Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Sustainable Cities and Society 80 (2022) 103713

Contents lists available at ScienceDirect

Sustainable Cities and Society

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

An efficient hardware architecture based on an ensemble of deep learning models for COVID -19 prediction

Sakthivel R [a], I. Sumaiya Thaseen [b], Vanitha M [b], Deepa M [b], Angulakshmi M [b], Mangayarkarasi R [b], Anand Mahendran [c], Waleed Alnumay [d], Puspita Chatterjee [e,*]

[a] School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India
[b] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India
[c] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India
[d] Riyadh Community College, CS Department, King Saud University, Saudi Arabia
[e] Department of CS, Tennesee State University, TN, USA

ARTICLE INFO

Keywords:
Accuracy
COVID-19
Deep learning
Ensemble
Pre-training
Performance
Latency of CNN
Data-aware computational unit

ABSTRACT

Deep learning models demonstrate superior performance in image classification problems. COVID-19 image classification is developed using single deep learning models. In this paper, an efficient hardware architecture based on an ensemble deep learning model is built to identify the COVID-19 using chest X-ray (CXR) records. Five deep learning models namely ResNet, fitness, IRCNN (Inception Recurrent Convolutional Neural Network), effectiveness, and Fitnet are ensembled for fine-tuning and enhancing the performance of the COVID-19 identification; these models are chosen as they individually perform better in other applications. Experimental analysis shows that the accuracy, precision, recall, and F1 for COVID-19 detection are 0.99,0.98,0.98, and 0.98 respectively. An application-specific hardware architecture incorporates the pipeline, parallel processing, reusability of computational resources by carefully exploiting the data flow and resource availability. The processing element (PE) and the CNN architecture are modeled using Verilog, simulated, and synthesized using cadence with Taiwan Semiconductor Manufacturing Co Ltd (TSMC) 90 nm tech file. The simulated results show a 40% reduction in the latency and number of clock cycles. The computations and power consumptions are minimized by designing the PE as a data-aware unit. Thus, the proposed architecture is best suited for Covid-19 prediction and diagnosis.

1. Introduction

Developing low power consumping, high throughput, fast computing solutions with high memory density and reliability, by incorporating the human intellengence and by balancing the social, economic and environmental sustainability is the need of this hour in building sustainable smart cities. The World Health Organization (WHO) professed that COVID-19 viral infection as an ongoing pandemic (Ahmad et al., 2021). The disease has affected more than 214 million people globally and over 4 million life losses around the world till August 2021. The illnesses usually affect the respiratory system such as the lungs and also result in symptoms similar to Pneumonia (Rubin et al., 2020). Reverse Transcription-Polymerase Chain Reaction (RT-PCR) study is the highest quality level to confirm the disease. Current RT-PCR test kits are minimal in number, the results of the test are obtained after long time, and there is a high probability of health care personnel becoming infected with the disease during the test, demands the use of other diagnostic approaches as an alternative to these test kits. The proposed work is a fast detection method using X-ray image analysis that would be a contribution to the society.

In under developed and developing countries where the doctor to patient ratio is very weak there is a need to provide a fair and equal healthcare facilities for everyone in this world. A modern technology based innovative solution which could cater the need of early prediction, isolation and treatment of an individual from the pandemic is the need of an hour. The prediction of COVID-19 using the proposed model can be helpful for medical experts to prioritize the resources correctly for COVID-19 prediction. In addition, if this prediction model is deployed in various cities, there will be minimal disruption of global supply chains with negligible job losses and impact on livelihood.

* Corresponding author.
E-mail address: pushpita.c@ieee.org (P. Chatterjee).

https://doi.org/10.1016/j.scs.2022.103713
Received 29 January 2021; Received in revised form 21 January 2022; Accepted 21 January 2022
Available online 3 February 2022

