Detection of COVID-19 in noisy X-ray images using learning-to-augment incorporated noise-robust deep CNN

Adel Akbarimajd[1*], Ali Asghar Neshat[2], Mohammad Arafat Hussain[3], Mohammad Momeny[4*]

[1]Department of Electrical and Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran

[2]Department of Environmental Engineering, Esfarayen Faculty of Medical Science, Esfarayen, Iran

[3]Boston Children's Hospital, Harvard Medical School, Boston, MA 02115

[4]Department of Computer Engineering, Yazd University, Yazd, Iran

*Email of Corresponding Authors:

akbarimajd@uma.ac.ir (A. Akbarimajd)

mohamad.momeny@gmail.com (M. Momeny)

Abstract

Deep convolutional neural networks (CNNs) are used for the detection of COVID-19 in X-ray images. The detection performance of deep CNNs may be reduced by noisy X-ray images. To improve the robustness of a deep CNN against impulse noise, we propose a novel CNN approach using adaptive convolution, with the aim to ameliorate COVID-19 detection in noisy X-ray images without requiring any preprocessing for noise removal. This approach includes an impulse noise-map layer, an adaptive resizing layer, and an adaptive convolution layer to the conventional CNN framework. We also used a learning-to-augment strategy using noisy X-ray images to improve the generalization of a deep CNN. We have collected a dataset of 2,093 chest X-ray images including COVID-19 (452 images), non-COVID pneumonia (621 images), and healthy ones (1,020 images). The architecture of pre-trained networks such as SqueezeNet, GoogleNet, MobileNetv2, ResNet18, ResNet50, ShuffleNet, and EfficientNetb0 has been modified to increase their robustness to impulse noise. Validation on the noisy X-ray images using the proposed noise-robust layers- and learning-to-augment strategy-incorporated ResNet50 showed 2% better classification accuracy compared with state-of-the-art method.

Keywords: Convolutional neural network, impulse noise, X-ray image classification, adaptive convolution, COVID-19.

1. Introduction

1.1. Detection of COVID-19 in X-ray Images

Coronavirus disease 2019 (COVID-19) has a devastating effect on public health, industry, and global economy. One major need to fight the pandemic is to have the ability to detect COVID-19 cases early, such as via the chest X-ray image examination. Prior studies suggest that chest X-ray images have beneficial diagnostic features as adjuvant diagnostic tool in COVID-19 compared to RT-PCR and can be useful to detect and initiate treatment early [1]. Deep learning algorithms such as deep convolutional neural networks (CNNs) previously demonstrated great promise in various disease diagnosis, often better than expert clinicians [2]. Thus, detection of COVID-19 in chest X-ray images using deep learning can also be used as a potential tool for evaluating and monitoring COVID severity [1-6].

1.2. Classification of X-ray Images using CNN

A CNN is an effective tool for image classification, which has been used in various fields such as health, economics, and agriculture [4-8]. Last year, various types of CNNs were extensively used in COVID-19 detection in medical images. For example, to detect COVID-19 in chest X-ray and computed tomography (CT) images, Jia et al. [2] used two variants of CNN, namely, improved-MobileNet and improved-ResNet. These deep CNNs were designed to dynamically combine features from different layers, a property that the baseline MobileNet and ResNet lacked. The improved-MobileNet has been used in the detection of COVID-19, viral and bacterial pneumonia (i.e., non-COVID pneumonia), and healthy images. Likewise, the improved-ResNet has been employed to discriminate COVID-19, non-COVID pneumonia, and healthy images. These approaches achieved an accuracy of 99.6% on chest X-ray images and 99.3% on the CT images. Thakur et al. also used deep CNN on X-ray images to detect COVID-19 [9]. This model was trained for binary classification on a database of X-ray images, containing 1,917 COVID-19 and 1,960 healthy cases. This method achieved a classification accuracy of 99.64%, F-measure of 99.59%, and receiver operating characteristics (ROC) of 100%. Munusamy et al. designed a CNN architecture by combining the Fractal blocks and U-Net [10] to classify X-ray images [11], and demonstrated better classification performance compared to state-of-the-arts such as ResNet50 [12], Xception [13], and InceptionResNetV2 [14]. In addition, their model was easily trainable on chest X-ray images. An ensemble model of ResNet50 Error Correcting Output Code (ECOC) was developed by Pathan et al. for the detection of COVID-19 in chest X-ray images [15]. The ensemble model included CNNs, which were optimized using Grey Wolf Optimizer [16] and Whale Optimization [17]. They achieved a multiclass classification accuracy of 98.8%, when the model classified chest X-ray images among COVID-19, healthy, and viral pneumonia cases. Mostafiz et al. proposed a hybrid method of CNN and discrete wavelet transform to detect COVID-19 in chest X-ray images [18]. After a preprocessing operation of X-ray image enhancement and segmentation, image features were extracted by deep CNN and discrete wavelet transform. Afterwards, the optimum features with minimum redundancy and maximum relevance were selected via the recursive feature elimination process. Finally, a random forest-based bagging method was used for the COVID-19 detection task, which demonstrated a classification accuracy of 98.5%.

1.3. Impulse Noise in X-ray Images

Chest X-ray images often get corrupted by the impulse (salt and pepper) noise [19-24]. This corruption is typically caused by a malfunctioning X-ray receiver, bit errors in X-ray image transmission, and faulty memory locations in hardware. The impulse noise corrupts pixel intensities in a X-ray image, causing the corrupted pixel having either the maximum or minimum gray level value. The bipolar impulse noise is defined as:

, (1)

where denotes the intensity of an impulse noise-corrupted pixel in a X-ray image. If , intensity will be seen as the brightest dot on the X-ray image and will be seen as the darkest dot. On the other hand, if either or , then the noise is of unipolar type. Finally, if , then the impulse noise will be similar to salt and pepper having a randomly distributed value. When the impulse noise reduces the quality of X-ray images considerably, detection of COVID-19 in the corrupted X-ray image becomes difficult. To address this problem, Lu et al. developed a method for impulse noise removal using a weighted neighbor pixel-based gain factor adaption [19]. In this method, all pixels in a selected window are sorted and grouped based on the gray level variation. After grouping the pixels, the median value and distribution ratio are calculated for each group to estimate the values of the gain factors. These gain factors eventually are used as wights for neighboring pixels that replace the noise-corrupted pixel. Using a fuzzy switching median filter and the concept of information sets, Arora et al. introduced a filter to remove the impulse noise from images [23]. This method works in two phases: the first phase detects pixels corrupted by the impulse noise, and the second phase operates the filter on noisy pixels using an adaptive switching criterion. Satti et al. proposed an impulse noise removing filter using min-max average pooling technique [22]. This approach showed an increase in peak signal to noise ratio (PSNR) of 1.2 dB in the restored medical images compared to the noisy counterparts. The classification performance of a CNN gets deteriorated, when input images are corrupted by the impulse noise [25]. Preprocessing input images to remove noise before feeding to a CNN usually improves the classification performance of the CNN. However, state-of-the-art filtering-based noise removing approaches, discussed above, are often time- and computation-intensive.

1.4. Proposed Method

To increase the robustness of a CNN to the impulse noise, we propose a novel CNN framework including a built-in noise-map layer, an adaptive resizing layer and an adaptive convolution layer. We summarize our technical contributions as:

1. We introduce a noise-map layer module in the CNN framework that generates a binary noise-map indicating the spatial location of noisy and normal pixels in an image, which ultimately helps to improve the task performance of a CNN by letting it avoid the noisy pixels during training. This module also helps to avoid preprocessing of images to remove noise.
2. We also introduce an adaptive image resizing module in the CNN framework that can simultaneously resize an image and remove noise from at the front end of a CNN.
3. Further, we introduce an adaptive convolution layer module that incorporates the noise-map from the first module into the convolution estimation function, which helps to effectively shut off remaining noisy pixels in the input image.
4. We show the efficacy of the proposed deep CNN framework on clinical X-ray images of COVID-19, non-COVID pneumonia and healthy subjects.

