NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2024; 13(3), 905-916

Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

FPGA implementation of deep learning model utilizing different normalization algorithms for COVID-19 diagnosis

COVID-19 teşhisi için farklı normalizasyon algoritmaları kullanılan derin öğrenme modelinin FPGA gerçeklemesi

Merve Zirekgür[1], Barış Karakaya[2,*]

[1] Malatya Turgut Özal University, Faculty of Engineering and Natural Sciences, Department of Electrical-Electronics Engineering, Malatya Türkiye
[2] Fırat University, Faculty of Engineering, Department of Electrical-Electronics Engineering, Elazığ Türkiye

Abstract

Normalization is utilized to remove outliers from the dataset and address network bias. In this research, Mean-Variance-Softmax-Rescale (MVSR) and Min-Max normalizations are employed in various combinations for the diagnosis of COVID-19 using a Convolutional Neural Network (CNN)-based Deep Learning (DL) model, aimed at enhancing network accuracy. To accomplish this, the CNN model is developed within the Google Colab environment and trained using a publicly available dataset consisting of chest X-ray images related to COVID-19. The dataset is normalized using different combinations of the MVSR and Min-Max normalization algorithms to compare model accuracy. Each normalized dataset is used for model training, and subsequently, each trained model has been saved as a *.h5* file and loaded into the Kria KV260 Vision AI Starter Kit FPGA for the testing phase. The most accurate results are obtained when MVSR and Min-Max normalizations are applied simultaneously. This high-performing scenario is re-evaluated with COVID-19 and normal X-ray images on FPGA configuration. Experimentally, the highest accuracy is achieved in real-time with the MVSR+Min-Max scenario, reaching 93%. The model's precision, recall, and F1-Score values are determined as 0.91, 0.96, and 0.93, respectively.

Keywords: Artificial intelligence, Deep learning, Image processing, Normalization

Öz

Normalizasyon, veri setindeki aykırı değerleri ortadan kaldırmak ve ağ yanlılığını gidermek için kullanılmaktadır. Bu çalışmada, COVID-19 hastalığının teşhisi için kullanılan Konvolüsyonel Sinir Ağları (CNN) tabanlı Derin Öğrenme (DL) modeli ile farklı kombinasyonlarda Mean-Variance-Softmax-Rescale (MVSR) ve Min-Max normalizasyonları kullanılarak ağın doğruluğunun artırılması amaçlanmıştır. Bu amaçla, CNN modeli Google Colab ortamında oluşturulmuş ve COVID-19 için göğüs X-ray görüntülerini içeren açık bir veri seti ile eğitilmiştir. Veri seti, model doğruluğunu karşılaştırmak için MVSR ve Min-Max normalizasyon algoritmalarının farklı kombinasyonlarıyla normalize edilmiştir. Her eğitilmiş model, bir .h5 dosyası olarak kaydedilmiş ve ardından test aşaması için Kria KV260 Vision AI Starter Kit FPGA kartına yüklenmiştir. En yüksek doğruluk sonuçları, MVSR ve Min-Max normalizasyonlarının birlikte uygulandığı senaryo ile elde edilmiştir. En iyi performansı veren senaryo, COVID-19 ve normal X-ray görüntüleri ile FPGA yapılandırmasında tekrar test edilmiştir. En yüksek doğruluk, MVSR+Min-Max senaryosuyla deneysel olarak gerçekleştirilmiş ve %93 olarak elde edilmiştir. Modelin kesinliği, duyarlılığı ve F1-Skor değerleri sırasıyla 0.91, 0.96 ve 0.93 olarak belirlenmiştir.

Anahtar kelimeler: Yapay zekâ, Derin öğrenme, Görüntü işleme, Normalizasyon

Introduction

Human intelligence is powered by experiences, observations, and acquired knowledge, whereas Artificial Intelligence (AI) is fueled by the data that is fed as input to the created model. The accuracy of the network is highly dependent on the selection of the method and the input parameters. Errors in selecting the data can lead to problems such as overfitting or underfitting of the created model. Normalization is a technique that involves scaling different data sources to the same range, which is often used to speed up data processing and suppress outliers. Training the network without data normalization can lead to issues, significantly complicating the training process and reducing the learning speed [1]. Normalization is a crucial pre-processing step that ensures each input comes from a standard distribution 1. This means that the range of pixel values in one image is the same as the range in another image, which helps to standardize the data and achieve a standard normal distribution 1. Normalization can aid in the training of neural networks by ensuring that diverse features share a comparable scale. This, in turn, contributes to the stabilization of the gradient descent step, enabling the utilization of larger learning rates or expediting the convergence of models with a given learning rate [2].

Numerous normalization techniques are frequently employed, which encompass Min-Max Normalization, Layer Normalization, Z-Score Normalization, and Batch Normalization (BN).

* Sorumlu yazar / Corresponding author, e-posta / e-mail: bkarakaya@firat.edu.tr (B. Karakaya)
Geliş / Recieved: 29.01.2024 Kabul / Accepted: 18.05.2024 Yayımlanma / Published: 15.07.2024
doi: 10.28948/ngumuh.1427827

Min-Max Normalization, one of the most commonly used normalization techniques, normalizes each input based on the minimum and maximum values of the dataset, while Z-Score Normalization normalizes each value in a dataset to have a mean of 0 and a standard deviation of 1. BN is performed within the hidden layers of a neural network and normalizes the data by grouping it into mini-batches and calculating the mini-batch mean and variance. Layer Normalization, on the other hand, directly estimates the normalization statistics from the inputs collected by neurons in a hidden layer and is generally effective in Recurrent Neural Network (RNN) applications, reducing training time and producing good results.

Rothe et al. [3] used normalization method to determine invariants. Many invariants are known to be obtained by normalization or other methods. The authors showed that the normalization method is much more general and allows many invariant sets to be derived from the second list.

Loffe et al. [4] tackled the issue of internal covariate shift by incorporating the normalization of layer inputs into the model architecture. They implemented normalization for every training mini-batch, effectively addressing the problem.

Kiros et al. [5] proposed Layer Normalization (LN) and showed that Layer Normalization can significantly reduce the training time compared to previously published techniques.