A few investigations deployed Deep Learning (DL) algorithms such as Convolutional Neural Network (CNN) models for distinguishing, restricting, or estimating the development of COVID-19 Virus in utilizing CXRs and Computed Tomography (CTs) (Rajaraman & Antani, 2020; Rajaraman et al., 2020). However, the Computer Aided Diagnosis (CADx) resolutions that utilize DL techniques for infection recognition including COVID-19 are huge confinements in the current methodologies based on the dataset type, model architecture, assessment, size. Thus, these concerns suggest new investigations to fulfill the crucial need for COVID-19 identification using CXRs.Ensemble deep learning classifiers are preferred for health care (Zhou et al., 2021) because the overall classification result increases in comparison to the individual classifiers. The ensemble accuracy can be higher and other evaluation parameters like sensitivity and specificity also increase therefore it is can be effective for the detection of COVID-19 rapidly. CNN's have proved to be an integral part of Machine Learning in recent times. With the unabated influence of Artificial Intelligence on every sphere of life, researchers have had the motivation to devise novel algorithms and architect Very Large Scale Integration (VLSI) implementations for the efficient and fast undertaking of those algorithms. One such revolutionary algorithm being the convolutional neural networks. CNN's have shown exemplary performance in the field of computer vision for segmentation, classification, detection, and retrieval-related tasks. Most of the companies like Intel, Google, and Facebook, etc. started exploring and using the AI algorithms and hardware architectures to the great extent (Chen, Krishna, Emer, & Sze, 2016).

The best feature of CNN architecture is its ability to extract details regarding the spatial and time domain. The computational complexity level of the CNN network is very high which pushes the hardware developers (Han et al., 2015) to come up with reconfigurable Field Programmable Gate Array (FPGA) architecture or ASIC-based architecture which could reduce the power, latency, and computational times of DotNetNuke (DNN). Numerous research works have been published in the area of efficient hardware development for PE design, Nonlinear activation function design, etc. The majority of the computational load in the CNNs is due to the convolutions whereas the majority of network parameters are from Fully connected layers. So to say, Fully Connected (FC) layers are easy to implement in hardware, but they require high power consumption due to frequent memory accesses (Han et al., 2016).

As the epidemic continues progressing, it negatively affects the flexibility of the global society from every aspect of daily life, environment, economy, and others, and thus it raises serious attention from health planners and policymakers internationally to attain the Sustainable Development Goals. In order to address the exceptional challenge, the scientific evidence can be obtained from the deep investigation of the dynamic progress of COVID-19 transmission. Accordingly, potential strategies and interventions can be formulated at an early stage for controlling or even blocking the sustained propagation, contributing to minimize the infectious and mortality rates. This work tries to develop a hardware software co design based feasible solution for a smart health care system.

The contribution of this study are,

- Analyze the single deep learner performance for choosing the appropriate models in the ensemble.
- Develop a deep learning ensemble model for predicting the COVID-19 effectively in terms of accuracy and other performance measures.
- Optimize the deep learning computations using a Reconfigurable/ ASIC hardware design for minimizing latency and power consumption.

The rest of the paper is structured as follows: the literature of various deep ensemble learning models is discussed in section 2. The proposed model is explained in section 3. The experimental analysis and setups are discussed in 4. Results and discussion of the results are done in section 5 The conclusion is given in section 6.

2. Literature survey

In this section, the literatures of various deep learning models for COVID-19 are analyzed. The COVID-19 is a dangerous disease as it spreads fast in comparison to other viruses. Ensemble-based Deep learning is widely used for predicting it. Three important deep learning models GoogleNet, AlexNet, and ResNet are integrated by majority voting for COVID prediction. This approach detects COVID better than other classifiers (Otoom, Otoum, Alzubaidi, Etoom, & Banihani, 2020). Various ensemble of deep learning models are deployed for better COVID-19 prediction (Chowdhury, Kabir, Rahman, & Rezoana, 2020; Elgendi, Fletcher, Howard, Menon, & Ward, 2020; Ghoshal & Tucker, 2020; Haghanifar, Majdabadi, Choi, Deivalakshmi, & Ko, 2020; Hussain et al., 2021; Karim et al., 2020; Melin, Monica, Sanchez, & Castillo, 2020; Polsinelli, Cinque, & Placidi, 2020; Shoeibi et al., 2020; Toraman, Alakus, & Turkoglu, 2020).

