weighted multi-class cross-entropy, after about 45 epochs, the learning process stabilizes.

5. Conclusions

In the medical field, automatic diagnosis of diseases with the help of computer tools has led to increased accuracy and reduced costs. Among its practical applications is the automatic diagnosis of lung diseases such as COVID-19.

In order to diagnose COVID-19 disease through medical images related to the chest, features must be extracted from the image. The presence of other lung diseases, such as various types of pneumonia, has made the diagnostic operation time-consuming, error-prone, and

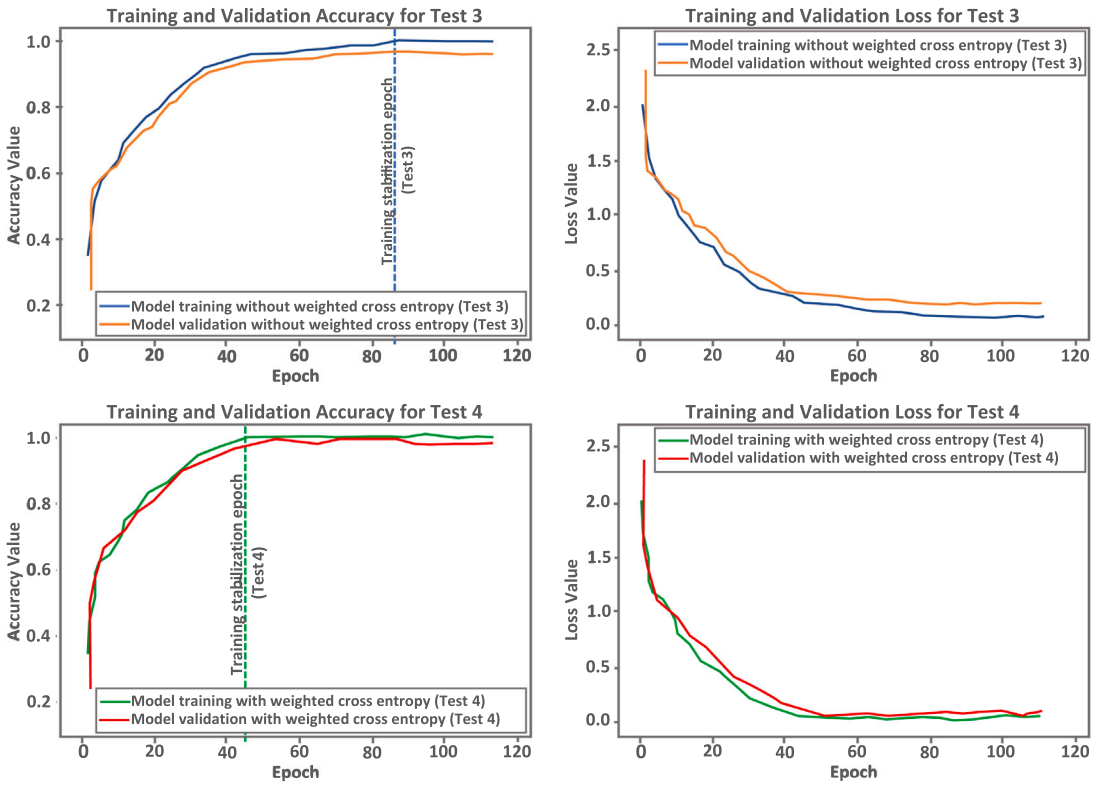

Fig. 17. Training and validation accuracy and loss for experimental tests, fuzzy pooling (3) and weighted cross-entropy (4).

requires specialized people. To overcome these limitations, a fine-tuned convolutional neural network-based model using fuzzy operations was proposed in this paper. By replacing max pooling with fuzzy pooling in the Xception CNN model, noise propagation in different layers of the model was countered. To improve the performance of pre-trained Xception model, two tasks were performed before entering the data into the model. First, two large datasets were combined to provide a dataset with 36290 cases with more class diversity. The combination of these two datasets allowed multi-class classification between normal, COVID-19, and other types of pneumonia. Second, a referenceless filtering tool (BRISQUE) was used to remove some of the noisy images in the combined dataset. In addition, a weighted multi-class cross-entropy was used at the end of the model to further address this problem. By combining all these techniques and tools, the accuracy of the proposed model reached 96.60%. After using weighted multi-class cross-entropy, the training process stabilized in fewer epochs. In addition, the gap between training and validation, which may lead to overfitting, was narrowed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] N.N. Das, N. Kumar, M. Kaur, V. Kumar, D. Singh, Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays, Innov. Res. Biomed. Eng. 43 (2) (2020) 114–119, http://dx.doi.org/10.1016/j.irbm.2020.07.001.