He et al. [6] proposed Group Normalization (GN) as a simple alternative to BN. GN divides channels into groups and calculates the mean and variance for each group for normalization.

Wu et al. [7] developed a Deep Learning (DL) based method that helps to identify COVID-19 disease quickly and accurately with X-ray images by training a multi-view fusion model. The multi-view DL fusion model achieved an Area Under the Receiver Operating Characteristic Curve (AUC) of 73.2% and 70% accuracy, 73% sensitivity, and 61.5% specificity values. In the test set, 81.9%, 76%, 81.1%, and 61.5% AUC, accuracy, sensitivity, and specificity were obtained respectively.

Pereira et al. [8] introduced a classification framework that incorporates both multi-class and hierarchical classification for the detection of the COVID-19 virus through chest X-ray images. They recommended the integration of resampling algorithms in the framework to address the inherent data imbalance in this domain. The proposed methodology was evaluated on the RYDLS-20 dataset, achieving a macro-average F1-Score of 65% in the multi-class approach and an 89% F1-Score specifically for identifying COVID-19 in the hierarchical classification scenario.

Rahman et al. [9] used five different image enhancement techniques to investigate the effect of image enhancement techniques on COVID-19 detection: Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), Image Complement, Gamma Correction, and Balance Contrast Enhancement Technique (BCET). The study resulted in accuracy, precision, sensitivity, F1-Score, and specificity of 95.11%, 94.55%, 94.56%, 94.53%, and 95% respectively.

Yaman et al. [10] introduced a novel normalization algorithm, encompassing mean, variance, softmax, and rescaling (MVSR) processes, aimed at enhancing the initial assessment and diagnosis of COVID-19. They applied this MVSR normalization method to datasets comprising chest X-ray and Sars-Cov-2 computed tomography images to demonstrate its impact. The dataset consists of 240 COVID-19, 120 normal, and 200 test images. The application of the MVSR normalization technique resulted in a notable improvement in the classification accuracy of a CNN model for binary class chest X-ray images, elevating it from 83.01% to 96.16%.

In this study, MVSR and Min-Max normalizations are employed in various combinations for the diagnosis of COVID-19 using a Convolutional Neural Network (CNN)-based DL model, aimed at enhancing network accuracy. To accomplish this, the CNN model is developed within the Google Colab environment and trained using a publicly available dataset consisting of chest X-ray images related to COVID-19. The dataset includes chest X-ray images of COVID-19 positive cases, Normal and Viral Pneumonia images. The dataset was prepared in collaboration with researchers from Doha, Qatar, and Dhaka University in Bangladesh, and doctors in Pakistan and Malaysia [11, 12].

The dataset is normalized using different combinations of the MVSR and Min-Max normalization algorithms to compare model accuracy. Each normalized dataset is utilized for model training, and subsequently, each trained model is saved as a *.h5* file for uploading into the Kria KV260 Vision AI Starter Kit FPGA during the testing phase, and the evaluation results are recorded for comparison purposes. The high-performing scenario is re-evaluated with COVID-19 and normal X-ray images on FPGA configuration. During re-evaluation implementations, Min-Max normalization is performed within the Colab environment, while the MVSR normalization algorithm is executed using the Kria KV260 Vision AI Starter FPGA Kit.

Herewith this introduction, Section 2 presents the definitions and information about the experimental platforms, the dataset, the normalization algorithms, and the CNN model. The proposed methodology and result analysis are given in Section 3. Consequently, Section 4 provides the discussion and conclusion of the study.

Material and methods

In image classification, pre-processing is commonly applied to increase accuracy and enable the network to generate optimum gradients. Although the normalization technique that will provide the optimum result depending on the dataset varies in image classification applications, most of these applications are pre-processed with Min-Max Normalization. In this study, MVSR Normalization, which has been proposed as a new method, and Min-Max Normalization algorithms, which are frequently applied in the literature, were tested for the diagnosis of COVID-19 disease [10] by using chest radiography images. Initially, a CNN model is created and trained sequentially using the same dataset with four different applications. The model is trained with the dataset in order regarding the usage of normalization algorithms, a) without normalization b) Min-Max Normalization c) MVSR Normalization, and d) MVSR+Min-Max Normalization. The dataset containing normal and COVID-19 chest X-ray images are divided into three parts: 600 for testing, 4676 for training, and 1985 for validation. The training and validation data are applied to the CNN model created with Python language in the Google Colab environment and evaluated with performance metrics and saved as a *.h5* file. Then, the *.h5* files of four scenarios are loaded to the XILINX Kria KV260 Vision AI Starter Kit and evaluated again with the test dataset to observe performance metrics. These operations are repeated for each of the selected normalization techniques, and the results are recorded. Finally, the model is trained with the normalization algorithm scenario that shows the highest success rate. The trained model is tested by COVID-19 and normal X-ray images. An overview of the study is shown in Figure 1.

XILINX Kria KV260 Vision AI starter kit

With the development of semiconductor technology and circuit elements, the use of integrated circuits has become more widespread and this technology has become more accessible. Xilinx, one of the world's largest 18 partnerships that produce integrated circuits that can be adjusted after production, also known as FPGA, is the manufacturer of devices in the FPGA type that have many applications in communication, consumption, defense, automotive, and other fields. Xilinx has many products for designing and producing programmable logic circuits for various fields. One of these products is the Kria KV260 Vision AI Starter Kit, which can be seen in Figure 2. As the name suggests, this kit enables the implementation of AI-based applications and is designed to develop advanced visual applications without requiring complex hardware design knowledge.

Kria KV260 Vision AI Starter Kit is designed to provide a platform for evaluating target applications for smart cities and artificial vision, security cameras, retail analytics, and other industrial applications. In addition to this, neural networks can be applied on cloud and edge platforms using the kit's AI development platform. With a low-cost development kit, smart image applications can be put into operation more quickly in the preferred design environment of developers [13].

