

A Novel Normalization Algorithm to Facilitate Pre-Assessment of Covid-19 Disease from Chest X-ray Image and its FPGA implementation

Sertaç Yaman (✉ sertacyaman@hakkari.edu.tr)

Hakkari University

Barış Karakaya

Firat University

yavuz erol

Firat University

Research Article

Keywords: Normalization, FPGA, Covid-19 disease, Deep Learning, Image Processing.

Posted Date: January 11th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1092597/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

COVID-19 is still a fatal disease, which has threatened all people by affecting the human lungs. Chest X-Ray or computed tomography (CT) imaging is commonly used to make a fast and reliable medical investigation to detect the COVID-19 virus from these medical images is remarkably challenging because it is a full-time job and prone to human errors. In this paper, a new normalization algorithm that consists of Mean-Variance-Softmax-Rescale (MVSR) processes respectively is proposed to provide facilitation pre-assessment and diagnosis Covid-19 disease. In order to show the effect of MVSR normalization technique on image processing, the algorithm is applied to chest X-ray images. Therefore, the normalized X-ray images with MVSR are used to recognize via one of the neural network models as known Convolutional Neural Networks (CNNs). At the implementation stage, the MVSR algorithm is executed on MATLAB environment, then it is implemented on FPGA platform. All the arithmetic operations of the MVSR normalization are coded in VHDL with the help of fixed-point fractional number representation format. The experimental platform consists of Zynq-7000 Development Board and VGA monitor to display the both original X-ray and MVSR normalized image. The CNN model is constructed and executed using Anaconda Navigator interface with python language. Based on the results of this study, infections of Covid-19 disease can be easily diagnosed for MVSR normalized image. The proposed MVSR normalization makes the accuracy of CNN model increase from 83.01%, to 96.16% for binary class of chest X-ray images.

1. Introduction

Normalization with mean and variance is a noise/error compensation method that is used in many applications such as signal, image and speech processing, artificial intelligence, big data, optimization algorithms and so on [1–3]. One of the compensation techniques is Batch normalization (BN) method to develop performance and stability of neural networks by rescaling values of hidden layer units and also to create more complicated deep learning models. One of the critical problems is the covariate shift in the learning process for deep learning models. To solve this problem, BN is proposed in 2015 [4]. Besides, Cepstral mean and variance normalization (CMVN) [5] is used to eliminate noise inside voice for speech processing applications. CMVN is a critical method for recognizing voice instructions to minimize the data lost for messages during normalization.

Noise in the image is a general problem for image processing applications [6]. It is also hot topic in computer vision and image processing. Many techniques based on image denoising model have been developed to overcome this problem [7–8]. In most studies, researchers firstly add noise (especially Gaussian noise) with different noise ratios to the original image, then they design deep CNNs to enhance image denoising performance (speed, quality and cost) and efficiency [9]. Some of these techniques consists of CNN with BN [10], image restoration by applying ResNet [11], several traditional linear and non-linear image filter techniques [12].

Deep Normalized Convolutional Neural Network (DNCNN) is developed for unbalanced classifications of neural networks and constructed based on ReLU and weight normalization strategy. In [13], In DNCNN, ReLU is employed as an activation function to overcome the gradient vanishing problem and enhance the optimization by using the parameters (weight, bias) normalization strategy. Weight Normalization (WN) is used to accelerate the convergence of stochastic gradient descent optimization in a neural network. In [14], WN is applied to show benefits of this technique in four different NN models. WN method is also used for image smoke detection in [15]. The result of this study shows that false alarm ratio is achieved under 0.60% and smoke detection accuracy is around 96.37%.

CNN models have been constructed in many different ways [16] for image processing (such as image segmentation, classification and denoising) but still, there are many challenge during image processing that requires amounts of parameters (image size, weights, and bias, etc.) and lots of convolution operations in the training and inference process [17–19]. This has inspired many researchers to focus their attention on more energy-efficient hardware platforms [20, 21]. Field Programmable Gate Array (FPGA) is a vital side for implementation of CNN models due to its hardware adaptability and parallel process capability. On the other hand, General Processing Units (GPUs) supply more resources in terms of memory and computational units. But, during the validation process, GPUs have a long execution time because of sequential logic design [22, 23]. FPGA has parallel architectures on its logic configuration and this feature makes it better choice for hardware designers during validation process of CNN configurations. At the implementation stage of CNN models on any hardware, normalization process ensures fewer number of hardware resources and reduced power consumption during the training and validation processes [22–25].

In this paper, a new normalization algorithm that consists of MVSR (named as acronym of Mean-Variance-Softmax-Rescale) processes respectively is proposed. In order to show effect of MVSR on image processing, the algorithm is applied to chest X-ray images to provide facilitation pre-assessment and diagnosis Covid-19 disease. Therefore, the normalized X-ray images with MVSR are used to recognize via one of the neural network models as known Convolutional Neural Networks (CNNs). At the implementation stage, the MVSR algorithm is executed on MATLAB, then it is implemented on FPGA platform. The experimental platform consists of Zynq-7000 Development Board and VGA monitor to display the both original X-ray and MVSR normalized image. The CNN model is constructed and executed using Anaconda Navigator interface with python language.