[2] F. Ucar, D. Korkmaz, Covidiagnosis-net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, Med. Hypotheses 140 (2020) http://dx.doi.org/10.1016/j.mehy.2020.109761.

[3] 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, 2020, http://dx.doi.org/10.48550/ARXIV.2003.11055, arXiv preprint arXiv:2003.11055.

[4] 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. 121 (2020) http://dx.doi.org/10.1016/j.compbiomed.2020.103792.

[5] M. Jawahar, L.J. Anbarasi, P. Jayachandran, M. Ramachandran, F. Al-Turjman, Utilization of transfer learning model in detecting COVID-19 cases from chest x-ray images, Int. J. E-Health Med. Commun. 13 (2) (2021) 1–11, http://dx.doi.org/10.4018/IJEHMC.20220701.oa2.

[6] L. Wang, Z.Q. Lin, A. Wong, Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, Sci. Rep. 10 (1) (2020) 1–12, http://dx.doi.org/10.1038/s41598-020-76550-z.

[7] K. Shankar, E. Perumal, V.G. Díaz, P. Tiwari, D. Gupta, A.K.J. Saudagar, K. Muhammad, An optimal cascaded recurrent neural network for intelligent COVID-19 detection using Chest X-ray images, Appl. Soft Comput. 113 (2021) http://dx.doi.org/10.1016/j.asoc.2021.107878.

[8] A. Shazia, T.Z. Xuan, J.H. Chuah, J. Usman, P. Qian, K.W. Lai, A comparative study of multiple neural network for detection of COVID-19 on chest X-ray, EURASIP J. Adv. Signal Process. 2021 (1) (2021) 1–16, http://dx.doi.org/10.1186/s13634-021-00755-1.

[9] B. Nigam, A. Nigam, R. Jain, S. Dodia, N. Arora, B. Annappa, COVID-19: Automatic detection from X-ray images by utilizing deep learning methods, Expert Syst. Appl. 176 (2021) http://dx.doi.org/10.1016/j.compbiomed.2020.103792.

[10] S. Tang, C. Wang, J. Nie, N. Kumar, Y. Zhang, Z. Xiong, A. Barnawi, EDL-COVID: ensemble deep learning for COVID-19 case detection from chest x-ray images, IEEE Trans. Ind. Inform. 17 (9) (2021) 6539–6549, http://dx.doi.org/10.1109/TII.2021.3057683.

[11] G.I. Okolo, S. Katsigiannis, T. Althobaiti, N. Ramzan, On the use of deep learning for imaging-based COVID-19 detection using chest X-rays, Sensors 21 (17) (2021) http://dx.doi.org/10.3390/s21175702.

[12] M. Umair, M.S. Khan, F. Ahmed, F. Baothman, F. Alqahtani, M. Alian, J. Ahmad, Detection of COVID-19 using transfer learning and grad-CAM visualization on indigenously collected X-ray dataset, Sensors 21 (17) (2021) http://dx.doi.org/10.3390/s21175813.

[13] A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, Pattern Anal. Appl. 24 (3) (2021) 1207–1220, http://dx.doi.org/10.1007/s10044-021-00984-y.

[14] S. Albahli, N. Ayub, M. Shiraz, Coronavirus disease (COVID-19) detection using X-ray images and enhanced DenseNet, Appl. Soft Comput. 110 (2021) http://dx.doi.org/10.1016/j.asoc.2021.107645.

[15] E.M. Paul, B. Perumal, M.P. Rajasekaran, Filters used in x-ray chest images for initial stage tuberculosis detection, in: International Conference on Inventive Research in Computing Applications, IEEE, 2018, pp. 235–239, http://dx.doi.org/10.1109/ICIRCA.2018.8597334.

[16] C.-T. Lu, M.-Y. Chen, J.-H. Shen, L.-L. Wang, C.-C. Hsu, Removal of salt-and-pepper noise for X-ray bio-images using pixel-variation gain factors, Comput. Electr. Eng. 71 (2018) 862–876, http://dx.doi.org/10.1016/j.compeleceng.2017.08.012.