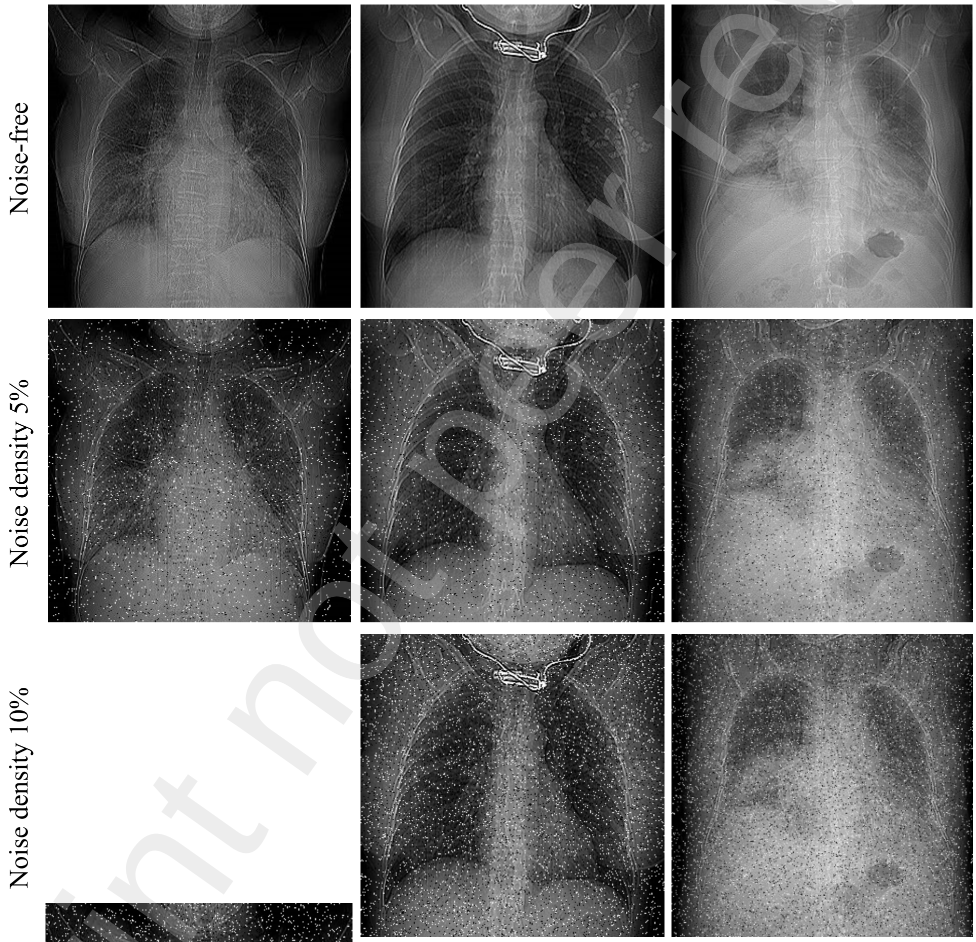
The remainder of this paper is structured as follows. We describe our dataset in Section 2. In Section 3, we detail the novel components of a CNN. Extensive experimentation and corresponding results are discussed in Section 4. The conclusion is presented in Section 5.

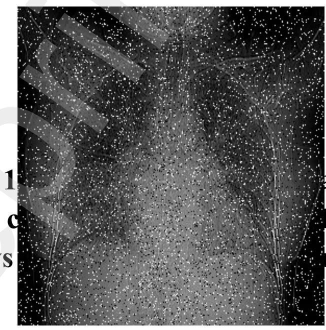
2. Data

We accessed a database of 2,093 chest X-ray images in the Esfarayen University of Medical Science, Esfarayen, Iran. Table 1 summarizes the patients' diagnoses. All the X-ray images were in Joint Photographic Experts Group (JPEG) file format. We resized all X-ray images to a common input size for the pre-trained CNNs (i.e., SqueezeNet, GoogleNet, MobileNetv2, ResNet18, ResNet50, ShuffleNet, and EfficientNetb0). We show samples of collected chest X-ray images (noisy and noise-free) for COVID-19, healthy, and non-COVID pneumonia cases in Figure 1.

Table 1: Summary of Patients' Diagnoses.

Diagnosis	Number of subjects/patients	Data collection timeline (years)
COVID-19	452	2020-2021
Non-COVID pneumonia	1,020	2018-2021
Healthy	621	2018-2021





(b)
Healthy

(c)
Non-COVID pneumonia

Figure 1 … ages of (a) COVID-19, (b) healthy, and (c) non-COVID pneumonia c… he first row contains the noise-free images. The second and third rows … ages with the noise density of 5% and 10%, respectively.

3. Methodology

In this section, detection of COVID-19 in noisy X-ray images using noise-robust deep CNN based on adaptive convolution is presented which classifies impulsive noisy images without any preprocessing for noise removal. Figure 2 illustrates the general process of the proposed method for detection of COVID-19 in noisy images.

3.1. Impulse Noise Detection

The pixels corrupted by the impulse noise can be detected using the analysis of local statistical properties of an image. In this paper, we use a switching technique-based fuzzified degree [26] to detect noise-free and noisy pixels in an image. Figure 3 illustrates the pipeline of 4-step noise detection procedure.

Step 1. Let denotes a selected processing window, which is a small patch of the corrupted image centered at location ( ). The size of the processing window is pixels. The processing window is further divided into pixels overlapped sub-windows (see Figure 4).

Figure 2: The schematic diagram of the proposed method.

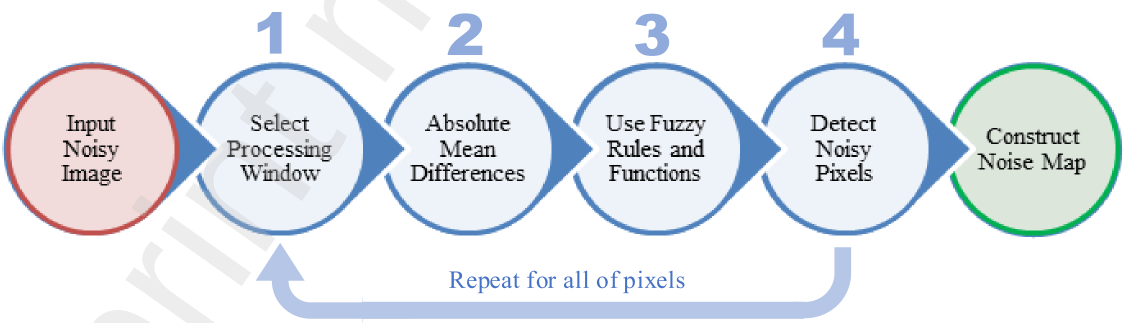


Figure 3: The 4-step pipeline of noise detection using the analysis of local statistical properties of an image [26].

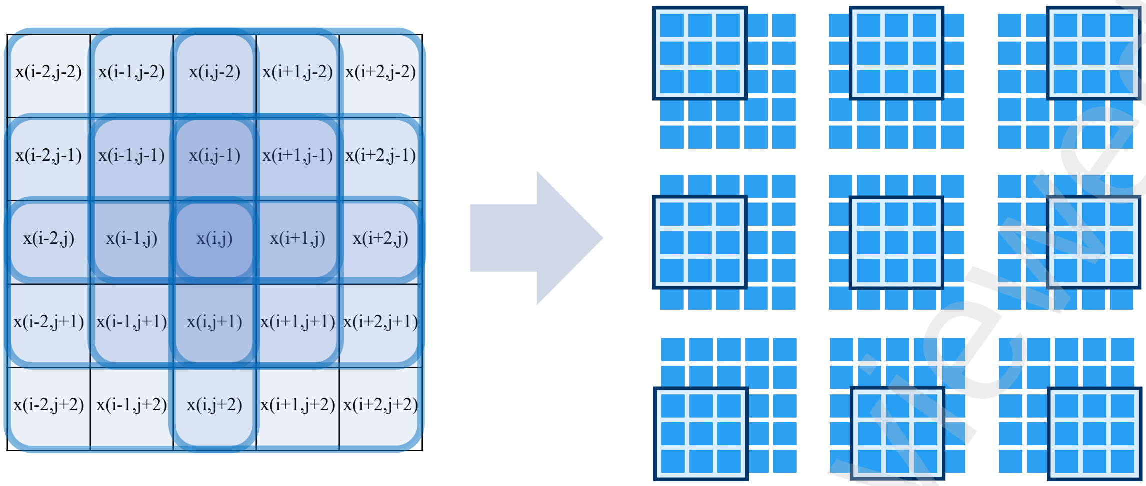


Figure 4: The pixels processing window is divided into pixels overlapped sub-windows.

Step 2. In this step, we calculate the absolute mean differences. Let indicates sub-window for . Medians of nine sub-windows are estimated as [26]:

(2)

These median values of nine sub-windows (in equation 2) are put in ascending order as [26]:

(3)

The absolute mean differences are then calculated as:

(4)

(5)

where, and are employed to determine noisy pixels of the image.

Step 3. In this step, we used fuzzy logic to detect if the current pixel is noisy or noise-free. To do this, we assign the degree of impulsiveness to each pixel by using fuzzy gradient values [26]. To distinguish noisy pixels from edges, the difference between the gradients is classified into nondeterministic features (or). Figure 5 shows the fuzzy membership functions and that represent fuzzy set and fuzzy set , respectively. The fuzzy membership functions are defined as [26]:

(6)

(7)

, (8)

(9)

(10)

(11)

where and are threshold parameters. The fuzzy membership degree is defined as [26]:

(12)

Step 4. In the fourth step, the switching technique based fuzzified degree [26] is applied to detect the noisy pixels. As shown in Figure 6, if , the interrogated pixel is noise-free. Otherwise, the interrogated pixel is noisy. Thus, a noise-map, can be defined as:

(13)

where is the location of the interrogated pixel. The noise-map for a whole image is then constructed by examining all the pixels in that particular image using the above-mentioned technique.