The Kria KV260 Vision AI Starter Kit enables developers, who may not possess extensive knowledge of FPGA technology, to effortlessly create AI applications using a widely-adopted programming language such as Python. Similar to operating systems with extensive communities like Ubuntu, the Kria KV260 Vision AI Starter Kit affords users the capability to develop AI applications in an environment similar to a computer setup. For instance, the development environment utilized in computers, such as Visual Studio, is also available in this kit, and by installing Jupyter Notebook within it, one can conveniently prepare AI applications. Despite its numerous advantages, the KV260 Vision AI Starter Kit has fewer pins compared to other Kria and FPGA board models, which may limit flexibility and expandability for users working on larger-scale projects.

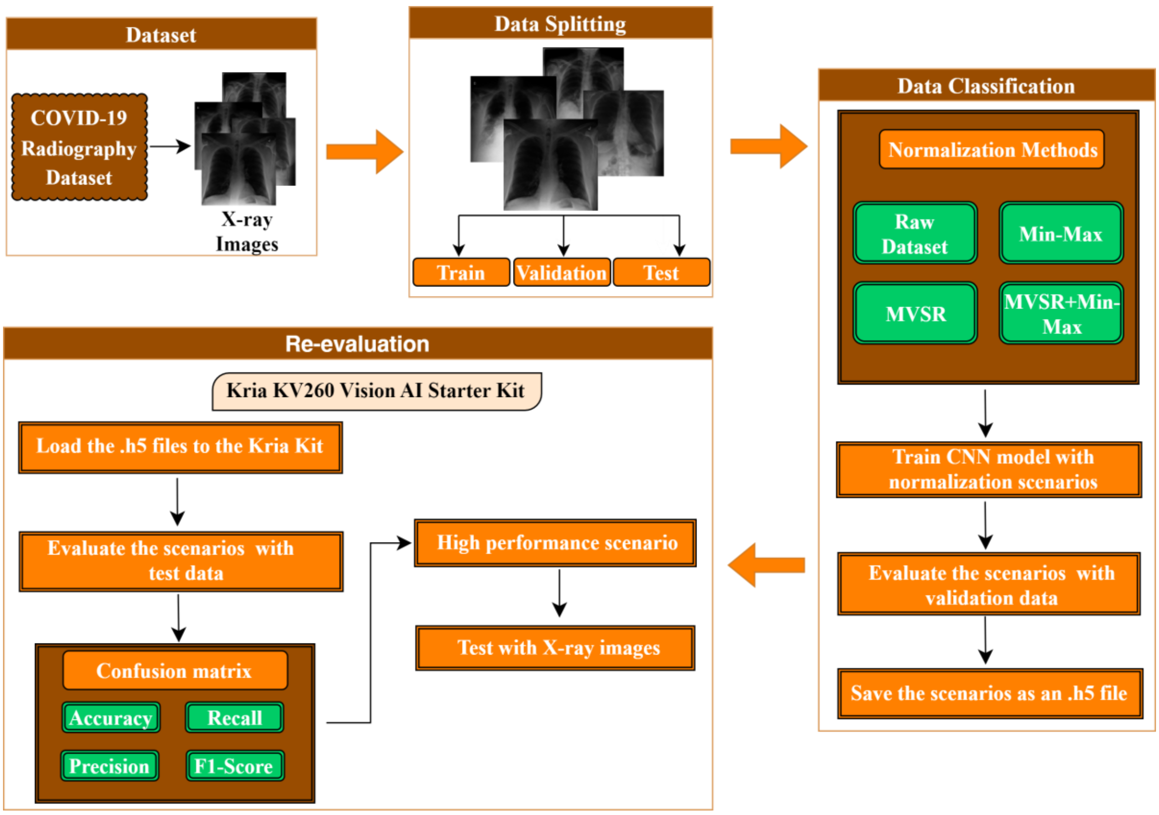


Figure 1. Overview of the proposed system.

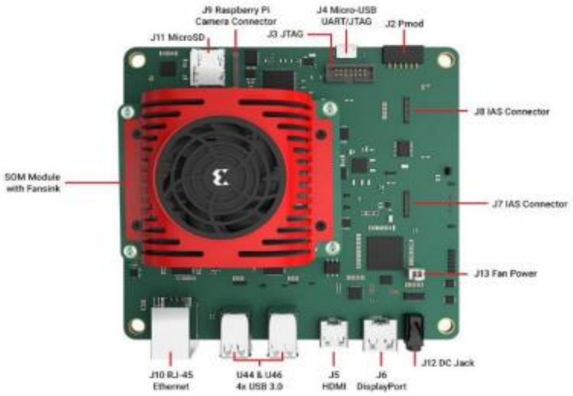


Figure 2. XILINX Kria KV260 Vision AI Starter Kit [13]

COVID-19 Radiography dataset

COVID-19, which has caused significant global losses, spreads rapidly through the respiratory tract and primarily affects the lungs, impairing vital functions. Lung radiography images are used for diagnosing the disease, and thus, lung radiography datasets are commonly employed for COVID-19 detection using AI. When the virus first emerged, researchers had limited data. However, as the number of cases increased, databases containing tomography images that displayed viral infection symptoms were created. Scientists from various disciplines collaborated to establish these databases. As a result, COVID-19 diagnosis gained momentum with AI applications. A research team from Dhaka University in Qatar, Bangladesh, and Pakistan, along with collaborators in Malaysia, created a database of chest X-ray images for COVID-19 positive cases in collaboration with medical doctors [11, 12]. The COVID-19 dataset, which consists of normal and viral pneumonia images, was released gradually with the normal and other lung infection datasets. The initial release of the dataset included a collection of chest X-ray images categorized into 219 cases of COVID-19, 1341 instances of normal scans, and 1345 cases of viral pneumonia. Subsequent updates saw an expansion of the dataset, with the COVID-19 category growing to include 1200 chest X-ray images. The most recent update further enriched the database to encompass 10,192 normal images, 6012 images of lung opacity indicative of non-COVID-19 infections, and 1345 images of viral pneumonia. Additionally, the update provided corresponding lung masks and increased the count of COVID-19 positive images to 3616 [11, 12].

This dataset is publicly accessible and has also been awarded the COVID-19 Dataset Award by the Kaggle community [14]. The dataset used in this study contains normal and COVID-19 chest X-ray images divided into three parts: 600 for testing, 4676 for training, and 1985 for validation.