The rest of the paper is organized as follows. The architecture and mathematic of the MVSR normalization technique and its necessity are detailed in Section 2. In Section 3, the MATLAB pre-study is accomplished for observing the effect of proposed normalization algorithm on chest X-ray images. Section 4 consists of FPGA implementation of MVSR normalization and the results for chest X-ray images on VGA screen. Then, in Section 5, the architecture of CNN model is constructed and the implementation of image classification (diagnosis of Covid-19 disease) for original and normalized chest X-ray images are executed on Anaconda Navigator interface with python language to prove the benefits of proposed MVSR normalization algorithm. Section 6 concludes the paper.

2. Mvsr Normalization Algorithm

In general, the standard deviation (the square root of the variance) is used to make a relation between each data and measure how to spread out or distribute around the mean of a data set [26]. The MVSR normalization is built on these features of mathematical operations and four-part processes,

- Mean
- Variance
- Softmax
- Rescaling of the data, respectively.

According to R. Duncan Luce's probability theory, it is also known as Luce's choice axiom, probability of one of the data is dependent on another in the same data set. The softmax function, based on Luce's choice axiom, is usually employed in the last activation function of multiclass Artificial Neural Network (ANN) models to make probability distribution of the output classes [27]. After calculation of the normalized intensity of the input, the dataset can have the negative and the positive fractional numbers. In order to conserve the effect of negative data and also nonlinearity, we used the softmax function. In the rescaling step, all data is transformed to 8-bit unsigned integer intervals change from 0-1 to 0-255. The MVSR normalization technique consists of following mathematical processes where μ_{MVSR} , σ_{MVSR}^2 , \hat{x}_i , $S(x)_i$ and r_i stand for mean of x_i , variance, the normalized intensity of the input, softmax and rescaled values, respectively.

$$\mu_{MVSR} = \frac{1}{M} \sum_{i=1}^M x_i \quad (1)$$

$$\sigma_{MVSR}^2 = \frac{1}{M} \sum_{i=1}^M (x_i - \mu_{MVSR})^2 \quad (2)$$

$$\hat{x}_i = \frac{(x_i - \mu_{MVSR})}{\sqrt{\sigma_{MVSR}^2 + \delta}} \quad (3)$$

$$S(x)_i = \frac{e^{\hat{x}_i}}{\sum_j e^{\hat{x}_j}} \quad (4)$$

$$r_i = \frac{S(x)_i}{\max(S(x)_i)} * 255 \quad (5)$$

In the normalized intensity value of the input (\hat{x}_i), a small positive number (δ) is added to the variance value to avoid the divisor is to be zero. In the equations above, x_i is the input data intensity and r_i is the result of MVSR normalization.

The MVSR normalization architecture is given in Figure 1. The mathematical operations are executed in order to represented flow. Input chest X-ray image is inputted the algorithm and the normalized version of input is applied to neural network model.

3. Pre-study Of Mvsr Normalization Algorithm On Matlab

In the pre-study, to observe the effect of MVSR normalization for Covid-19 disease diagnosis from chest X-ray images, the algorithm is executed on MATLAB. Chest X-ray images of Covid-19 infected and No-Finding are represented after and before MVSR normalization in the figures 2. the source of the X-ray images dataset is publicly available at Hospital Israelita Albert Einstein [28]. According to the expert view, Infectious Diseases Doctor, after MVSR normalization the infection on chest X-ray images can be more clearly observed. At this stage, the starting point of this study is based on this view and this normalization process is thought to be used in image classification applications. The accuracy of Covid-19 diagnose can be increased by using MVSR normalization before training process for the neural network models. The rest of the paper is based on this projection.

The MATLAB-based MVSR normalization used for the pre-phase was taken as a reference to design an FPGA-based MVSR normalization. In order to show benefit of the MVSR normalization, more images are normalized and presented in Figure 3.

4. Mvsr Normalization On Fpga

Calculations of standard deviation, mean, and variance are required to apply MVSR operation. To obtain these operations, we need to define real numbers as a fixed point number type to embedded systems or any hardware. Multiplication, division, addition, subtraction, and especially square root calculations are essential mathematical arithmetic to calculate these operations in FPGA. MVSR normalization is implemented by using a static random access memory (SRAM) on re-programmable FPGA. All the data is read from SRAM and defined arrays. Systolic arrays are used for these operations to perform parallel integrations such as convolution, correlation, linear algebra or additional data tasks [29]. All normalization processes are constructed with pure VHDL at Vivado 2018.1 version.

4.1. Division of the fixed-point fractional binary numbers

Division is one of the important operation to calculate standard deviation, mean, and variance. Fixed-point fractional number format is a way to represent fractional numbers. The fixed point division process is slightly different from the standard binary division operation. It is needed to scale the fractional part of the division after finding the remainder [30].