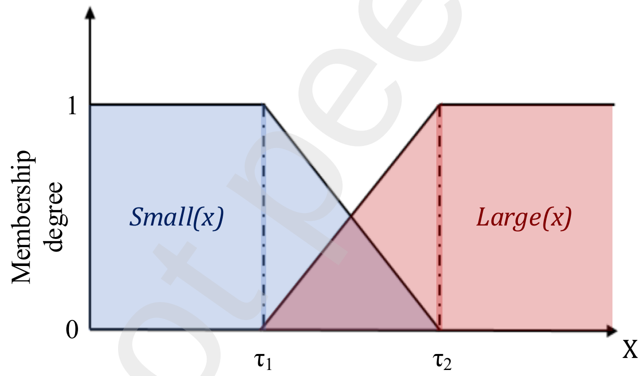


Figure 5: Fuzzy membership functions ‘ ’ or ‘ ’.

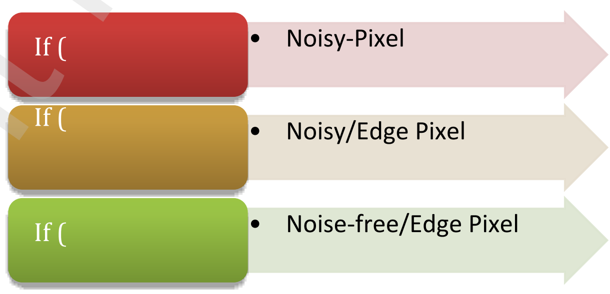


Figure 6: Noise detection rules.

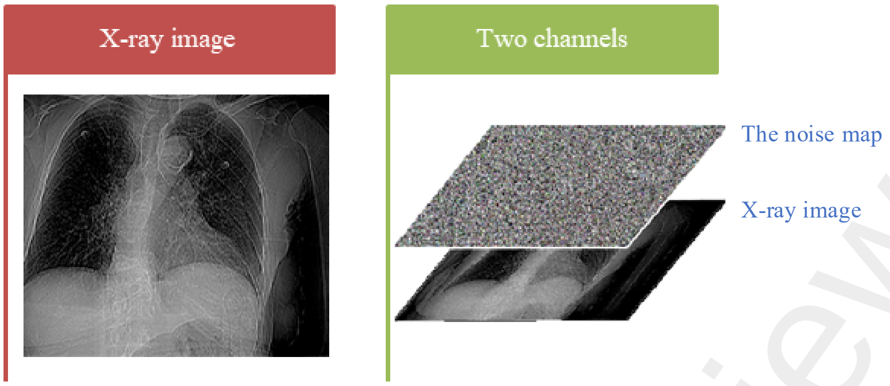


Figure 7: Two channels for each image.

In this paper, to make our CNN framework robust to impulse noise, we use the estimated noise map as the second channel for the corresponding X-ray image, when fed to the CNN. As shown in Figure 7, each image contains two channels: noise-map channel, grayscale X-ray channel.

3.2. Adaptive Resizing

All input images to a CNN (e.g., GoogleNet, MobileNetv2, ResNet18, ResNet50, ShuffleNet, EfficientNetb0, etc.) usually have a common dimension (e.g., pixels). Also, if a model is pretrained, the dimension of the input images during finetuning should match to the dimension of the images on which the model is pretrained. Since the models we use in this paper are pretrained, we need to resize our X-ray images so that their dimension matches to the dimension of the images used in pretraining. In this paper, rather than using a conventional interpolation-based image resizing approach, we adopted an adaptive image resizing approach, which is more robust on noisy images. To illustrate the mechanism of this resizing approach, we demonstrate resizing two pixels noisy images with low and high noise density, respectively, to pixels noise-reduced images in Figure 8. In the first step, an image is divided into blocks for subsampling (Figure 8b). We can see in Figure 8c that if the pixel value in the resized image is taken from the central pixel of the corresponding block, then noisy pixel values from the original image are easily passed to the resized image. To avoid this issue, we adopt the adaptive resizing [25] technique, where noisy pixel values do not get passed to the resized image (Figure 8d). After resizing images using corresponding noise-maps, we also resize the noise-maps so that the updated noise-map size matches the updated image size. Assuming that w is a set of candidate pixels to resize in the selected sub-window, updated noise-mapa are obtained using following.

If (all of pixels in w is noisy) then

Set the corresponding coordinates of updated noise-map to noisy.

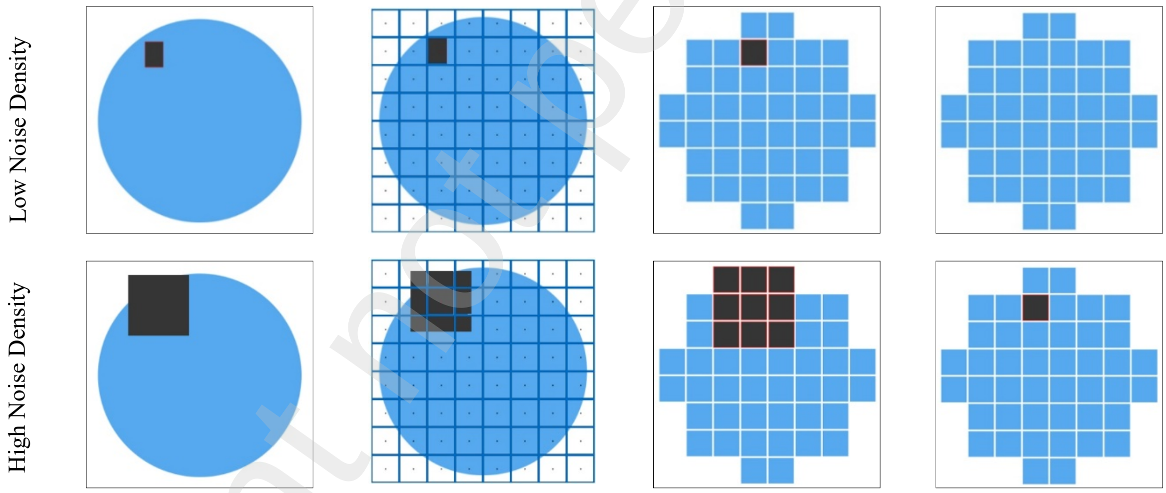
Else

Set the corresponding coordinates of updated noise-map to non-noisy.

End if

This adaptive pixel selection works by eliminating noisy pixels in reduceing the image size and makes the noisy pixels are not participated in the process of X-ray image dimension reduction.

In this study, we incorporated this adaptive resizing function as a layer in the CNN framework to increase its robustness to noisy X-ray images. In the proposed adaptive resizing layer, an original noisy X-ray image (e.g., pixels) is resized to pixels by using the information of the spatial distribution of noise, derived from the corresponding noise-map (discussed in Section 3.1). Using this adaptive resizing layer in our CNN framwork, as shown in Figure 9, we can avoid the transmission of noisy pixels from the original X-ray image to the resized X-ray image. Figure 10 illustrates the pipeline of our adaptive image resizing at the front end of the CNN.



(a) Image with noisy (dark) and regular (blue) pixels

(b) Dividing the image into pixels blocks (small blue boxes) for subsampling

(c) Subsampled image using the central pixel selection method

(d) Subsampled image using the adaptive resizing method

Figure 8: Comparison of image resizing techniques using the central pixel selection and adaptive pixel selection (i.e., adaptive resizing).

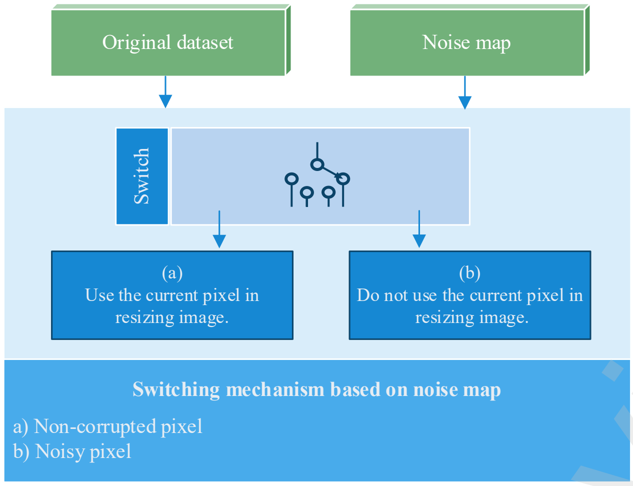


Figure 9: The architecture of the adaptive resizing layer [25].