Normalization methods

In AI applications, one of the most important element of success is undoubtedly the dataset. Large dataset increases performance but also leads to disadvantages such as increased computational load and memory usage. Therefore, processing datasets of appropriate size most effectively will minimize these disadvantages. Pre-processing of data encompasses the necessary procedures to condition the data prior to its utilization in Machine Learning (ML) or DL models [15].

Many studies have been done on the normalization process, which is such a big part of AI, and techniques have been developed as a result.

Min-Max Normalization is one of the most common data normalization technique and is also known as feature scaling. As the name suggests, it scales the data points to a range between *0* and *1* by setting the minimum value in the dataset to 0, the maximum value to 1, and scaling all values in between proportionally [16].

The main idea behind Min-Max Normalization is to ensure that variables with different scales contribute equally to the model and to reduce deviation in the model by re-arranging them. Data scientists generally use this technique to transform features to the same scale using Min-Max Normalization before using them to train ML such as clustering and linear regression [17].

The technique that suppresses the effect of outliers, preserves the relationship between data values, and results in lower standard deviation outputs. If the original data has a large standard deviation, the normalized data will be closer to zero in the output. To perform data normalization, one should deduct the dataset's minimum value from every individual data point. Subsequently, this difference should be divided by the dataset's range, which is the maximum value minus the minimum value. This method effectively scales the data within a normalized range. Equation (1) defines Min-Max Normalization.

$$x_{scaled}= \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

x_{scaled} represents the value obtained after normalization.

MVSR normalization is a normalization technique that has emerged from four newly proposed steps in the literature [10]. The technique is based on the characteristics of mathematical operations consisting of four parts: mean, variance, softmax, and rescale. In most normalization algorithms, the new version of the data population is normalized to have a *0* mean and *1* variance. Typically, the standard deviation, which is the square root of the variance, is employed to determine the dispersion of data points with the dataset's mean [10]. The softmax function is commonly utilized in the final layer of multi-class neural networks to transform the outputs into a probabilistic distribution across various classes [18]. After determining the normalized density values of the inputs, the dataset might exhibit a range of fractional numbers, both negative and positive. The softmax function is then applied to retain the impact of the negative values and the non-linear characteristics of the dataset.

In the rescaling step, all data is converted to 8-bit unsigned integers ranging from 0-255. All the functions are defined in Equations (2), (3), (4), (5), and (6), respectively.

$$\mu_{MVSR} = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{2}$$

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{M} (x_i - \mu_{MVSR})^2 \tag{3}$$

$$\hat{x}_i = \frac{x_i - \mu_{MVSR}}{\sqrt{\sigma^2{}_{MVSR} + \delta}} \tag{4}$$

$$S_{(x)_i} = \frac{e^{\hat{x}_i}}{\sum_j e^{\hat{x}_j}} \tag{5}$$

$$r_i = \frac{S_{(x)_i}}{\max(S_{(x)_i})} * 255 \tag{6}$$

In the equations, μ_{MVSR}, σ^2, $\hat{x}_i$, $S_{(x)_i}$, and r_i represent the mean, variance, normalized density of the input, softmax, and rescaled values, respectively. A small positive number (δ) is added to the variance value in the normalized density of the input $\hat{x}_i$ to avoid the denominator from being zero. In the equations, $\hat{x}_i$ is the input data density, and r_i is the result of MVSR Normalization. Figure 3, shows the application of MVSR Normalization to a chest X-ray image.

Evaluation metrics

Various evaluation criteria assess different aspects of ML algorithms. The experimental evaluation of algorithms and classifiers is a topic of ongoing debate among researchers. Most of the current measurements focus on the classifier's ability to identify classes correctly [19]. A confusion matrix consists of two main sections: real and predicted. By comparing the real classes with the predicted classes, true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values are determined as shown in Figure 4 example.

Accuracy in ML quantifies the proportion of accurate predictions by comparing the count of correct predictions to the overall number of observations within the dataset. This value indicates how often the classifier makes correct predictions. Equation (7) shows the calculation of the accuracy value.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{7}$$

Precision is also a ML metric that measures the proportion of data points that were positive among those that were predicted to be positive by the model. Equation (8) shows the calculation of the precision metric.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{8}$$

Another metric is Recall which measures the proportion of data points that were positive among those that should have been predicted to be positive by the model. Equation (9) shows the calculation of the recall metric.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{9}$$

F1-Score is the metric that measures the harmonic mean of the precision and recall values. Equation (10) shows the calculation of the F1-Score [20].

$$F1\ Score = 2 * \frac{\mathrm{Recall} * \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}} \tag{10}$$