Vantaggiato et al. (2021) created two databases to identify COVID-19 lung diseases. In the first database, they have considered three classes to distinguish COVID-19, Health and Pneumonia and in the second database, they have considered five classes to distinguish COVID-19, Lung Opacity No Pneumonia, Healthy, Viral Pneumonia, and Bacterial Pneumonia. They evaluated three CNN architectures like ResNet-50, Inception-v3, and DenseNet-161 to distinguish between different lung diseases and proposed an Ensemble-CNN approach. The results show high performance resulting in 98.1% accuracy in three-class and five-class scenarios respectively for identifying COVID-19 infection.

Shalbaf and Vafaeezadeh (2021) improved the recognition performance by using 15 pre-trained convolutional neural network architectures. Deep transfer learning architecture like EfficientNetB3, Xception, EfficientNetB5, Inception_resnet_v2, and EfficientNetB0 achieved better results in identifying COVID or any other lung diseases. CNN models like DenseNet201, Resnet50V2, and Inceptionv3 have been adopted in this proposed work (Das et al., 2021). They have trained the models individually for independent prediction and combined it using the weighted average ensemble technique and achieved the classification accuracy of 91.62%. They have developed a GUI interface that will be useful for doctors to detect COVID patients.

During the past few years, it has been a trend to increase the number of layers of convolution to improve the Miss-classification Rates (MCR) (Lane & Georgiev, 2015). This trend has led the CNNs to become extremely bulky and high-demanding on memory and energy consumption hence limiting their implementation on resource-constrained and battery-operated devices (Ardakani, Condo, & Gross, 2016). Also, all-purpose CPUs and GPUs have shown to be unbearably un-optimized for latency and energy constraints.

The limitation of the related works is that prediction of a single deep learner model cannot be trusted due to the minimum number of data samples. In addition, the computing power is not sufficient to use deep learning models for COVID-19 prediction. Few deep learning models result in generalization metric issues (Shorten, Khoshgoftaar, & Furht, 2021). The diagnosis of COVID19 is based on the assessment and evaluation of the radiologist's CT image. However, this work is tedious, and there is often a high degree of inter-server variability which leads to uncertainty. Therefore, to overcome the stated limitations, an automated, reliable, and repeatable approach using advanced deep learning is required. This system can overcome these limitations and can be used anywhere without the need for highly trained radiologists (Ali et al., 2020). The dataset is still insufficient for a practical and accurate deep learning solution that can be accepted as a standard for identifying COVID19 infection in patients from radiographic images. Many kinds of literature have focused on reducing the memory access time using stochastic computing (Smithson, Boga, Ardakani, Meyer, & Gross, 2016). Researchers have concentrated on developing hardware for efficient computation, less latency, etc. Thus, in the proposed deep learning ensemble, an efficient parallel and pipelined fully connected

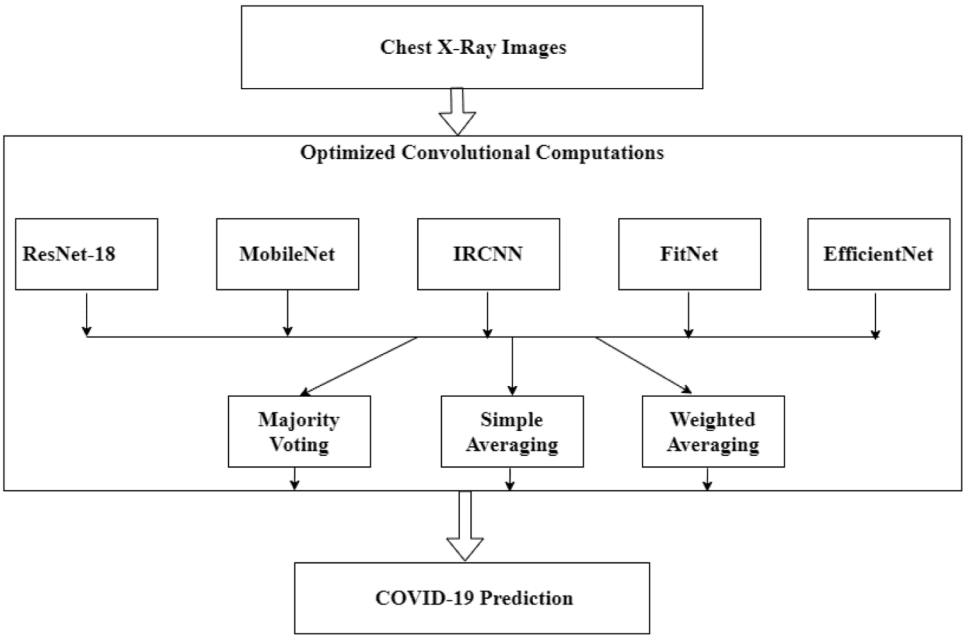


Fig. 1. General architecture of the proposed model.