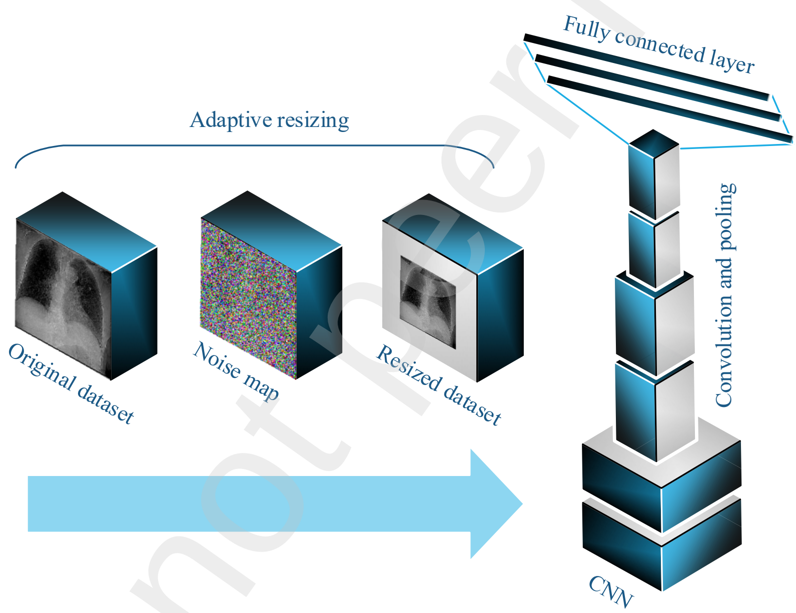


Figure 10: The pipeline of adaptive resizing operation at the front end of the CNN.

3.3. Adaptive Convolution

After adaptive resizing (discussed in Section 3.2), there still might exist noisy pixels in the X-ray images (second row of Figure 8d). Therefore, we design our convolution layer adaptive to make it more robust to image noise. Typically, a new feature map is generated by a convolution layer of a CNN as [27]:

(14)

where is the location coordinate of the k^{th} kernel, is the input image/feature patch, is the learned weight matrix of the k^{th} convolution kernel, and is the bias of the convolution layer. In this paper, we modify the conventional convolution layer of a CNN to make it more robust to noise by incorporating our noise-map as [25]:

$$. \qquad (15)$$

Since noise-map ***s*** is a binary map, incorporating it into the convolution kernel helps not to propagate noisy pixel value forward along the network. We illustrate this operation in Figure 11, where we see that the noisy pixels get shut off (i.e., having a value of 0) during feature calculation. Figure 12 illustrates the architecture of the adaptive convolutional layer for robustness of deep CNN to noisy X-ray images. Eliminating noisy connections avoids inputting impulsive noisy pixels to the next layers.

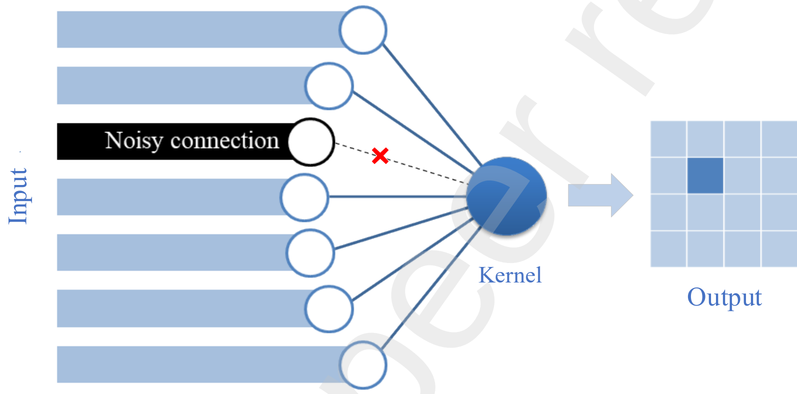


Figure 11: Schematic diagram illustrating the shutting off a noisy pixel during convolution operation in a CNN convolution layer.

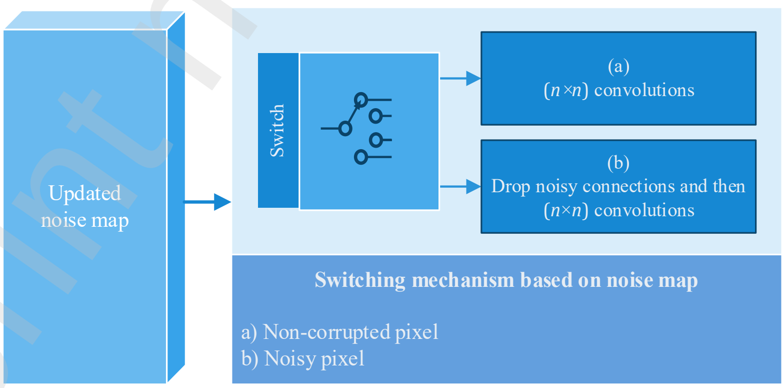


Figure 12: The architecture of the proposed noise-robust adaptive convolution layer.

3.4. Learning-to-Augment Using Noisy Data

Adding some noise to data (e.g., the impulse noise, the Gaussian noise) is a strategy for data augmentation [8]. We have employed a learning-to-augment strategy [8] using noisy X-ray images to generate the new data. The noise density ( ) is the parameter of impulse noise [28-30] and the mean ( ) and variance ( ) are parameters of the Gaussian noise [31, 32]. As shown in Figure 13, the learning to augment using noisy data is composed of a noisy data generator, a controller, an augmenter, and child models. Firstly, the original dataset is partitioned into two folds. Then a noisy data generator adds impulse noise and Gaussian noise to the X-ray images in each fold, separately. The augmenters generate new X-ray images based on the parameters that the Bayesian optimizer has found. Then, each fold is separately fed to the child CNN models. Using the output of child CNNs, the controller increases performance of weak policies and keeps improved policies. The controller employs the Bayesian optimization algorithm to optimize augmentation policies (the parameters of impulse noise and Gaussian noise). Assuming that is the search space and is the loss function of a child CNN, the Bayesian optimization algorithm can be defined as [8]:

(16)

The optimizer algorithm in equation 16 obtains that minimizes for in a bounded domain . The final loss function of the Bayesian optimization algorithm is composed of the individual loss values from child CNNs. The optimization process lasts until the optimized parameters is achieved.

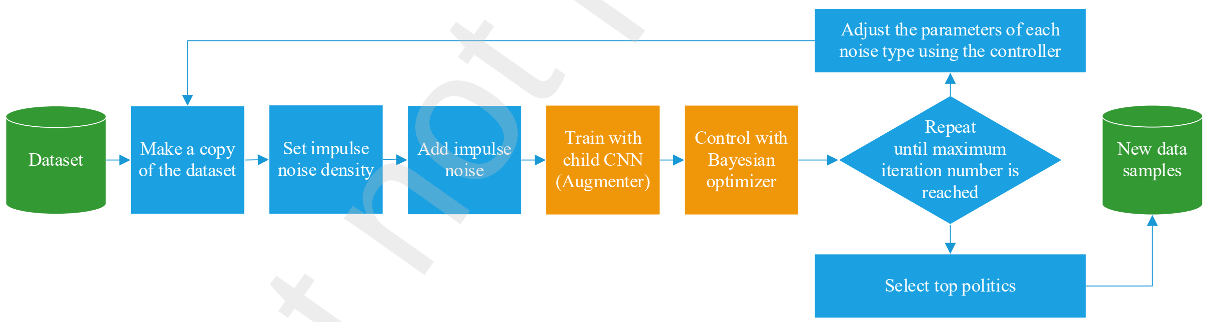


Figure 13: Flowchart of learning-to-augment using noisy data.

4. Implementation Details

In this section, we discuss detials of our extensive experimental setup in detecting COVID-19 in noisy X-ray images. We divided our X-ray dataset into training (70%), validation (10%), and test (20%) sets as shown in Table 2. We also list the types of CNNs (i.e., conventional CNN or noise-robust CNN) used during training, validation, and testing in Table 3. We used learning-to-augment strategy using noisy data only in the training and employed the proposed noise-robust method only in the testing phase. This overall strategy ensures that our model learnes from both noise-free and noisy data. Learning-to-augment strategy using the noisy X-ray images starts by setting the impulse noise density. The noisy data generator creates X-ray images corrupted by the impulse noise. Bayesian optimization algorithm finds the optimum data augmentation policy (i.e., the impulse noise density,), where AlexNet [27] is used as the backbon. The Bayesian optimizer found the optimal value of to be 22%.

Table 2: Data partitioning for training, validation, and testing in this study.

Phase	Data splitting	# of original images	# of augmented images	Total # of images
Train	70%	1,466	1,466	**2,932**
Validation	10%	209	-	**209**
Test	20%	418	-	**418**

Table 3: Types of CNNs (conventional/noise-robust) used in training, validation, and testing.