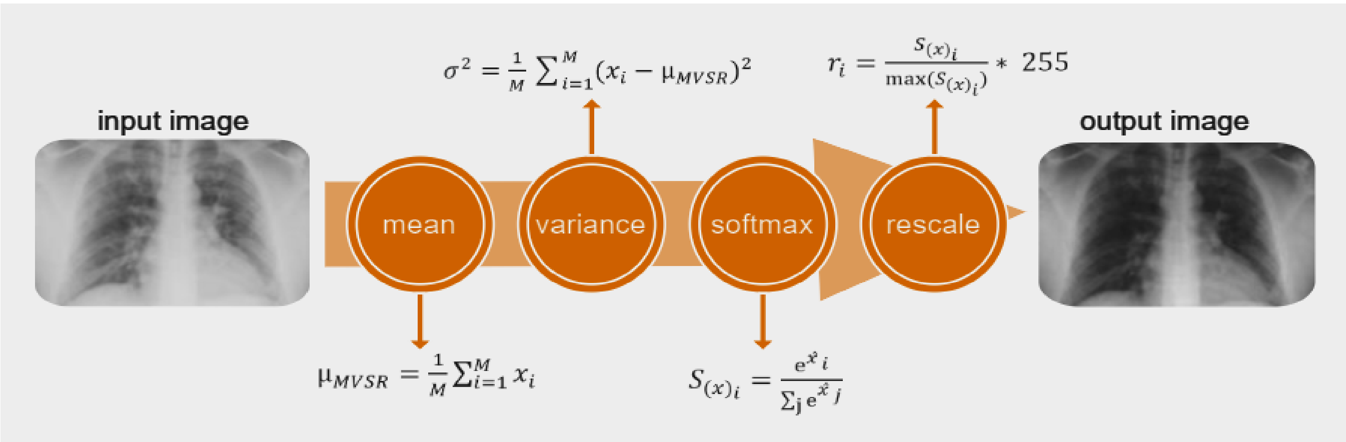


Figure 3. Architecture of MVSR normalization

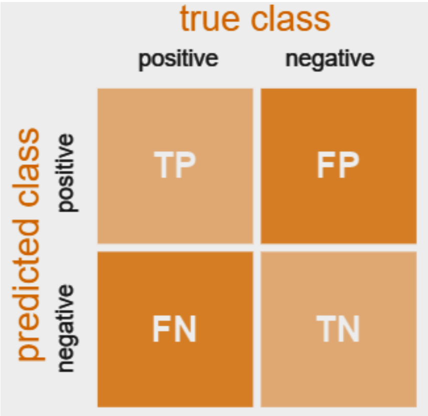


Figure 4. Confusion matrix

Findings and discussion

This study examines the effectiveness of normalization algorithms through experiments conducted on datasets derived from the COVID-19 Radiography Database, which are categorized into two distinct classes: Normal and COVID. During the study, normalization algorithms are applied to the entire dataset, which is divided into three parts: test, training, and validation. At the beginning of the study, a CNN is constructed in the Colab environment for image processing and classification application, and hyper parameters of the model are adjusted to achieve the highest performance. The network is executed with the dataset in four different ways, and the results are recorded to be compared with performance metrics.

The dataset has chest X-ray images of 600 for testing, 4676 for training, and 1985 for validation. The distribution of COVID and Normal classes according to the dataset is shown in Figure 5, and the distribution of classes within the dataset is shown in Figure 6. Figure 7 in the Colab environment shows randomly selected radiography images from both classes.

Firstly, raw data is applied to the network without any normalization algorithm. Then, the same network is fed with the dataset that is normalized using Min-Max Normalization, MVSR Normalization, and MVSR+Min-Max Normalizations, sequentially. MVSR normalization is coded and executed on the Kria KV260 Vision AI Starter Kit, while Min-Max Normalization in the Colab environment. Four different normalization scenarios are evaluated with the validation dataset and saved as a *.h5* file for re-evaluation on the Kria FPGA Kit. And then the *.h5* files are uploaded to the Kria KV260 Vision AI Kit and re-evaluated using a test dataset, with the entire process being executed in the Python language.

The CNN model that is used in applications comprises four convolutional layers and four pooling layers. The convolutional layers use 3x3 filters, with 16, 32, 64, and 128 filters in each layer, respectively. The pooling type is maximum pooling, with a pooling size of 2x2. After the flattening process, a fully connected layer with 256 neurons is added, followed by a two-neuron output layer for all layers except the last layer, which uses the Sigmoid binary classification, and ReLU is chosen as the activation function. The model is trained using the Adam optimizer and Categorical Crossentropy loss function. The model has a total of 3,374,722 trainable parameters. The structure of the CNN model is given in Table 1 and the model is trained for 40 epochs for each application.

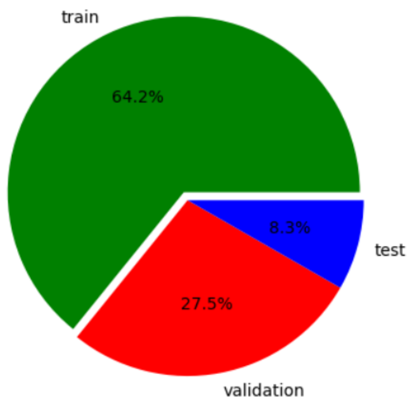


Figure 5. Training, validation, and testing datasets

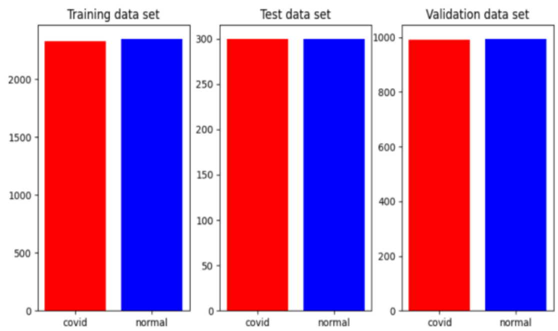


Figure 6. Distribution of COVID and Normal classes in the dataset

The MVSR Normalization algorithm is executed on the Kria KV260 Vision AI Starter Kit during all phases of the network by using the source code given in Table 2. Figure 8 shows the effect of the MVSR normalization by giving the original and the normalized versions of both normal and COVID-19 chest X-ray images.

Table 1. CNN architecture

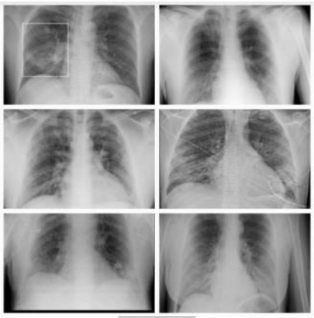
Layers	Layer Configurations			
	S	N	W	H
Input	-	-	150	150
Conv1	3x3	16	150	150
Max-Pool1	2x2	16	75	75
Conv2	3x3	32	75	75
Max-Pool2	2x2	32	38	38
Conv3	3x3	64	38	38
Max-Pool3	2x2	64	19	19
Conv4	3x3	128	19	19
Max-Pool4	2x2	128	10	10
FC	256			
Output	2			