[17] K.B. Khan, A.A. Khaliq, M. Shahid, J.A. Shah, A new approach of weighted gradient filter for denoising of medical images in the presence of Poisson noise, Tehnički Vjesnik 23 (6) (2016) 1755–1762, http://dx.doi.org/10.17559/TV-20150620130712.

[18] V. Vignesh, K. Kothavari, Classification and detection of lung nodules using virtual dual energy in CXR images, in: International Conference on Green Computing Communication and Electrical Engineering, IEEE, 2014, pp. 1–6, http://dx.doi.org/10.1109/ICGCCEE.2014.6922409.

[19] S. Rawat, K. Rana, V. Kumar, A novel complex-valued convolutional neural network for medical image denoising, Biomed. Signal Process. Control 69 (2021) http://dx.doi.org/10.1016/j.bspc.2021.102859.

[20] B.G. Santa Cruz, M.N. Bossa, J. Sölter, A.D. Husch, Public covid-19 x-ray datasets and their impact on model bias–a systematic review of a significant problem, Med. Image Anal. 74 (2021) http://dx.doi.org/10.1016/j.media.2021.102225.

[21] N. Alrefai, O. Ibrahim, Deep learning for COVID-19 diagnosis based on chest X-ray images, Int. J. Electr. Comput. Eng. 11 (5) (2021) 4531–4541, http://dx.doi.org/10.11591/ijece.v11i5.

[22] A. Krishnaswamy Rangarajan, H.K. Ramachandran, A fused lightweight CNN model for the diagnosis of COVID-19 using CT scan images, Automatika 63 (1) (2022) 171–184, http://dx.doi.org/10.1080/00051144.2021.2014037.

[23] S. Shambhu, D. Koundal, P. Das, C. Sharma, Binary classification of COVID-19 CT images using CNN: COVID diagnosis using CT, Int. J. E-Health Med. Commun. 13 (2) (2021) 1–13, http://dx.doi.org/10.4018/IJEHMC.20220701.oa4.

[24] T. Kaur, T.K. Gandhi, Classifier fusion for detection of COVID-19 from CT scans, Circuits Systems Signal Process. 41 (2022) 3397—3414, http://dx.doi.org/10.1007/s00034-021-01939-8.

[25] S. Serte, F. Al-Turjman, COVID-19 detection on CT scans using local binary pattern and deep learning, in: International Summit Smart City 360°, Springer, 2020, pp. 101–107, http://dx.doi.org/10.1007/978-3-030-76063-2_7.

[26] N.S. Shaik, T.K. Cherukuri, Transfer learning based novel ensemble classifier for COVID-19 detection from chest CT-scans, Comput. Biol. Med. 141 (2022) http://dx.doi.org/10.1016/j.compbiomed.2021.105127.

[27] H.A. Dehkordi, H. Kashiani, A.A.H. Imani, S.B. Shokouhi, Lightweight local transformer for COVID-19 detection using chest CT scans, in: 11th International Conference on Computer Engineering and Knowledge, IEEE, 2021, pp. 328–333, http://dx.doi.org/10.1109/ICCKE54056.2021.9721517.

[28] T. Wu, C. Tang, M. Xu, N. Hong, Z. Lei, Ulnet for the detection of coronavirus (COVID-19) from chest X-ray images, Comput. Biol. Med. 137 (2021) http://dx.doi.org/10.1016/j.compbiomed.2021.104834.

[29] S.H. Khan, A. Sohail, A. Khan, M. Hassan, Y.S. Lee, J. Alam, A. Basit, S. Zubair, COVID-19 detection in chest X-ray images using deep boosted hybrid learning, Comput. Biol. Med. 137 (2021) http://dx.doi.org/10.1016/j.compbiomed.2021.104816.

[30] J. Sun, X. Li, C. Tang, S.-H. Wang, Y.-D. Zhang, MFBCNNC: Momentum factor biogeography convolutional neural network for COVID-19 detection via chest X-ray images, Knowl.-Based Syst. 232 (2021) http://dx.doi.org/10.1016/j.knosys.2021.107494.