Phase	Type of CNN	# of noise-free images	# of noisy images	Total # of images
Train	Conventional CNN	1,466	1,466	**2,932**
Validation	Conventional CNN	209	-	**209**
Test	The noise-robust CNN	-	418	**418**

We incorporated our proposed noise-robust modules in the state-of-the-art networks such as SqueezeNet [33], GoogleNet [34], MobileNetv2 [35], ResNet18 [12], ResNet50 [12], ShuffleNet [36], and EfficientNetb0 [37]. As an example, Figure 14 and Table 4 show the architecture and configuration of our proposed model using pre-trained SqueezeNet, respectively, for the classification of X-ray images. We can see in Figure 14 and Table 4 that the SqueezeNet model takes input through the proposed adaptive resizing layer that resizes an X-ray image from pixels to pixels, where new interpolated pixel values are estimated from the noise-free neighboring pixels in the original image. Also, in the first convolutional layer, the convolution kernel incorporates the binary noise-map so that noisy pixel values of the input images do not propagate to the next layer. Similarly, the adaptive resizing layer is incorporated in other deep CNNs (GoogleNet, MobileNetv2, ResNet18, ResNet50, ShuffleNet, and EfficientNetb0) to robustify those to impulse noise as well. Also, the first convolution layer of all the networks were modifiedto incorporate the adaptive convolution to make those robust to noise. It is also woth noting that the size of the original X-ray images in our dataset is larger than the usual input image size of pre-trained deep CNNs. Consequently, after performing adaptive resizing, an input image usually becomes less noisy before it is fed to the adaptve convolution layer.

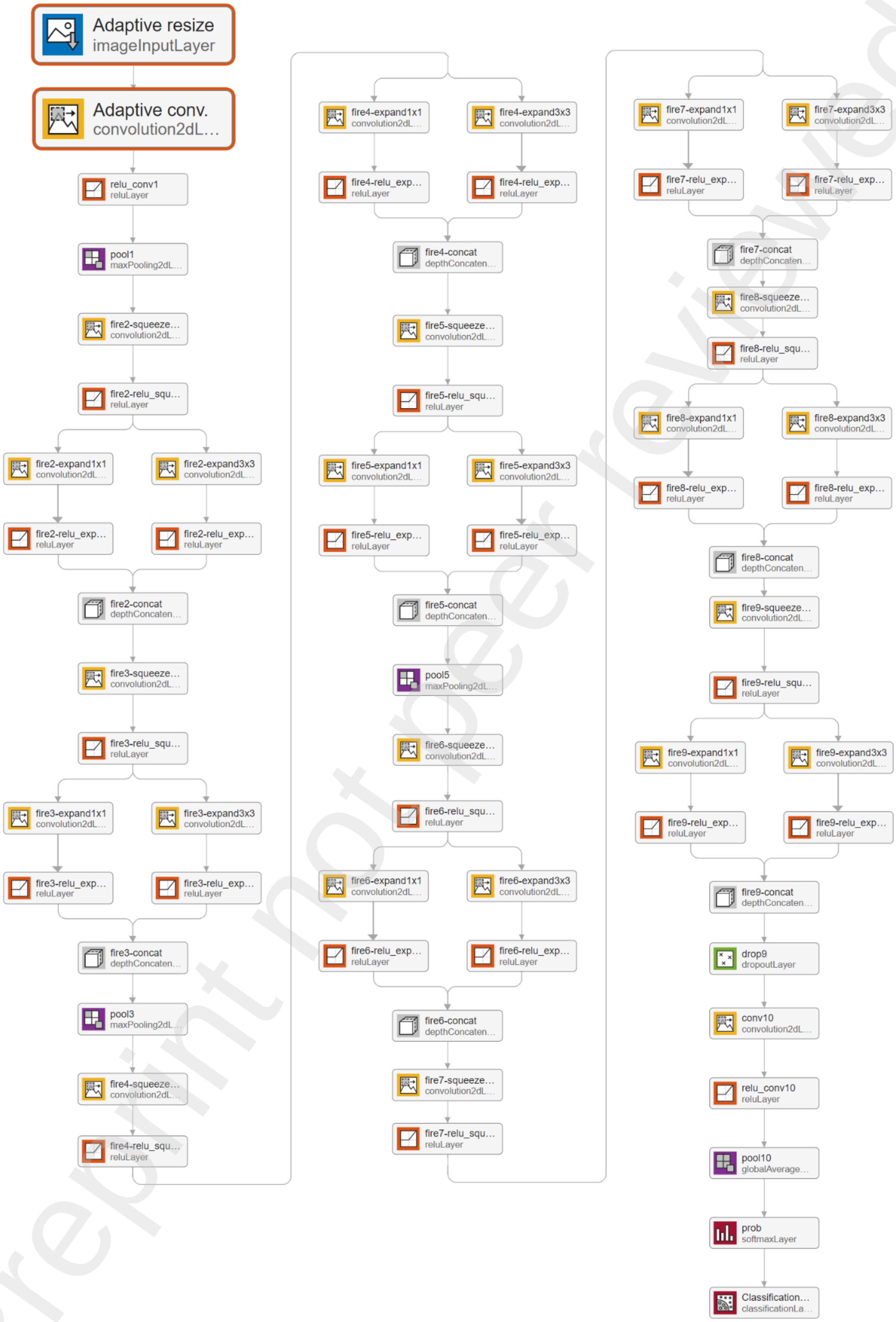


Figure 14: The architecture of the proposed noise-robust SqueezeNet model.

Table 4: The configuration of the proposed noise-robust SqueezeNet model.

#	Type	Descriptions
0	Adaptive Resizing	512×512×3 images
1	Input of Image and Noise-Map	227×227×3 images with 'zerocenter' normalization
2	Adaptive Convolution	64 3×3 convolutions stride [2 2] and padding [0 0 0 0]
3	ReLU	
4	Max Pooling	3×3 max pooling stride [2 2] and padding [0 0 0 0]
5	Convolution	16 1×1 convolutions stride [1 1] and padding [0 0 0 0]
6	ReLU	
7	Convolution	64 1×1 convolutions stride [1 1] and padding [0 0 0 0]
8	ReLU	
9	Convolution	64 3×3 convolutions stride [1 1] and padding [1 1 1 1]
10	ReLU	
11	Concatenation	Depth concatenation of 2 inputs
12	Convolution	16 1×1 convolutions stride [1 1] and padding [0 0 0 0]
13	ReLU	
14	Convolution	64 3×3 convolutions stride [1 1] and padding [1 1 1 1]
15	ReLU	
16	Convolution	64 1×1 convolutions stride [1 1] and padding [0 0 0 0]
17	ReLU	
18	Concatenation	Depth concatenation of 2 inputs
19	Max Pooling	3×3 max pooling with [2 2] and padding [0 1 0 1]
20	Convolution	32 1×1 convolutions stride [1 1] and padding [0 0 0 0]
21	ReLU	
22	Convolution	128 3×3 convolutions stride [1 1] and padding [1 1 1 1]
23	ReLU	
24	Convolution	128 1×1 convolutions stride [1 1] and padding [0 0 0 0]
25	ReLU	
26	Concatenation	Depth concatenation of 2 inputs
27	Convolution	32 1×1 convolutions stride [1 1] and padding [0 0 0 0]
28	ReLU	
29	Convolution	128 3×3 convolutions stride [1 1] and padding [1 1 1 1]
30	ReLU	
31	Convolution	128 1×1 convolutions stride [1 1] and padding [0 0 0 0]
35	Convolution	48 1×1 convolutions stride [1 1] and padding [0 0 0 0]
36	ReLU	
37	Convolution	192 3×3 convolutions stride [1 1] and padding [1 1 1 1]
38	ReLU	
39	Convolution	192 1×1 convolutions stride [1 1] and padding [0 0 0 0]
40	ReLU	
41	Concatenation	Depth concatenation of 2 inputs
42	Convolution	48 1×1 convolutions stride [1 1] and padding [0 0 0 0]
43	ReLU	
44	Convolution	192 1×1 convolutions stride [1 1] and padding [0 0 0 0]
45	ReLU	
46	Convolution	192 3×3 convolutions stride [1 1] and padding [1 1 1 1]
47	ReLU	
48	Concatenation	Depth concatenation of 2 inputs
49	Convolution	64 1×1 convolutions stride [1 1] and padding [0 0 0 0]
50	ReLU	
51	Convolution	256 3×3 convolutions stride [1 1] and padding [1 1 1 1]
52	ReLU	
53	Convolution	256 1×1 convolutions stride [1 1] and padding [0 0 0 0]
54	ReLU	
55	Concatenation	Depth concatenation of 2 inputs
56	Convolution	64 1×1 convolutions stride [1 1] and padding [0 0 0 0]
57	ReLU	
58	Convolution	256 1×1 convolutions stride [1 1] and padding [0 0 0 0]
59	ReLU	
60	Convolution	256 3×3 convolutions stride [1 1] and padding [1 1 1 1]
61	ReLU	
62	Concatenation	Depth concatenation of 2 inputs
63	Dropout	50% dropout
64	Convolution	1000 1×1 convolutions stride [1 1] and padding [0 0 0 0]
65	ReLU	
66	Pooling	Global Average Pooling
67	Softmax	
68	Classification	Output

Table 5: Properties of the pretrained CNN models we used in this study.