Figure 7. Randomly selected images of COVID and Normal classes

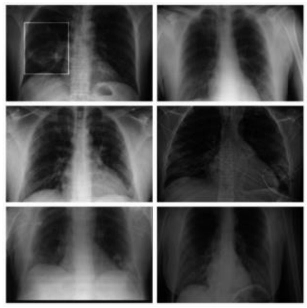
Table 2. MVSR Normalization source codes on the FPGA application platform

```
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
x1 = img.astype(np.float64)
xx, xy = x1.shape
muB = np.sum(x1) / (xx * xy)
sigma2 = (x1 - muB) ** 2
sigb = np.sqrt(np.sum(sigma2) / (xx * xy) + 1e-8)
xnew = (x1 - muB) / sigb
```

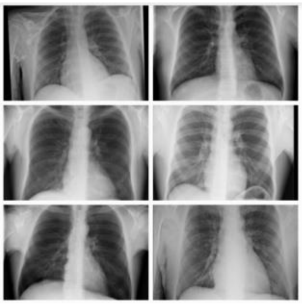
```
# Softmax
a = np.exp(xnew)
 ss = 1 / np.sum(a)
 a = a * ss
 # Rescale
 a = a / np.max(a)
 a = a * 255
 a = np.round(a).astype(np.uint8)
 return a
```



(a) COVID-19 X-Ray



(b) COVID-19 MVSR



(c) Normal X-Ray

(d) Normal MVSR

Figure 8. (a) and (c) are the original chest X-ray images (b) and (d) are the normalized versions of (a) and (c)

Table 3-4 compares the evaluation metrics of four scenarios. It can be stated that the highest average performance is achieved with MVSR+Min-Max Normalization by comparing four performance metrics which are accuracy, precision, recall, and F1-Score. The loss and performance graphs of the MVSR+Min-Max normalization scenario is illustrated in Figure 9. Confusion matrices obtained by evaluating four normalization scenarios on a test dataset are shown in Figure 10.

Table 3. Performance metrics of each normalization scenario

Normalization Type	Image Class	Precision		Recall		F1-Score	
		Validation (%)	Test (%)	Validation (%)	Test (%)	Validation (%)	Test (%)
Raw Dataset	**COVID**	93	93	92	91	92	92
Min-Max		95	93	89	87	92	90
MVSR		97	97	91	83	94	89
MVSR+Min-Max		93	91	97	96	95	93
Raw Dataset	**Normal**	92	91	93	93	92	92
Min-Max		90	88	95	93	92	90
MVSR		92	85	97	97	94	91
MVSR+Min-Max		97	95	92	90	95	93

Table 4. Model performances of each normalization scenario

	Accuracy	
	Validation (%)	Test (%)
Raw Dataset	92	92
Min-Max	92	90
MVSR	94	90
MVSR+Min-Max	95	93

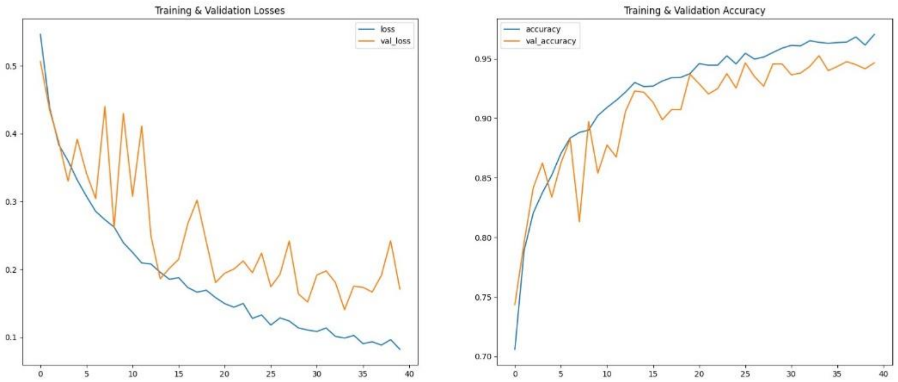
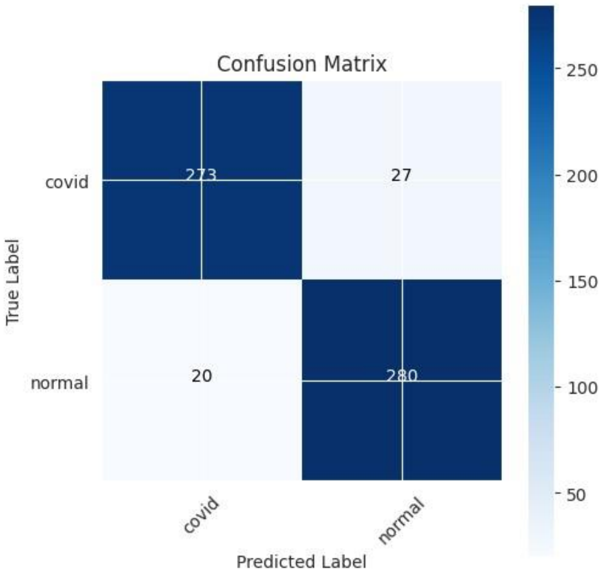
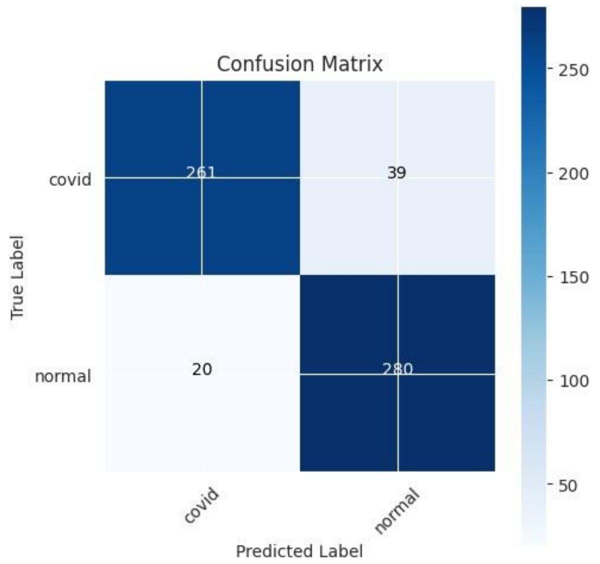