[31] A. Kumar, A.R. Tripathi, S.C. Satapathy, Y.-D. Zhang, SARS-net: COVID-19 detection from chest x-rays by combining graph convolutional network and convolutional neural network, Pattern Recognit. 122 (2022) http://dx.doi.org/10.1016/j.patcog.2021.108255.

[32] R.K. Gupta, N. Kunhare, R.K. Pateriya, N. Pathik, A deep neural network for detecting coronavirus disease using chest X-Ray images, Int. J. Healthcare Inf. Syst. Inform. 17 (2) (2022) 1–27, http://dx.doi.org/10.4018/IJHISI.20220401.oa1.

[33] N. Khasawneh, M. Fraiwan, L. Fraiwan, B. Khassawneh, A. Ibnian, Detection of COVID-19 from chest X-ray images using deep convolutional neural networks, Sensors 21 (17) (2021) http://dx.doi.org/10.3390/s21175940.

[34] A. Laouarem, C. Kara-Mohamed, E.-B. Bourenane, A. Hamdi-Cherif, A deep learning model for CXR-based COVID-19 detection, in: International Conference on Engineering and Emerging Technologies, IEEE, 2021, pp. 1–5, http://dx.doi.org/10.1109/ICEET53442.2021.9659492.

[35] A.M. Anter, D. Oliva, A. Thakare, Z. Zhang, AFCM-LSMA: New intelligent model based on Lévy slime mould algorithm and adaptive fuzzy C-means for identification of COVID-19 infection from chest X-ray images, Adv. Eng. Inform. 49 (2021) http://dx.doi.org/10.1016/j.aei.2021.101317.

[36] A. Das, Adaptive unet-based lung segmentation and ensemble learning with CNN-based deep features for automated COVID-19 diagnosis, Multimedia Tools Appl. 81 (4) (2022) 5407–5441, http://dx.doi.org/10.1007/s11042-021-11787-y.

[37] R. Nirthika, S. Manivannan, A. Ramanan, R. Wang, Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study, Neural Comput. Appl. 34 (7) (2022) 5321–5347, http://dx.doi.org/10.1007/s00521-022-06953-8.

[38] K. Takahashi, T. Fujioka, J. Oyama, M. Mori, E. Yamaga, Y. Yashima, T. Imokawa, A. Hayashi, Y. Kujiraoka, J. Tsuchiya, et al., Deep learning using multiple degrees of maximum-intensity projection for PET/CT image classification in breast cancer, Tomography 8 (1) (2022) 131–141, http://dx.doi.org/10.3390/tomography8010011.

[39] L. Fang, X. Wang, Brain tumor segmentation based on the dual-path network of multi-modal MRI images, Pattern Recognit. 124 (2022) http://dx.doi.org/10.1016/j.patcog.2021.108434.

[40] T. Bhardwaj, R. Mittal, H. Upadhyay, L. Lagos, Applications of swarm intelligent and deep learning algorithms for image-based cancer recognition, in: Artificial Intelligence in Healthcare, Springer, 2022, pp. 133–150, http://dx.doi.org/10.1007/978-981-16-6265-2_9.

[41] A.M. Alqudah, A. Alqudah, Improving machine learning recognition of colorectal cancer using 3D GLCM applied to different color spaces, Multimedia Tools Appl. 81 (8) (2022) 10839–10860, http://dx.doi.org/10.1007/s11042-022-11946-9.

[42] F. Tian, Y. Gao, Z. Fang, J. Gu, Automatic coronary artery segmentation algorithm based on deep learning and digital image processing, Appl. Intell. 51 (12) (2021) 8881–8895, http://dx.doi.org/10.1007/s10489-021-02197-6.

[43] H. Zamanian, A. Mostaar, P. Azadeh, M. Ahmadi, Implementation of combinational deep learning algorithm for non-alcoholic fatty liver classification in ultrasound images, J. Biomed. Phys. Eng. 11 (1) (2021) 73–84, http://dx.doi.org/10.31661/jbpe.v0i0.2009-1180.

[44] W. Chen, X. Han, J. Wang, Y. Cao, X. Jia, Y. Zheng, J. Zhou, W. Zeng, L. Wang, H. Shi, et al., Deep diagnostic agent forest (DDAF): A deep learning pathogen recognition system for pneumonia based on CT, Comput. Biol. Med. 141 (2022) http://dx.doi.org/10.1016/j.compbiomed.2021.105143.