Network	Depth	Size	Parameters (Millions)	Input Image Size
SqueezeNet	18	5.2 MB	1.24	227 227
GoogleNet	22	27 MB	7.00	224 224
MobileNetv2	53	13 MB	3.50	224 224
ResNet18	18	44 MB	11.70	224 224
ResNet50	50	96 MB	25.60	224 224
ShuffleNet	50	5.4 MB	1.40	224 224
EfficientNetb0	82	20 MB	5.30	224 224

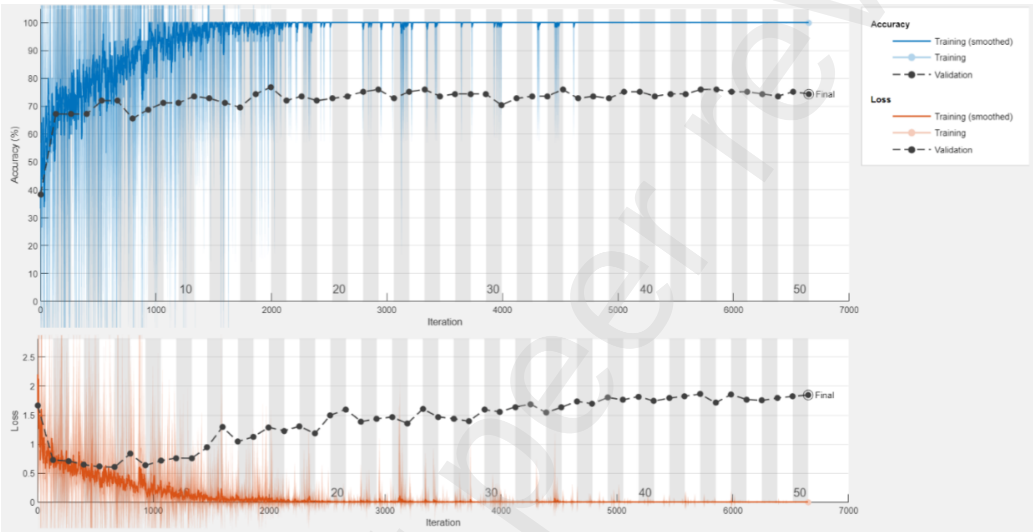


Figure 15: Aaccuracy *vs.* iteration and Loss *vs.* iteration curves for the training and validation of GoogleNet.

We ran our deep learning experiments using the deep learning toolbox of MATLAB 2021a in an Intel(R) Core(TM) i7-7700HQ CPU 2.81 GHz with 32 GB of RAM, and Nvidia GTX 1070 GPU with 8 GB VRAM. We employed stochastic gradient descent (SGD) with a learning rate of 0.001 to finetune the pretrained CNNs. In Table 5, we show a summary of the properties of CNNs we used in our study.

5. Comparison of Time Complexity

The run-time of the methods that improve the quality of noisy X-ray images is of great important [38], especially for the point-of-care machines in the clinical environment. Typically, X-ray images corrupted by the impulse noise are enhanced in two phases [22, 39, 40]. In the first phase, noise-free or noisy pixels are identified. Then in the final phase, enhancement of the quality of X-ray images is done. Following the same workflow, we do noise detection by generating the noise-map of an X-ray image by using a switching technique-based fuzzified degree in the first phase of our proposed method. Afterwards, as a manifestation of the second phase, we design our CNN such that it becomes robust to noise and does not require any preprocessing of an X-ray image in terms of noise reduction.

Recent works [41-44] suggested that the median filtering is one of the fastest method of removing the impulse noise. However, the time complexity of computing the median filter kernel by quick sort algorithm is O( ). In contrast, the proposed model does not require to sort the data, rather it uses the switching technique with time complexity of O( ) to identify noisy pixels. Thus, the comparison of time complexity between the proposed method and the median filtering (i.e., one of the fastest method for removal of impulse noise [41-44]) indicates the superiority of the proposed method.

6. Experimental Results

In this section, we discuss the performance comparison of the proposed approach with respect to the state-of-the-arts on the detection of COVID-19 in noisy X-ray images. The COVID-19 detection accuracy curves during GoogleNet training and validation with the impulse ( ) noise-corrupted X-ray images are illustrated in Figure 15. We compare the classification performance by the proposed method to that of the state-of-the-art methods in three scenarios: (i) training conventional CNNs using data without augmentation, (ii) training conventional CNNs with data augmentated by learning-to-augment strategy, and (iii) training proposed noise-robust CNNs with data augmented by learning-to-augment strategy. Figure 16 illustrates the COVID-19 detection performance for the test X-ray dataset corrupted by impulse noise with of 4%, 6%, 8%, and 10% for all three scenarios. It can be seen that the COVID-19 detection accuracy by the pretrained networks using scenario-iii is the best among all three scenarios.

We also show the COVID-19 detection errors (i.e., CNN classification error) on the impluse noise-corrupted X-ray testset for in three scenarios. We see in Table 4 that the performance by the ResNet50 in scenario-iii is the best among other error performances. It reduced the error in scenario-iii compared to scenario-ii by 2% (i.e., 31% to 29%), and compared to scenario-i by massive 53% (i.e., 82% to 29%) for . Thus, it is clear from Table 4 that our proposed approach using adaptive resizing, adaptive convolution, and learning-to-augment strategy has great efficacy in accurately classfying noisy image data. Finally, we show the line charts of COVID-19 detection accuracy using the impulse noise-corrupted X-ray data with for three scenarios. We see in Figure 17 that the scenario-iii showed the best detection performance among all three scenarios. Thus, it becomes more evident that the proposed method can effetively classify noisy images with higher accuracy.

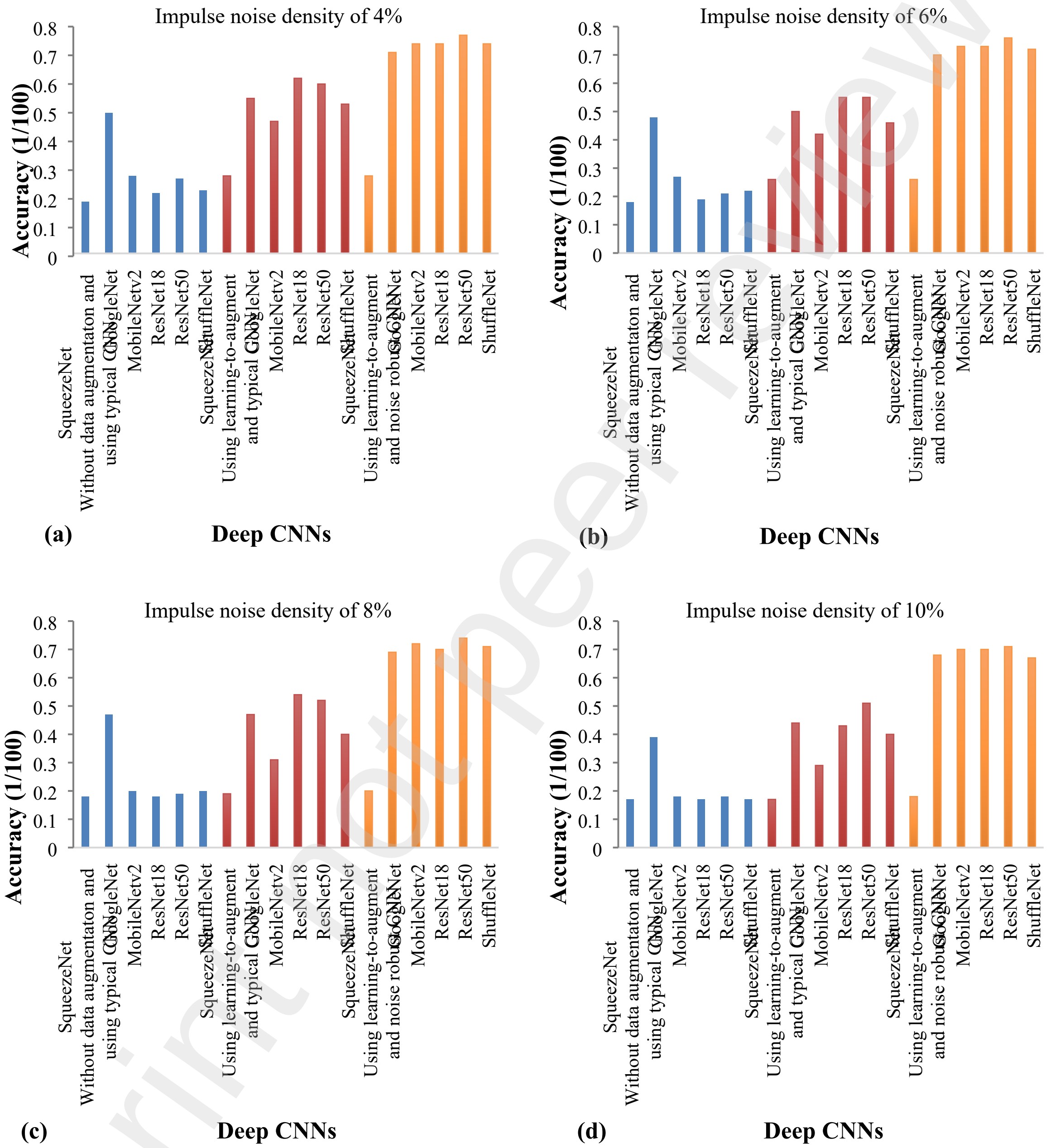


Figure 16: The accuracy of COVID-19 detection by different methods for noisy X-ray images corrupted by the impulse noise with (a) , (b) , (c) , and (d) .