Algorithm 1 Pseudo Code for Division Operation

If Dividend < 0

Dividend = Not Dividend +1

else if Divisor < 0

```

Divisor = Not Divisor +1

else if Divisor or Dividend = 0

Quotient=0

Remainder=0

else

Remainder = Dividend rem Divisor

Quotient= Dividend / Divisor

nRemainder=100* Remainder/ Divisor

mRemainder = LUT_operation (nRemainder)

End

Output= Quotient& mRemainder

```

The *LUT_operation* consists of lookup table to provide programmable hardware functionality for division operation and to validate remainder values by matching against a list of valid items in the array. The values stored in LUT are in the unsigned fractional fixed point binary numbers. For example, when the *nRemainder* is equal to "00010110" (stands for 0.22 in decimal), *mRemainder* is "00111000" in 8-bit format.

4.2. Square root of the fixed-point fractional binary numbers

This section represents calculating the fixed-point square root. Square root of the fixed point binary numbers are essential to have normalized input data with standard deviation from variance operation in the MVSR operation. After making the groups of data by 2^n ($n=1, 2... m, m=data\ size/2$), it is gradually compared with the multiplied expected data. If the group of the data is smaller than expected, the result is equal to 1, vice versa 0.

Algorithm 2. Square root operation on VHDL

```

datau:=unsigned(datain);

expectdata (1 downto 0):="01";

for i in 0 to length loop

```

```

if (datau(size downto size-1-2*i) < expectdata (2*i+1 downto 0)) then
dtu(length-i):='0';
else
dtu(length-i):='1';
end if;
expectdata (2*i+3 downto 0) :=(dtu(length downto length-i)&'1')*(dtu(length downto length-i)&'1');
dt:=std_logic_vector(dtu);
end loop;
return dt;

```

The FPGA implementation platform where Xilinx Zynq-7000 Development Board is used to implement the MVSR normalization algorithm for input images and VGA monitor to display the both original X-ray and MVSR normalized image.

Results of FPGA implementation of MVSR normalization algorithm is given in Figure 4. As seen in the figure, the algorithm is implemented on FPGA board and the results are shown on VGA monitor successfully. The main result of FPGA implementation stage is that the X-ray image can be normalized on FPGA and the expert views support our opinions on the starting point of this study.

Table 1
Resource utilization of FPGA Chip

Resource	Utilization	Available	Utilization %
LUT	193	53200	0.36
FF	136	106400	0.13
BRAM	30	140	21.43
IO	18	200	9

In the Table 1, information about resource utilization of FPGA implementation stage of study is shown. This result shows that the MVSR normalization algorithm is performed on FPGA chip with the fewer number of resource.

5. Proposed Convolutional Neural Network Architecture

The architecture of the CNN model is constructed with seven convolutional layers and five max-pooling layers and this model can be seen in Table.2. This model is executed for original and MVSR normalized chest X-ray images on Anaconda Navigator interface with python language to prove the benefits of the proposed MVSR normalization algorithm.

Table 2
CNN architecture

Layers	Layer Configurations			
	S	N	W	H
Input	-	-	150	150
Conv1	3x3	32	150	150
Conv2	3x3	32	150	150
Conv3	3x3	64	150	150
Max-Pool	2x2	-	75	75
Conv4	3x3	128	75	75
Max-Pool	2x2	-	37	37
Conv5	3x3	64	37	37
Max-Pool	2x2	-	18	18
Conv6	3x3	128	18	18
Max-Pool	2x2	-	9	9
Conv7	3x3	128	9	9
Max-Pool	2x2	-	4	4
FC1	2048			
FC2	512			
Output	1			

In the Table.2, *S* is kernel size, *N* is number of kernel, *W* is image size (weight) and *H* is image size (height). Model *learning rate* is settled to 0.01 with *decay steps* as 300 and *decay rate* 0.95. Adam optimization [30] is used with binary cross entropy loss function, *epochs* is 48 and *batch size* is 48. In order to eliminate the over fitting and have better classification, batch-normalization (*momentum*=0.07) is employed before *max-pool* layers and *dropout* is used with 25% between *FC1* and *FC2*. *Relu*, one of the activation function, is used after convolutional layers except for the output layer, where the sigmoid activation function is chosen instead. The Convolutional Neural Network model is trained and validated

with the Covid-19 dataset [32] that includes 200 training images. The same CNN model is used for two data sets, which are original and MVSR normalized chest X-ray images. Validation of these data sets have been performed for 365 X-ray images of 150x150 pixels labeled as *Covid-19* and *No-Finding*. The accuracy of the CNN model reached 96.16% from 80.01% after MVSR normalization.

There are four-type of prediction approaches, four distinct combinations of predicted and actual values, to evaluate results of classification model via confusion matrix [33], [41]. These are True Positives (T_P), True Negatives (T_N), False Positives (F_P) and False Negatives (F_N).