[45] A.D. Algarni, W. El-Shafai, G.M. El Banby, A. El-Samie, N.F. Soliman, et al., An efficient CNN-based hybrid classification and segmentation approach for COVID-19 detection, Comput. Mater. Contin. 70 (3) (2022) 4393–4410, http://dx.doi.org/10.32604/cmc.2022.020265.

[46] M. Abd Elaziz, A.A. Ewees, D. Yousri, H.S.N. Alwerfali, Q.A. Awad, S. Lu, M.A. Al-Qaness, An improved marine predators algorithm with fuzzy entropy for multi-level thresholding: Real world example of COVID-19 CT image segmentation, IEEE Access 8 (2020) 125306–125330, http://dx.doi.org/10.1109/ACCESS.2020.3007928.

[47] P.S. Sajja, Application of fuzzy convolutional neural network for disease diagnosis: A case of Covid-19 diagnosis through CT scanned lung images, in: Tracking and Preventing Diseases with Artificial Intelligence, Springer, 2022, pp. 177–199, http://dx.doi.org/10.1007/978-3-030-76732-7_8.

[48] R. Kundu, P.K. Singh, S. Mirjalili, R. Sarkar, COVID-19 detection from lung CT-Scans using a fuzzy integral-based CNN ensemble, Comput. Biol. Med. 138 (2021) http://dx.doi.org/10.1016/j.compbiomed.2021.104895.

[49] L.S. Chow, R. Paramesran, Review of medical image quality assessment, Biomed. Signal Process. Control 27 (2016) 145–154, http://dx.doi.org/10.1016/j.bspc.2016.02.006.

[50] S. Masoudi, S. Harmon, S. Mehralivand, N. Lay, U. Bagci, B.J. Wood, P.A. Pinto, P. Choyke, B. Turkbey, No-reference image quality assessment of T2-weighted magnetic resonance images in prostate cancer patients, in: IEEE 18th International Symposium on Biomedical Imaging, IEEE, 2021, pp. 1201–1205, http://dx.doi.org/10.1109/ISBI48211.2021.9434027.

[51] D. Morland, P. Lalire, S. Guendouzen, D. Papathanassiou, N. Passat, A no-reference respiratory blur estimation index in nuclear medicine for image quality assessment, Medicine 98 (48) (2019) http://dx.doi.org/10.1097/MD.0000000000018207.

[52] A. Mittal, A.K. Moorthy, A.C. Bovik, No-reference image quality assessment in the spatial domain, IEEE Trans. Image Process. 21 (12) (2012) 4695–4708, http://dx.doi.org/10.1109/TIP.2012.2214050.

[53] Papers with code - imagenet benchmark (image classification), 2022, https://paperswithcode.com/sota/image-classification-on-imagenet, Accessed: 2022-03-25.

[54] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, http://dx.doi.org/10.48550/arXiv.1409.1556, arXiv preprint arXiv:1409.1556.

[55] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, http://dx.doi.org/10.1145/3065386.

[56] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[57] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[58] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.

[59] T. Sharma, V. Singh, S. Sudhakaran, N.K. Verma, Fuzzy based pooling in convolutional neural network for image classification, in: IEEE International Conference on Fuzzy Systems, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/FUZZ-IEEE.2019.8859010.