Table 4: COVID-19 detection error (1/100) on X-ray images corrupted by the impulse noise with . Here,

scenarios: (i) training conventional CNNs using data without augmentation, (ii) training conventional CNNs with data augmentated by learning-to-augment strategy, and (iii) training proposed noise-robust CNNs with data augmented by learning-to-augment strategy

Networks	Scenario	Impulse noise density									
		1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
SqueezeNet	i	0.79	0.80	0.80	0.81	0.82	0.82	0.82	0.82	0.83	0.83
	ii	0.56	0.69	0.71	0.72	0.72	0.74	0.75	0.81	0.82	0.83
	iii	0.55	0.70	0.70	0.72	0.72	0.74	0.75	0.80	0.82	0.82
GoogleNet	i	0.48	0.49	0.49	0.50	0.50	0.52	0.53	0.53	0.58	0.61
	ii	0.33	0.39	0.44	0.45	0.48	0.50	0.52	0.53	0.53	0.56
	iii	0.26	0.28	0.28	0.29	0.29	0.30	0.31	0.31	0.32	0.32
MobileNetv2	i	0.59	0.62	0.67	0.72	0.72	0.73	0.77	0.80	0.80	0.82
	ii	0.28	0.34	0.36	0.53	0.58	0.58	0.69	0.69	0.70	0.71
	iii	0.24	0.24	0.26	0.26	0.26	0.27	0.27	0.28	0.28	0.30
ResNet18	i	0.51	0.70	0.75	0.78	0.78	0.81	0.82	0.82	0.83	0.83
	ii	0.26	0.27	0.30	0.38	0.39	0.45	0.45	0.46	0.53	0.57
	iii	0.22	0.25	0.25	0.26	0.26	0.27	0.28	0.30	0.30	0.30
ShuffleNet	i	0.66	0.76	0.77	0.77	0.77	0.78	0.79	0.80	0.80	0.83
	ii	0.35	0.38	0.41	0.47	0.47	0.54	0.54	0.60	0.60	0.60
	iii	0.22	0.25	0.26	0.26	0.28	0.28	0.29	0.29	0.32	0.33
ResNet50	i	0.65	0.66	0.71	0.73	0.78	0.79	0.79	0.81	0.82	0.82
	ii	0.49	0.49	0.48	0.45	0.45	0.42	0.40	0.31	0.28	0.31
	iii	**0.21**	**0.22**	**0.23**	**0.23**	**0.24**	**0.24**	**0.25**	**0.26**	**0.26**	**0.29**

7. Conclusion

In this report, we propose a novel noise-robust deep CNN framework for improving detection of COVID-19 in the impulse noise-corrupted X-ray images. Our proposed framework includes several novel image processing modules. The noise-map layer module can effectively improve detection in a noisy image by making use of switching technique based on fuzzified degree. The adaptive resizing layer module can simultaneously remove noisy pixels while performing interpolation-based image resizing. In addition, the adaptive convolution layer module incorporates noise-map from the first module into the convolution operation that effectively shuts off the remaining noisy pixels in the input image. We further incorporated the learning-to-augment strategy for automatic augmentation of training images, which improved the generalizability of the deep models on X-ray images. We incorporated our novel modules into several pretrained state-of-the-art deep CNNs such as SqueezeNet, GoogleNet, MobileNetv2, ResNet18, ResNet50, ShuffleNet, and EfficientNetb0. Validation of the proposed noise-robust model on clinically acquired X-ray images from COVID-19, non-COVID pneumonia and healthy subjects demonstrated better COVID-19 detection performance on noisy X-ray images compared to the state-of-the-art models. Moreover, the proposed model requires no preprocessing for impulse noise removal, rather noise removal happens on-the-fly because of our novel modules, which speeds up the classification of noisy X-ray. Therefore, our data suggest that the proposed deep CNN framework could be very effective in classification tasks, even on the noisy data, and could improve the generalization of deep CNN. In the near future, we aim to examine the ability of our noise-robust CNN to improve such classification task in the high-density noise-corrupted X-ray images.

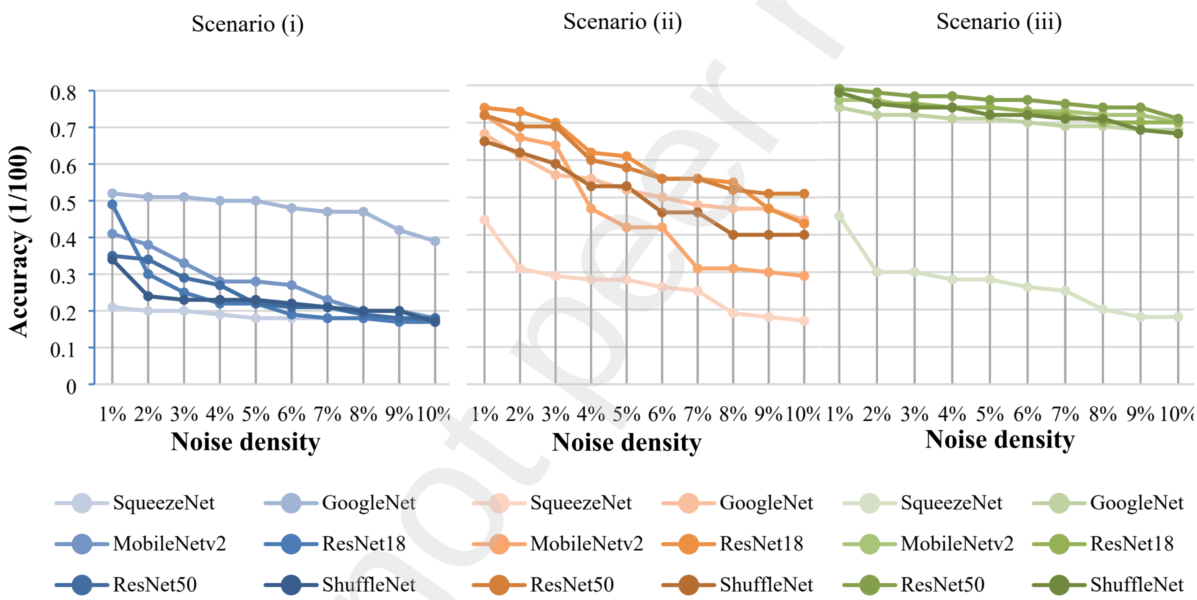


Figure 17: Line chart of COVID-19 detection accuracy using the impulse noise-corrupted X-ray data with .

References

[1] M. K. Hasan *et al.*, "COVID-19 identification from volumetric chest CT scans using a progressively resized 3D-CNN incorporating segmentation, augmentation, and class-rebalancing," *Informatics in Medicine Unlocked,* p. 100709, 2021/08/28/ 2021, doi: https://doi.org/10.1016/j.imu.2021.100709.

[2] G. Jia, H.-K. Lam, and Y. Xu, "Classification of COVID-19 chest X-Ray and CT images using a type of dynamic CNN modification method," *Computers in Biology and Medicine,* vol. 134, p. 104425, 2021/07/01/ 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104425.

[3] A. M. Ismael and A. Şengür, "Deep learning approaches for COVID-19 detection based on chest X-ray images," *Expert Systems with Applications,* vol. 164, p. 114054, 2021/02/01/ 2021, doi: https://doi.org/10.1016/j.eswa.2020.114054.

[4] A. Jahanbakhshi, Y. Abbaspour-Gilandeh, K. Heidarbeigi, and M. Momeny, "A novel method based on machine vision system and deep learning to detect fraud in turmeric powder," *Computers in Biology and Medicine,* vol. 136, p. 104728, 2021/09/01/ 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104728.

[5] A. Jahanbakhshi, Y. Abbaspour-Gilandeh, K. Heidarbeigi, and M. Momeny, "Detection of fraud in ginger powder using an automatic sorting system based on image processing technique and deep learning," *Computers in Biology and Medicine,* vol. 136, p. 104764, 2021/09/01/ 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104764.

[6] A. Jahanbakhshi, M. Momeny, M. Mahmoudi, and Y.-D. Zhang, "Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks," *Scientia Horticulturae,* vol. 263, p. 109133, 2020/03/15/ 2020, doi: https://doi.org/10.1016/j.scienta.2019.109133.