- T_P : These are the people who are diagnosed with Covid-19 infection by clinical tests and the CNN model validates them as Covid-19 infected.
- T_N : These are the people who are not diagnosed with Covid-19 disease by clinical tests and the CNN model validates them as not infected.
- F_P : These are the people who are not diagnosed with Covid-19 disease by clinical tests but the CNN model classified them as Covid-19 infected.
- F_N : These are the people who are diagnosed with Covid-19 disease by clinical tests but the CNN model classified them as not infected.

F1-score is the harmonic mean of the *Precision* and *Recall*. Therefore, f1-score gives a combined idea of these two measurements. F1-score is maximum when the *Precision* is equal to *Recall*. The precision determines whether the model is reliable or not. Prediction according to the model, the recall is given how many true positive cases are accurately predicted.

Precision; It counts how precise the executed model operates by examining TP from the predicted ones

$$P = \frac{T_P}{(T_P + F_P)}$$

6

Recall (Sensitivity); It is the percentage of recognition of positive samples

$$R = \frac{T_P}{(T_P + F_N)}$$

7

F-1 Score; It is a function to find a balance between sensitivity and precision.

$$F1 = \frac{2}{\left(\frac{1}{R} + \frac{1}{P}\right)}$$

Table 3
Obtained F1-scores

	F1-score	F1-score MVSR
Covid-19	0.51	0.92
No-Finding	0.85	0.96

The results of the f1-scores given in Table.3, performed with original chest X-ray and MVSR normalized images, the normalized images perform the better F1-Score (accuracy).

Figure 5. shows the confusion matrix of models, which are trained with original and MVSR normalized images. This matrix demonstrate how the CNN models classify and misclassify Covid-19 disease on chest X-ray images. Furthermore, the precision, recall and f1-score can be easily computable by using confusion matrix.

Table 4
Normalization techniques used for the detection of COVID-19 patients in literature

Preprocessing	DL Technique	Number of Cases	Performance		Study
			Accuracy (%)		
			Without Norm.	With Norm.	
Segmentation, Rescaling, Multi-view Fusion	ResNet50	368 COVID-19 Patients, 127 Patients with Other Pneumonia	NA	76	[34]
Different Features, Early Fusion, Late Fusion, Different Resampling Algorithms	Inception-V3	90 COVID-19, 10 MERS, 11 SARS, 10 Varicella, 12 Streptococcus, 11 Pneumocystis Samples	83	88	[35]
GAN, Rescaling, Cropping	DECAPS Architecture	349 COVID-19 Images, 397 Non-COVID-19 Images	84,3	87,6	[36]
Rescaling	COVIDX-Net (VGG19, DenseNet201)	25 COVID-19, 25 Normal Cases	83	90	[37]
Lung x-ray aperture in images region removed, Histogram equalization and bilateral filter	VGG16, Transfer learning with ImageNet	415 covid-19 infected, 5179 other to viruses having and 2880 normal	88	94,5	[38]
CXI image processing techniques	GoogleNet M-inception, Transfer learning with ImageNet	1065 images, Covid-19 and Normal CXI	82,5	93	[39]
Histogram equalization, negative conversion, gamma correction, Contrast enhancement techniques	ResNet18, ResNet50, ResNet101, InceptionV3, DenseNet201, ChexNet	18479 images, 3616 Covid-19	93,22	Av. 96,29	[40]

Preprocessing	DL Technique	Number of Cases	Performance		Study
			Accuracy (%)		
			Without Norm.	With Norm.	
MVSR normalization	CNN	240 covid-19, 120 normal, 200 test	83,01	96,16	This study

In order to compare the MVSR normalization with the other normalization techniques, which are used to detect Covid-19 virus on chest X-ray images. The results of the classification by using traditional machine learning models are shown in Table 4.

6. Conclusion

COVID-19 is a pandemic disease, and it can threaten the health of all people in the world. It directly attacks the lung cells, and if not diagnosed early, can cause damage, including mortality. In this paper, we propose a novel method, the MVSR normalization architecture, to facilitate pre-assessment of Covid-19 disease from chest X-ray images. Furthermore, effectiveness of this new normalization technique is proved by using different evaluation methods, which include expert view, MATLAB, Anaconda Navigator interface and FPGA embedded system. MVSR normalization considerably improves the accuracy of the CNN model and according to expert view, the difference of Covid-19 infected and uninfected X-ray images can be seen more clearly at the pre-assessment stage. Finally, it is understood from the study, MVSR normalization can be applied to complex images like chest X-ray images for recognizing the difference between infected and uninfected patients with Covid-19. As a result of this paper, infections of Covid-19 disease can be efficiently diagnosed for MVSR normalized image. The proposed MVSR normalization technique makes the accuracy of the CNN model increase from 83.01%, to 96.16% on chest X-ray images

Declarations

Acknowledgements

The authors would like to thank Prof. Dr. Kutbeddin DEMİRDAĞ for his valuable expert view, discussions and suggestions

References

1. Scalco, Elisa, et al. "T2w-MRI signal normalization affects radiomics features reproducibility." *Medical physics*4 (2020): 1680-1691.
2. Song, Guojie, et al. "Deep Convolutional Neural Network based Medical Concept Normalization." *IEEE Transactions on Big Data*(2020).