[60] D.E. Diamantis, D.K. Iakovidis, Fuzzy pooling, IEEE Trans. Fuzzy Syst. 29 (11) (2020) 3481–3488, http://dx.doi.org/10.1109/TFUZZ.2020.3024023.
[61] H. Naeem, A.A. Bin-Salem, A CNN-LSTM network with multi-level feature extraction-based approach for automated detection of coronavirus from CT scan and X-ray images, Appl. Soft Comput. 113 (2021) http://dx.doi.org/10.1016/j.asoc.2021.107918.
[62] S. Panchapagesan, M. Sun, A. Khare, S. Matsoukas, A. Mandal, B. Hoffmeister, S. Vitaladevuni, Multi-task learning and weighted cross-entropy for DNN-based keyword spotting, in: Interspeech, Vol. 9, 2016, pp. 760–764, http://dx.doi.org/10.21437/Interspeech.2016-1485.
[63] M. Akil, R. Saouli, R. Kachouri, et al., Fully automatic brain tumor segmentation with deep learning-based selective attention using overlapping patches and multi-class weighted cross-entropy, Med. Image Anal. 63 (2020) http://dx.doi.org/10.1016/j.media.2020.101692.
[64] 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) http://dx.doi.org/10.1016/j.compbiomed.2020.103805.
[65] P. Bhowal, S. Sen, J.H. Yoon, Z.W. Geem, R. Sarkar, Choquet integral and coalition game-based ensemble of deep learning models for COVID-19 screening from chest X-ray images, IEEE J. Biomed. Health Inf. 25 (12) (2021) 4328–4339, http://dx.doi.org/10.1109/JBHI.2021.3111415.
[66] T. Tuncer, F. Ozyurt, S. Dogan, A. Subasi, A novel Covid-19 and pneumonia classification method based on F-transform, Chemometr. Intell. Lab. Syst. 210 (2021) http://dx.doi.org/10.1016/j.chemolab.2021.104256.
[67] A.K. Mondal, A. Bhattacharjee, P. Singla, A. Prathosh, xViTCOS: Explainable vision transformer based COVID-19 screening using radiography, IEEE J. Transl. Eng. Health Med. 10 (2021) 1–10, http://dx.doi.org/10.1109/JTEHM.2021.3134096.
[68] A. Nandal, M. Blagojevic, D. Milosevic, A. Dhaka, L.N. Mishra, Fuzzy enhancement and deep hash layer based neural network to detect Covid-19, J. Intell. Fuzzy Systems 41 (1) (2021) 1341–1351, http://dx.doi.org/10.3233/JIFS-210222.
[69] A. Malhotra, S. Mittal, P. Majumdar, S. Chhabra, K. Thakral, M. Vatsa, R. Singh, S. Chaudhury, A. Pudrod, A. Agrawal, Multi-task driven explainable diagnosis of COVID-19 using chest X-ray images, Pattern Recognit. 122 (2022) http://dx.doi.org/10.1016/j.patcog.2021.108243.
[70] S. Karakanis, G. Leontidis, Lightweight deep learning models for detecting COVID-19 from chest X-ray images, Comput. Biol. Med. 130 (2021) http://dx.doi.org/10.1016/j.compbiomed.2020.104181.
[71] X. Gu, F.-L. Chung, S. Wang, Bayesian Takagi–Sugeno–Kang fuzzy classifier, IEEE Trans. Fuzzy Syst. 25 (6) (2016) 1655–1671.
[72] M. Versaci, G. Angiulli, P. Crucitti, D. De Carlo, F. Laganà, D. Pellicanò, A. Palumbo, A fuzzy similarity-based approach to classify numerically simulated and experimentally detected carbon fiber-reinforced polymer plate defects, Sensors 22 (11) (2022).
[73] T. Zhou, F.-l. Chung, S. Wang, Deep TSK fuzzy classifier with stacked generalization and triplely concise interpretability guarantee for large data, IEEE Trans. Fuzzy Syst. 25 (5) (2016) 1207–1221.
[74] Z. Chen, Z. Zhang, R. Zhu, Y. Xiang, P.B. Harrington, Diagnosis of patients with chronic kidney disease by using two fuzzy classifiers, Chemometr. Intell. Lab. Syst. 153 (2016) 140–145.
[75] A. Tompe, K. Sargar, X-Ray image quality assurance, in: StatPearls [Internet], StatPearls Publishing, 2020.
[76] X. Ying, An overview of overfitting and its solutions, in: Journal of Physics: Conference Series, Vol. 1168, (2) IOP Publishing, 2019, http://dx.doi.org/10.1088/1742-6596/1168/2/022022.
[77] How to mitigate overfitting with dimensionality reduction, 2021, https://towardsdatascience.com/how-to-mitigate-overfitting-with-dimensionality-reduction-555b755b3d66, Accessed: 2022-11-16.
[78] Labeled optical coherence tomography (OCT) and chest X-Ray images for classification, 2021, https://data.mendeley.com/datasets/rscbjbr9sj/2, Accessed: 2022-03-26.
[79] COVIDx CXR-2, 2021, https://www.kaggle.com/andyczhao/covidx-cxr2, Accessed: 2022-03-26.
[80] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27, http://dx.doi.org/10.1145/1961189.1961199.