[7] M. Momeny, A. Jahanbakhshi, K. Jafarnezhad, and Y.-D. Zhang, "Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach," *Postharvest Biology and Technology,* vol. 166, p. 111204, 2020/08/01/ 2020, doi: https://doi.org/10.1016/j.postharvbio.2020.111204.

[8] M. Momeny *et al.*, "Learning-to-augment strategy using noisy and denoised data: Improving generalizability of deep CNN for the detection of COVID-19 in X-ray images," *Computers in Biology and Medicine,* vol. 136, p. 104704, 2021/09/01/ 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104704.

[9] S. Thakur and A. Kumar, "X-ray and CT-scan-based automated detection and classification of covid-19 using convolutional neural networks (CNN)," *Biomedical Signal Processing and Control,* vol. 69, p. 102920, 2021/08/01/ 2021, doi: https://doi.org/10.1016/j.bspc.2021.102920.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation. arXiv 2015," *arXiv preprint arXiv:1505.04597,* 2019.

[11] H. Munusamy, J. M. Karthikeyan, G. Shriram, S. Thanga Revathi, and S. Aravindkumar, "FractalCovNet architecture for COVID-19 Chest X-ray image Classification and CT-scan image Segmentation," *Biocybernetics and Biomedical Engineering,* vol. 41, no. 3, pp. 1025-1038, 2021/07/01/ 2021, doi: https://doi.org/10.1016/j.bbe.2021.06.011.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[13] 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.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[15] S. Pathan, P. C. Siddalingaswamy, and T. Ali, "Automated Detection of Covid-19 from Chest X-ray scans using an optimized CNN architecture," *Applied Soft Computing,* vol. 104, p. 107238, 2021/06/01/ 2021, doi: https://doi.org/10.1016/j.asoc.2021.107238.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software,* vol. 69, pp. 46-61, 2014/03/01/ 2014, doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.

[17] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software,* vol. 95, pp. 51-67, 2016/05/01/ 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[18] R. Mostafiz, M. S. Uddin, N.-A. Alam, M. Mahfuz Reza, and M. M. Rahman, "Covid-19 detection in chest X-ray through random forest classifier using a hybridization of deep CNN and DWT optimized features," *Journal of King Saud University - Computer and Information Sciences,* 2020/12/31/ 2020, doi: https://doi.org/10.1016/j.jksuci.2020.12.010.

[19] C.-T. Lu, M.-Y. Chen, J.-H. Shen, L.-L. Wang, and C.-C. Hsu, "Removal of salt-and-pepper noise for X-ray bio-images using pixel-variation gain factors," *Computers & Electrical*

Engineering, vol. 71, pp. 862-876, 2018/10/01/ 2018, doi: https://doi.org/10.1016/j.compeleceng.2017.08.012.
[20] B. Garg, P. S. Rana, and V. S. Rathor, "Significance driven inverse distance weighted filter to restore impulsive noise corrupted X-ray image," *Journal of Ambient Intelligence and Humanized Computing,* 2021/03/03 2021, doi: 10.1007/s12652-021-02962-y.
[21] M. Azhar, H. Dawood, H. Dawood, G. I. Choudhary, A. K. Bashir, and S. H. Chauhdary, "Detail-preserving switching algorithm for the removal of random-valued impulse noise," *Journal of Ambient Intelligence and Humanized Computing,* vol. 10, no. 10, pp. 3925-3945, 2019/10/01 2019, doi: 10.1007/s12652-018-1153-0.
[22] P. Satti, N. Sharma, and B. Garg, "Min-Max Average Pooling Based Filter for Impulse Noise Removal," *IEEE Signal Processing Letters,* vol. 27, pp. 1475-1479, 2020, doi: 10.1109/LSP.2020.3016868.
[23] S. Arora, M. Hanmandlu, and G. Gupta, "Filtering impulse noise in medical images using information sets," *Pattern Recognition Letters,* vol. 139, pp. 1-9, 2020/11/01/ 2020, doi: https://doi.org/10.1016/j.patrec.2018.06.002.
[24] R. Pugalenthi, A. S. Oliver, and M. Anuradha, "Impulse noise reduction using hybrid neuro-fuzzy filter with improved firefly algorithm from X-ray bio-images," *International Journal of Imaging Systems and Technology,* https://doi.org/10.1002/ima.22453 vol. 30, no. 4, pp. 1119-1131, 2020/12/01 2020, doi: https://doi.org/10.1002/ima.22453.
[25] M. Momeny, A. M. Latif, M. Agha Sarram, R. Sheikhpour, and Y. D. Zhang, "A noise robust convolutional neural network for image classification," *Results in Engineering,* vol. 10, p. 100225, 2021/06/01/ 2021, doi: https://doi.org/10.1016/j.rineng.2021.100225.
[26] M. Nadeem, A. Hussain, A. Munir, M. Habib, and M. T. Naseem, "Removal of random valued impulse noise from grayscale images using quadrant based spatially adaptive fuzzy filter," *Signal Processing,* vol. 169, p. 107403, 2020/04/01/ 2020, doi: https://doi.org/10.1016/j.sigpro.2019.107403.
[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM,* vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
[28] N. Zarmehi and F. Marvasti, "Removal of sparse noise from sparse signals," *Signal Processing,* vol. 158, pp. 91-99, 2019/05/01/ 2019, doi: https://doi.org/10.1016/j.sigpro.2019.01.002.
[29] L. Wang, D. Xiao, W. S. Hou, X. Y. Wu, and L. Chen, "Weighted Schatten p-norm minimization for impulse noise removal with TV regularization and its application to medical images," *Biomedical Signal Processing and Control,* vol. 66, p. 102123, 2021/04/01/ 2021, doi: https://doi.org/10.1016/j.bspc.2020.102123.
[30] M. Nooshyar and M. Momeny, "Removal of high density impulse noise using a novel decision based adaptive weighted and trimmed median filter," in *2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, 2013: IEEE, pp. 387-391.
[31] Y. Zhu, W. Shen, F. Cheng, C. Jin, and G. Cao, "Removal of high density Gaussian noise in compressed sensing MRI reconstruction through modified total variation image denoising method," *Heliyon,* vol. 6, no. 3, p. e03680, 2020/03/01/ 2020, doi: https://doi.org/10.1016/j.heliyon.2020.e03680.
[32] N. Karimi and M. R. Taban, "A convex variational method for super resolution of SAR image with speckle noise," *Signal Processing: Image Communication,* vol. 90, p. 116061, 2021/01/01/ 2021, doi: https://doi.org/10.1016/j.image.2020.116061.
[33] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv preprint arXiv:1602.07360,* 2016.
[34] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.
[36] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848-6856.
[37] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019: PMLR, pp. 6105-6114.
[38] C. Lien, C. Huang, P. Chen, and Y. Lin, "An Efficient Denoising Architecture for Removal of Impulse Noise in Images," *IEEE Transactions on Computers,* vol. 62, no. 4, pp. 631-643, 2013, doi: 10.1109/TC.2011.256.
[39] K. Panetta, L. Bao, and S. Agaian, "A New Unified Impulse Noise Removal Algorithm Using a New Reference Sequence-to-Sequence Similarity Detector," *IEEE Access,* vol. 6, pp. 37225-37236, 2018, doi: 10.1109/ACCESS.2018.2850518.
[40] C. Lin, Y. Li, S. Feng, and M. Huang, "A Two-Stage Algorithm for the Detection and Removal of Random-Valued Impulse Noise Based on Local Similarity," *IEEE Access,* vol. 8, pp. 222001-222012, 2020, doi: 10.1109/ACCESS.2020.3040760.
[41] A. Shah *et al.*, "Comparative analysis of median filter and its variants for removal of impulse noise from gray scale images," *Journal of King Saud University - Computer and Information Sciences,* 2020/03/27/ 2020, doi: https://doi.org/10.1016/j.jksuci.2020.03.007.
[42] C. J. J. Sheela and G. Suganthi, "An efficient denoising of impulse noise from MRI using adaptive switching modified decision based unsymmetric trimmed median filter," *Biomedical Signal Processing and Control,* vol. 55, p. 101657, 2020/01/01/ 2020, doi: https://doi.org/10.1016/j.bspc.2019.101657.
[43] O. Appiah, M. Asante, and J. B. Hayfron-Acquah, "Improved approximated median filter algorithm for real-time computer vision applications," *Journal of King Saud University - Computer and Information Sciences,* 2020/04/15/ 2020, doi: https://doi.org/10.1016/j.jksuci.2020.04.005.
[44] R. Huang, K. Qing, D. Yang, and K.-S. Hong, "Real-time motion artifact removal using a dual-stage median filter," *Biomedical Signal Processing and Control,* vol. 72, p. 103301, 2022/02/01/ 2022, doi: https://doi.org/10.1016/j.bspc.2021.103301.