3. Xu, Liming, et al. "Low-dose chest X-ray image super-resolution using generative adversarial nets with spectral normalization." *Biomedical Signal Processing and Control*55 (2020): 101600.
4. Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *International conference on machine learning*. PMLR, 2015.
5. Joshi, Vikas, N. Vishnu Prasad, and Srinivasan Umesh. "Modified mean and variance normalization: transforming to utterance-specific estimates." *Circuits, Systems, and Signal Processing*5 (2016): 1593-1609.
6. Hong, Jung Hee, et al. "Incremental image noise reduction in coronary CT angiography using a deep learning-based technique with iterative reconstruction." *Korean journal of radiology*10 (2020): 1165.
7. Islam, Mohammad Tariqul, et al. "Mixed Gaussian-impulse noise reduction from images using convolutional neural network." *Signal Processing: Image Communication*68 (2018): 26-41.
8. Noor, Alam, et al. "Median filters combined with denoising convolutional neural network for Gaussian and impulse noises." *Multimedia Tools and Applications*25 (2020): 18553-18568.
9. Zhang, Kai, et al. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising." *IEEE transactions on image processing*7 (2017): 3142-3155.
10. Tian, Chunwei, Yong Xu, and Wangmeng Zuo. "Image denoising using deep CNN with batch renormalization." *Neural Networks*121 (2020): 461-473.
11. Ren, Dongwei, et al. "Progressive image deraining networks: A better and simpler baseline." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
12. Motwani, Mukesh C., et al. "Survey of image denoising techniques." *Proceedings of GSPX*. Vol. 27. 2004.
13. Jia, Feng, et al. "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization." *Mechanical Systems and Signal Processing*110 (2018): 349-367.
14. Salimans, Tim, and Diederik P. Kingma. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks." *arXiv preprint arXiv:1602.07868*(2016).
15. Yin, Zhijian, et al. "A deep normalization and convolutional neural network for image smoke detection." *Ieee Access*5 (2017): 18429-18438.
16. Supraja, P., et al. "3D convolution neural network-based person identification using gait cycles." *Evolving Systems*4 (2021): 1045-1056.
17. Köpüklü, Okan, et al. "Real-time hand gesture detection and classification using convolutional neural networks." *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 2019.
18. Nayak, Deepak Ranjan, Ratnakar Dash, and Banshidhar Majhi. "Automated diagnosis of multi-class brain abnormalities using MRI images: A deep convolutional neural network based method." *Pattern Recognition Letters*138 (2020): 385-391.

19. Yamashita, Rikiya, et al. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging*4 (2018): 611-629.
20. Duan, Yunzhi, et al. "Energy-Efficient Architecture for FPGA-based Deep Convolutional Neural Networks with Binary Weights." *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*. IEEE, 2018.
21. Pang, Wei, Chenglu Wu, and Shengli Lu. "An Energy-Efficient Implementation of Group Pruned CNNs on FPGA." *IEEE Access*8 (2020): 217033-217044.
22. Wang, Yu, Qian Chen, and Baeomin Zhang. "Image enhancement based on equal area dualistic sub-image histogram equalization method." *IEEE Transactions on Consumer Electronics*1 (1999): 68-75.
23. Zhijie, Yang, et al. "Bactran: a hardware batch normalization implementation for CNN training engine." *IEEE Embedded Systems Letters*(2020).
24. Nurvitadhi, Eriko, et al. "Accelerating binarized neural networks: Comparison of FPGA, CPU, GPU, and ASIC." *2016 International Conference on Field-Programmable Technology (FPT)*. IEEE, 2016.
25. Baptista, Darío, Leonel Sousa, and Fernando Morgado-Dias. "Raising the Abstraction Level of a Deep Learning Design on FPGAs." *IEEE Access*8 (2020): 205148-205161.
26. Singh, Dalwinder, and Birmohan Singh. "Investigating the impact of data normalization on classification performance." *Applied Soft Computing*97 (2020): 105524.
27. Blavatsky, Pavlo R. "Dual choice axiom and probabilistic choice." *Journal of Risk and Uncertainty*1 (2020): 25-41.
28. Schwab, Patrick, et al. "Clinical predictive models for COVID-19: systematic study." *Journal of medical Internet research*10 (2020): e21439.
29. Wei, Xuechao, et al. "Automated systolic array architecture synthesis for high throughput CNN inference on FPGAs." *Proceedings of the 54th Annual Design Automation Conference 2017*. 2017.
30. Yaman Sertaç, Barış Karakaya, and Yavuz Erol. "Real Time Edge Detection via IP-Core based Sobel Filter on FPGA." *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*. Vol. 1. IEEE, 2019.
31. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*(2014).
32. "Covid-19: covid-19 x-ray images", GitHub, available online, <https://github.com/search?q=covid-19+x-ray+images>.
33. Zeng, Guoping. "On the confusion matrix in credit scoring and its analytical properties." *Communications in Statistics-Theory and Methods*9 (2020): 2080-2093.
34. Wu, Xiangjun, et al. "Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: A multicentre study." *European Journal of Radiology* 128 (2020): 109041.
35. Pereira, Rodolfo M., et al. "COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios." *Computer Methods and Programs in Biomedicine* 194 (2020): 105532.