architecture for COVID- 19 prediction is built that could provide lower latency, lower power consumption, and less computational complexity thereby resulting in best performance for classification and recognition (Shin, Lee, Lee, & Yoo, 2017; Wang, Zhou, Han, & Yoshimura, 2017).

Data privacy, epidemic pattern unpredictability, regulation and clarity, and the differentiation between COVID-19 and non-COVID-19 symptoms are among the obstacles and issues raised by existing investigations (Bhattacharya et al., 2021). Single Shot Multibox Detector is used for face detector and MobilenetV2 architecture used as classifier framework (Nagrath et al., 2021).The proposed method provides higher accuracy and F1 score for COVID face mask detection

2.1. Ensemble models

In the proposed model, different CNN like ResNet, FitNet, IRCNN, MobileNet and Efficientnet are integrated to form ensemble. A simplified ResNet (Li, Jiao, Han, & Weissman, 2016) is built in the proposed model by calculating a minimum distance among all the data points and labels. In the training process of a zero-initialized deep residual network, the weights are near the initial point. The network is optimized by gradient descent when the condition number is small. An intelligent teacher model is incorporated into the FitNet. Lopez-Paez et al. (Lopez-Paz, Bottou, Schölkopf, & Vapnik, 2015) developed the process of generalized distillation and presented that generalized distillation minimizes the knowledge distillation if x_i*=x_i for all *'i'* with few constraints and reduces to Vapnik and Izmailov (2015) learning if x_i* is a privileged description of x_i with few constraints.

- Learn teacher $f_t \epsilon F_t$ utilizing the input-output pairs (xi*,yi)ni=1 and Equation(1).
- Determine teacher soft labels {σ(ft(xi*)/T)}nt=1, using temperature parameter $T>0$
- Learn student fsϵ FS using the input output pairs $((xi*, yi)ni = 1, \{(xi*, si)ni = 1\}$ and imitation parameter

$$f_i = \underset{f \in F_i}{argmin} \frac{1}{n} \sum_{i=1}^{n} \mathscr{L}\left(y_i, \sigma(f(x_i))\right) + \Omega(||f||) \quad (1)$$

IRCNN network is one of the latest advancements in deep learning models, such as Inception Nets (Chen & Su, 2018) and RCNNs. The tasks of each Recurrent Convolution Layer (RCL) in the IRCNN block is observed as a pixel ordered at *(i,j)* for a specific information test in the RCL on the kth include map. This is the yield $y_{1ijk}(t_1)$at time step t_1 which is written in Eq. (2) below:

$$y_{1ijk}(t_1) = (w_1 f_k)\, T_1\, x_{1f}^{(i,j)}(t_1) + (w_{1k}^r) T_1\, x_{1f}^{(i,j)}(t_1 - 1) + b_{1k} \quad (2)$$

Here $x_{1f}^{(i,j)}$ (t_1) and $x_{1f}^{(i,j)}$ *$(t_1 - 1)$)* denotes the inputs for RCL and a standard convolutional layer correspondingly. $w_{1k}{}^r$,$w_{1k}{}^f$ and b_{1k} represent the weights for RCL, standard convolutional layer and the bias.The final output at time step *t* is given in Eq. (3):

$$z_1 ijk(t_1) = f_1(y(t_1)) = \max(0, y_1 ijk(t_1)) \quad (3)$$

Where,f_1 denotes the Rectified Linear Unit (ReLU) activation function (Ahmad, Farooq, & Ghani, 2021).

MobileNet is a smoothed-out engineering to build lightweight deep convolutional neural networks and results in a productive model for installed and portable vision applications (Li et al., 2016). Further processing of convolution parameters are described in the Appendix A section.