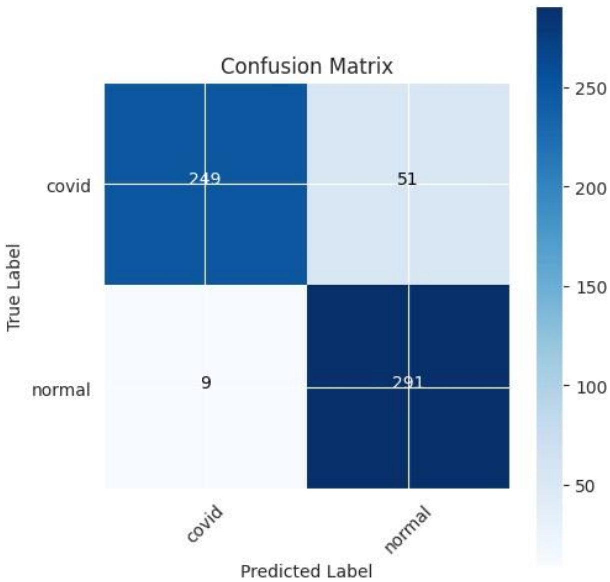
Figure 9. Loss and accuracy graphs of the CNN model for the MVSR+Min-Max Normalization scenario



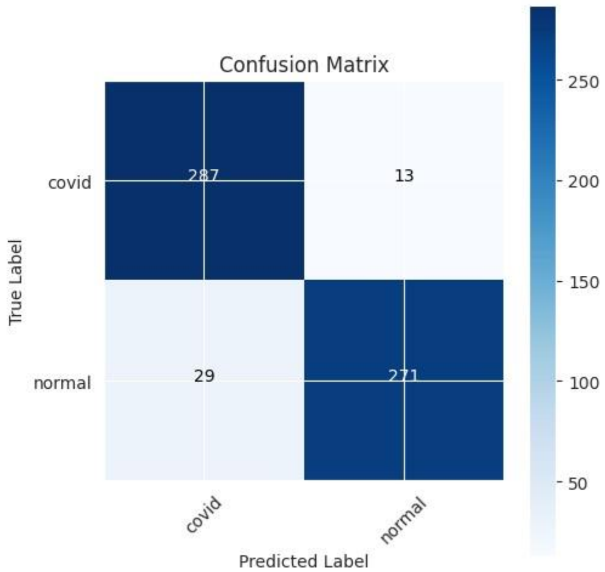
(a) Raw dataset



(b) Min-Max



(c) MVSR



(d) MVSR+Min-Max

Figure 10. The confusion matrices of four normalization scenarios

This study is conducted in real-time on a XILINX Kria Kit operating at a frequency of 1.2 GHz, within an Ubuntu operating system environment installed on the Kit, utilizing the Jupyter Notebook interface integrated within the Visual Studio Code environment.

During the application, only 20% of the 64-bit quad-core Cortex A53 processor is utilized. Of the total used space, 7% was occupied by the GNOME interface, while GNOME Shell accounted for 9%. Out of the 4 GB of RAM available to the processor, the operating system consumed 1 GB, and the network operations utilized another 1 GB, totaling 2 GB of RAM used.

Upon examining the confusion matrices of the test dataset, it is observed that the COVID class is predicted with the highest accuracy rate across all scenarios.

The MVSR+Min-Max scenario which is the best choice for test and validation performance is applied to the CNN and executed. In the given scenario, the FPGA Kit environment is utilized to display 10 COVID-19 and 10 normal X-ray images, with the class probabilities of the outputs obtained being delineated in Table 5.

As shown in Figure 11, a real-time testing platform can be organized using the camera of the same Kria kit, and classification can be performed by capturing images from the camera.

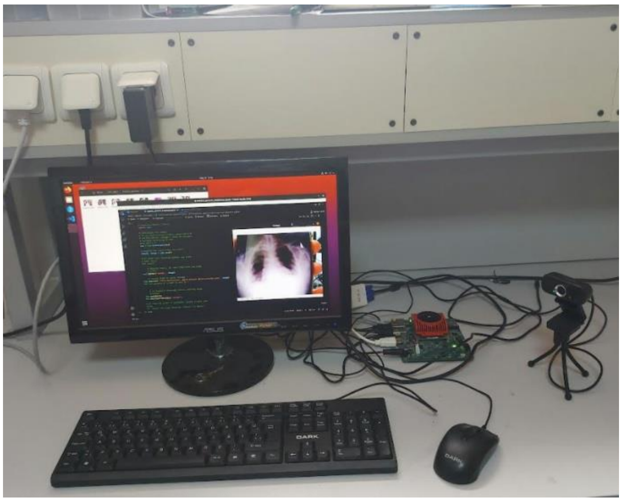


Figure 11. The working mechanism of the real-time testing platform used the Kria KV260 Vision AI Starter Kit

Table 5. The implementation results with the experimental setup of the MVSR+Min-Max Normalization algorithm

True Label of Images		Model decision possibility		Predicted Label of Images
		COVID	Normal	
COVID	Image-1	0.9696682	0.00152129	COVID
	Image-2	0.9979092	0.00114199	COVID
	Image-3	9.9999988e-01	4.0557937e-04	COVID
	Image-4	0.8729858	0.09654933	COVID
	Image-5	0.8520709	0.05296244	COVID
	Image-6	9.998508e-01	9.313257e-04	COVID
	Image-7	0.98699224	0.00118503	COVID
	Image-8	0.96766907	0.17125143	COVID
	Image-9	0.96306807	0.208236	COVID
	Image-10	0.8314386	0.08490623	COVID
Normal	Image-1	0.06010169	0.98435336	Normal
	Image-2	2.4713738e-05	9.9999666e-01	Normal
	Image-3	0.11396333	0.9211578	Normal
	Image-4	0.23098588	0.4459954	Normal
	Image-5	1.9339639e-05	9.9999750e-01	Normal
	Image-6	0.01860543	0.996901	Normal
	Image-7	6.464141e-06	9.999987e-01	Normal
	Image-8	0.02560098	0.9937652	Normal
	Image-9	3.9540315e-05	9.9999279e-01	Normal
	Image-10	0.19743828	0.78217196	Normal

Conclusion

AI technologies are widely used in various fields such as education, healthcare, autonomous vehicles, home appliances, and social platforms. The choice of normalization algorithm depends on the task performed in such a diverse technology. This is why every normalization algorithm provides easy access to the optimal result for every application. Normalization algorithms are employed to accelerate data processing and suppress outliers.