36. Mobiny, Aryan, et al. "Radiologist-level covid-19 detection using ct scans with detail-oriented capsule networks." *arXiv preprint arXiv:2004.07407*(2020).
37. Hemdan, Ezz El-Din, Marwa A. Shouman, and Mohamed Esmail Karar. "Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images." *arXiv preprint arXiv:2003.11055* (2020).
38. Heidari, Morteza, et al. "Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms." *International journal of medical informatics* 144 (2020): 104284.
39. Wang, Shuai, et al. "A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)." *European radiology* (2021): 1-9.
40. Rahman, Tawsifur, et al. "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images." *Computers in biology and medicine* 132 (2021): 104319.
41. Shoebibi, Afshin, et al. "Automated detection and forecasting of covid-19 using deep learning techniques: A review." *arXiv preprint arXiv:2007.10785* (2020).

Figures

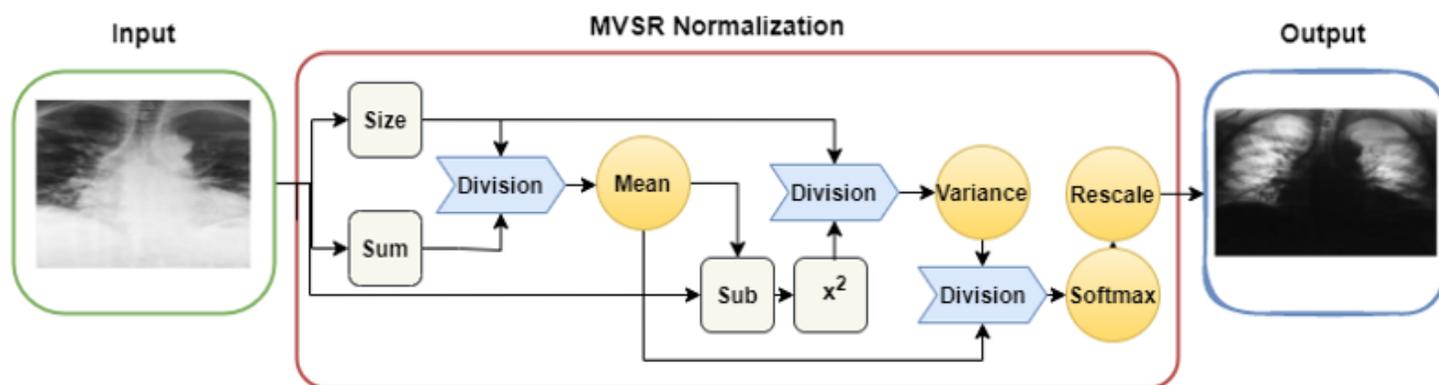
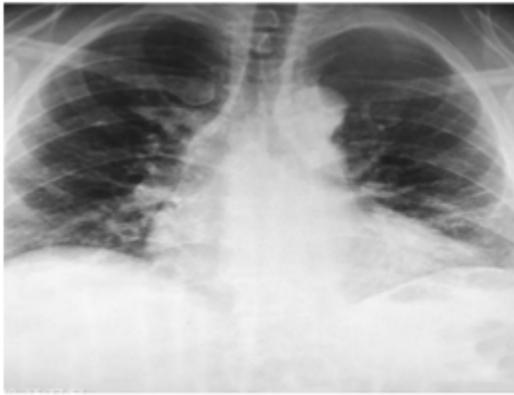
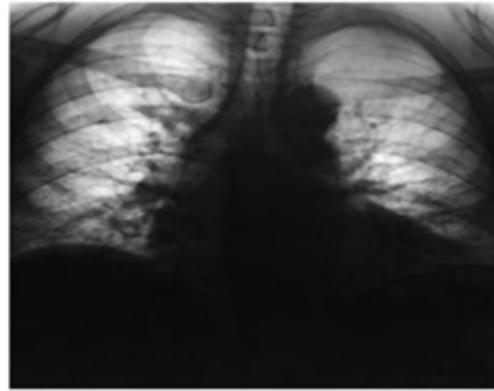