According to this study, the highest accuracy is achieved MVSR+Min-Max Normalization scenario with 95% and 93% in validation and test datasets, respectively. The performances of other applications on the test dataset are 92%, 90%, and 90% respectively for models raw dataset, Min-Max Normalization, and MVSR Normalization. The highest precision is achieved MVSR+Min-Max Normalization scenario with 97% and 93% averages for both COVID and Normal cases in the validation and test datasets, respectively. Also, the MVSR+Min-Max application has achieved the highest F1-Score of 95% and 93% respectively for test and validation data.

When the confusion matrix parameters are examined, the most correct prediction for the COVID class is obtained in MVSR + Min-Max Normalization, while the most incorrect prediction for the Normal class is in the same normalization algorithm. However, considering the total performance of the network, the number of correct predictions in the images belonging to the COVID class tolerates the number of incorrect predictions in the images belonging to the Normal class. The normalization with the highest number of correctly predicted images belonging to the Normal class is MVSR Normalization.

The results obtained in this study, conclude that MVSR Normalization facilitates the diagnosis of COVID-19 disease by the CNN model in chest X-ray images. The findings align with existing literature but are influenced by the smaller size of the datasets utilized for training, validation, and testing as noted in reference [10]. Specifically, the training and validation datasets are approximately 19 times smaller, while the testing dataset is three times smaller. It is widely acknowledged that employing larger training and validation datasets often leads to improved model performance. However, expanding the testing dataset may introduce greater diversity, enabling a more comprehensive evaluation of the model's robustness while potentially compromising its performance.

The observed decrease in performance when using larger datasets can be attributed to various factors, including the complexity of the model, data quality, and preprocessing techniques. These findings underscore the ongoing need for research and analysis in the field of ML. Future studies should concentrate on addressing these areas to enhance model performance and understanding.

In the pandemic scenarios, the rapid and accurate diagnosis of infectious diseases is of critical importance. The application of AI can significantly reduce the workload of healthcare professionals and expedite diagnostic processes. Tools such as the Kria KV260 AI Starter Kit, which are accessible and compatible with widely-used programming languages like Python, aim to enhance support for healthcare workers in such emergencies. Future research endeavors seek to increase efficiency, with the ultimate goal of improving the speed and accuracy of disease diagnosis and treatment processes.

Acknowledgment

This paper is produced from the part of Merve Zirekgür's Master Thesis.

Conflict of Interest

There is no conflict of interest.

Similarity Rate (Turnitin): 17%

References

[1] M. Riva, Batch Normalization in convolutional neural networks. Baeldung, San Francisco, 2023.

[2] An intro to Convolutional Neural Networks (CNN), https://lamiae-hana.medium.com/an-intro-to-convolutional-neural-networks-cnn, Accessed 20 March 2023.

[3] I., Rothe, H., Susse and K., Voss, The method of normalization to determine invariants. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(4), 366-376, 1996.

[4] S., Ioffe and C., Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning (pp. 448-456), 2015.

[5] J. L., Ba, J. R., Kiros and G. E., Hinton, Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[6] Y., Wu and K., He, Group normalization. Proceedings of the European Conference on Computer Vision (ECCV) (pp. 3-19), 2018.

[7] X., Wu, H., Hui, M., Niu, L., Li, L., Wang, B., He and Y., Zha, Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: a multicentre study. European Journal of Radiology, 128, 109041, 2020.

[8] R. M., Pereira, D., Bertolini, L. O., Teixeira, C. N., Silla Jr. and Y. M., Costa, COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios. Computer Methods and Programs in Biomedicine, 194, 105532, 2020.

[9] T., Rahman, A., Khandakar, Y., Qiblawey, A., Tahir, S., Kiranyaz, S. B. A., Kashem and M. E., Chowdhury, Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. Computers in Biology and Medicine, 132, 104319, 2021.

[10] S., Yaman, B., Karakaya, and Y., Erol, A novel normalization algorithm to facilitate pre-assessment of COVID-19 disease by improving accuracy of CNN and its FPGA implementation. Evolving Systems, 1-11, 2022.

[11] M. E. H., Chowdhury, T., Rahman, A., Khandakar, R., Mazhar, M. A., Kadir and A. Emadi, Can AI help in screening viral and COVID-19 pneumonia?. IEEE Access, vol. 8, pp. 132665-132676, 2020, doi: 10.1109/ACCESS.2020.3010287.

[12] T., Rahman, A., Khandakar, Y., Qiblawey, A., Tahir, S., Kiranyaz, S. B. A., Kashem and M. E. Chowdhury, Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. Computers in Biology and Medicine, 132, 104319, 2021.

[13] AMD, Kria KV260 Vision AI Starter Kit data sheet (DS986). California, 2022.

[14] COVID-19 Radiography Database, https://www.kaggle.com/datasets/tawsifurrahman/COVID19-radiography-database, Accessed 8 April 2024.

[15] M., Canesche, L., Bragança, O. P. V., Neto, J. A., Nacif and R., Ferreira, Google Colab CAD4U: Hands-on cloud laboratories for digital design. 2021 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5) IEEE, 2021.

[16] The Standard Normal Distribution Calculator, Examples & Uses, https://www.scribbr.com/statistics/standard-normal-distribution/, Accessed 18 July 2023.

[17] J. McEwen, A Gentle introduction to min-max data normalization. Texas, 2022.

[18] D., Singh, B., Singh, Investigating the impact of data normalization on classification performance. Appl Soft Comput 97:105524, 2020.

[19] M., Sokolova, N., Japkowicz and S., Szpakowicz, Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Proceedings 19 (pp. 1015-1021), Springer Berlin Heidelberg, Hobart, Australia, December 4-8 2006.

[20] Z. C., Lipton, C., Elkan and B. Narayanaswamy, Thresholding classifiers to maximize F1-Score. arXiv preprint arXiv:1402.1892, 2014.