Figure 1

MVSR architecture



a. Covid-19 X-ray



b. Covid-19 MVSR



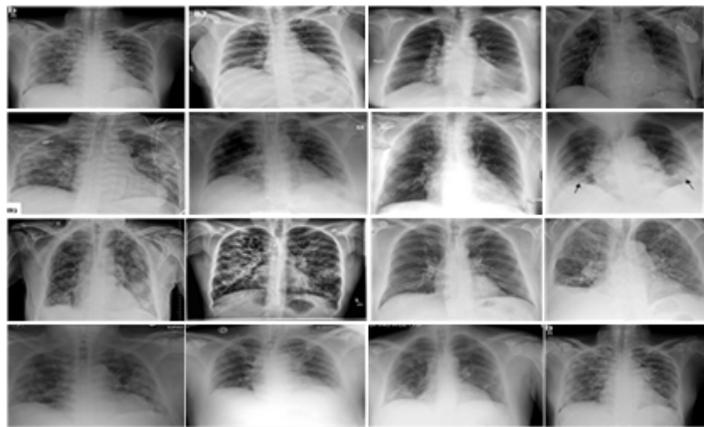
c. No-Finding X-ray



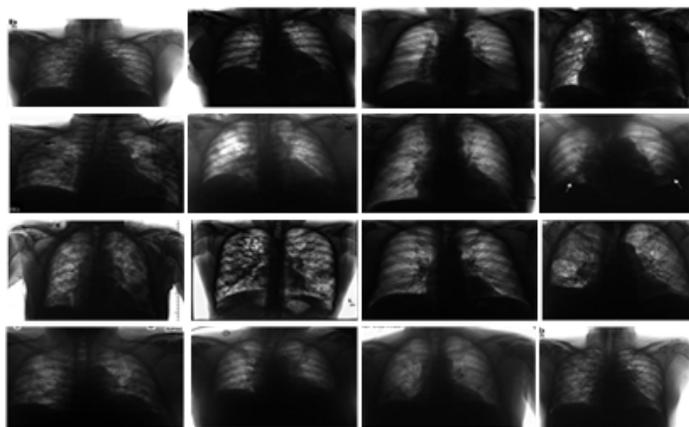
d. No-Finding MVSR

Figure 2

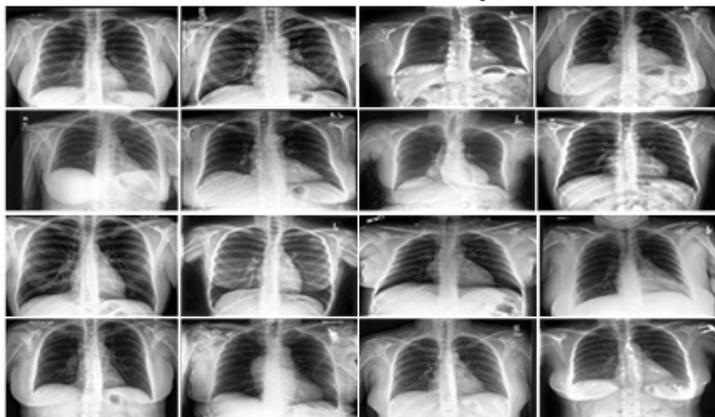
(a), (c) are original chest X-ray images while (b), (d) are the normalized versions of (a), (c)



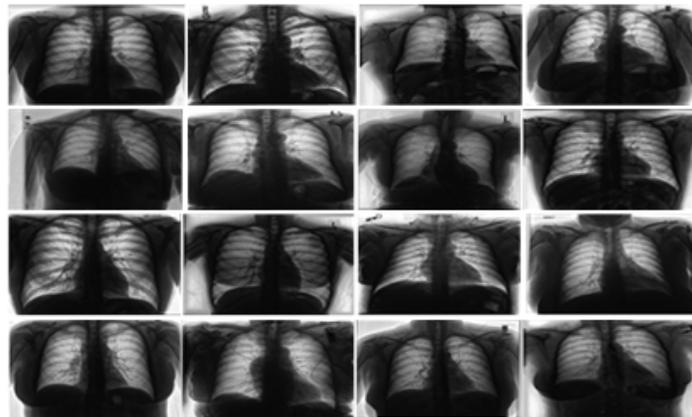
a. Covid-19 X-ray



b. Covid-19 MVSR



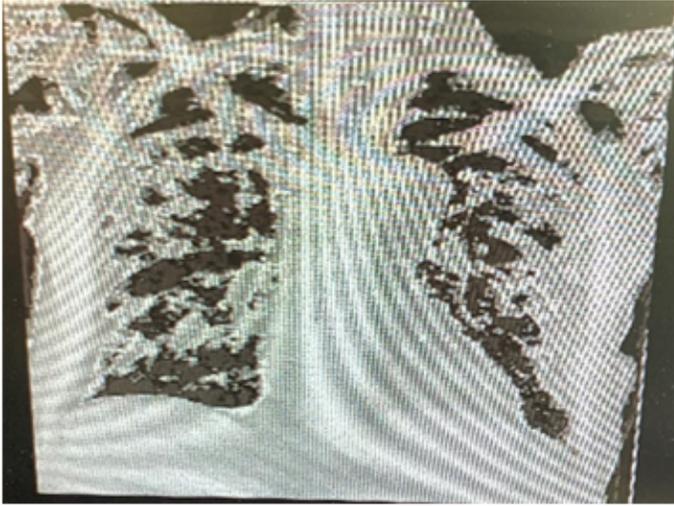
c.No-Finding X-ray



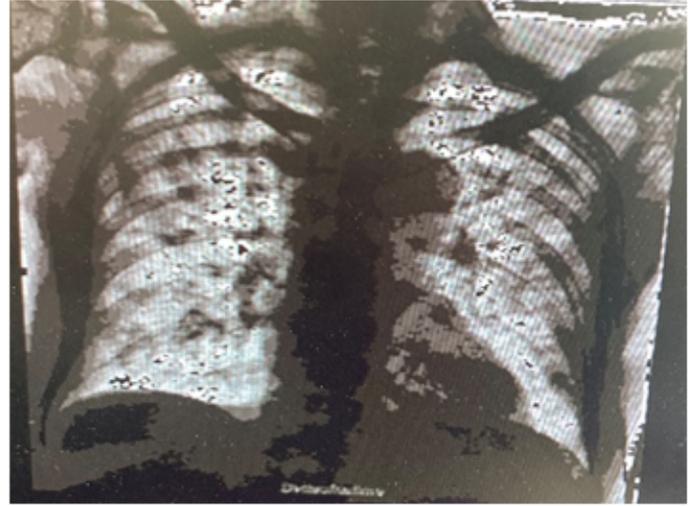
d. No-Finding MVSR

Figure 3

(a), (c) are original chest X-ray images while (b), (d) are the normalized versions of (a), (c)



Covid-19 X-ray



Covid-19 MVSR



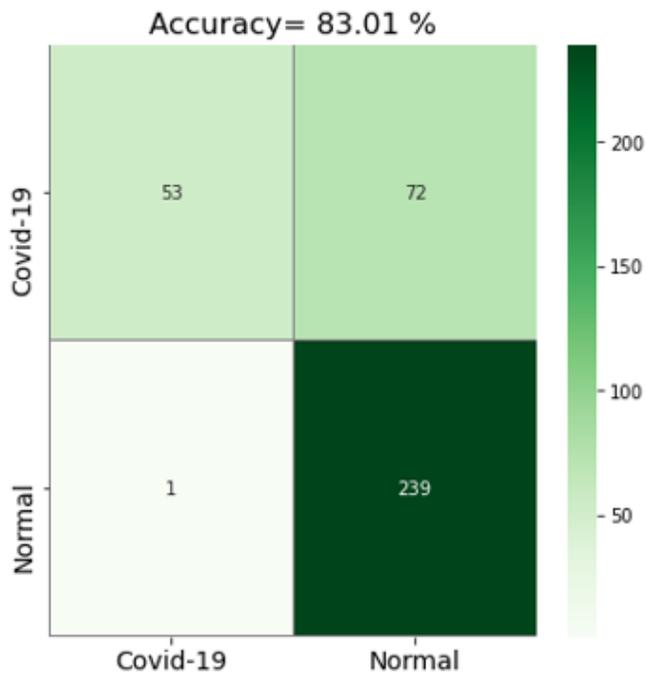
No-Finding X-ray



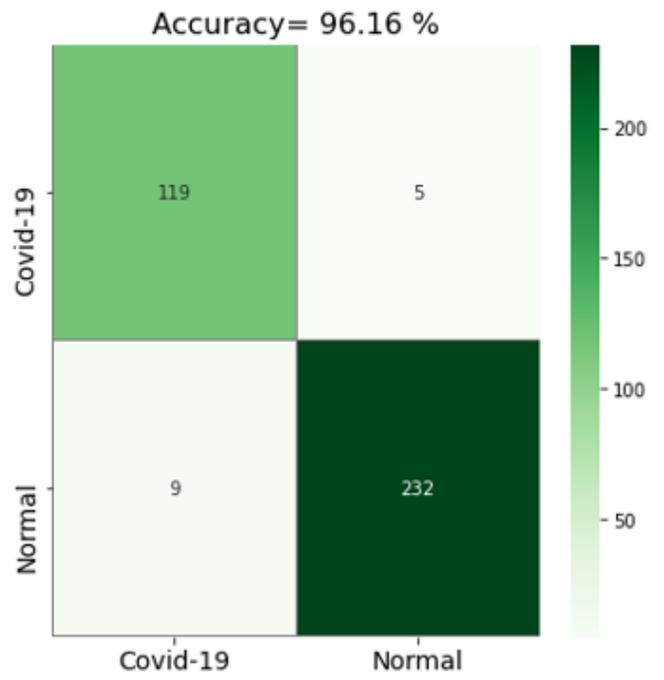
No-Finding MVSR

Figure 4

FPGA implementation of MVSR normalization and the results for chest X-ray images on VGA screen



Without Normalization



After MVSR Normalization

Figure 5

Confusion Matrix