EfficientNet has the advantages of providing high accuracy, reducing the variables, and FLOPS (Floating Point Operations per Second). The component scaling method is implemented in the width, depth, and resolution of the network dimension. This model has high supremacy in providing high performance. Hence, the model uses the component coefficient to control component scaling equally in all dimensions.

2.2. Sequential least-squares programming method (SLSQP)

SLSQP is utilized to assign weight to each classification learner and the prediction of each classifier is integrated using the soft voting approach. In the ensemble model, SLSQP and the voting approach are used for enhancing prediction accuracy. This approach is used in mathematical problems for which objective function and constraints are twofold continuously differentiable (Melchiorre et al., 2013).

Fig. 2. First Stage Pre Training of CNN models in the proposed approach.

3. Model

3.1. Motivation

The proposed model aims to build efficient hardware-based deep learning ensemble model for predicting the COVID-19. Five deep learning models namely ResNet, FitNet, IRCNN, EffectiveNet, and MobileNet are fine-tuned to improve the class-specific performance of individual models and parallel processing to minimize the fully-connected neural network computations. The hardware architecture acts as a CNN accelerator for massive computing which yields the best results for COVID-19 prediction. Thus, better accuracy is obtained with low latency and less computational power. The five deep learning models are chosen as they have outperformed existing deep learning models in performance as given in the literature for COVID-19 prediction. In general, the EfficientNet and FitNet models (Tan & Le, 2019) provide higher accuracy and better efficiency over existing CNNs.Yan et al. (2021) have demonstrated that in comparison to other models in their proposed work, ResNet-18 has the highest accuracy with few parameters. In addition, ResNet minimizes the training complexity and result in performance improvements in terms of both training and generalization error. It was very closely followed by Akbarian, Seyyed-Kalantari, Khalvati, and Dolatabadi (2020) have demonstrated that FitNet model which is a knowledge transfer learning framework has performed better in classifying medical images by reducing overfitting. IRCNN is one of the effective deep CNN denoiser for image restoration (Zhang, Zuo, Gu, & Zhang, 2017) and has been widely used for denoising of COVID-19 CXR images. Each of the models is chosen in the ensemble due to their advantage over other models in performance.

The flowchart containing the various components is given in Fig. 1. Initially, all the images are fed to five deep learning models which are implemented in a hardware-based architecture. The Convolutional computations are optimized in each of the learners and the results are fed to various ensemble models like majority voting, simple averaging, and weighted averaging. The performance of every ensemble is analyzed and the best results are obtained in the weighted averaging approach as it deploys a dynamic approach for calculating the weights based on the previous classifier results. The fine-tuned CNNs on ensemble models prove to be superior in COVID-19 prediction.

3.2. Recurrent cxr pre-training and fine-tuning

The images are pre-processed using reconstruction techniques such as fuzzy color (Ahmad et al., 2021) and image stacking. In the proposed method, a stepwise training approach is performed. Pre-trained models like ImageNet and custom CNN are arranged for retraining a large collection of images. The features of normal and infected lung images help to train the model (Shastri, Singh, Kumar, Kour, & Mansotra, 2021). Here 90% of datasets are divided for training and 10% for testing during the training phase. From the training dataset, 10% is randomly allocated for validation.

In the initial phase of pre-training as shown in Fig. 2, CNNs are assigned with the pre-trained ImageNet weights and then fine-tuned at middle layers to efficiently study the main feature of dataset images and advance the classification accuracy. The trimmed models are added with padding with zero. Then, classified with a convolutional layer of 3×3 which has 1024 feature maps. A drop-out layer with a 0.5 drop-out ratio, the model is added with the Global Average Pooling (GAP) layer. The final dense layer uses a softmax activation function where the prediction probability is calculated at the last. This model classifies images as normal or infected lungs. The initial stage of the recurrent CXR-precise pre-training is shown in Fig. 2 with the detailed architecture of the pre-trained CNNs.

The information learned from the initial stage pre-trained model is obtained and performed again to classify images as normal lungs or COVID affected lungs which are shown in the second stage of the CXR-specific pre-training model. Fig. 3 shows the second stage pre-training of the CXRs which are pooled in a precise manner. During the training phase, the training data is split into a ratio of 80% for training and 20% for testing. For the validation purpose, a random allocation of 10% of the training data is used.

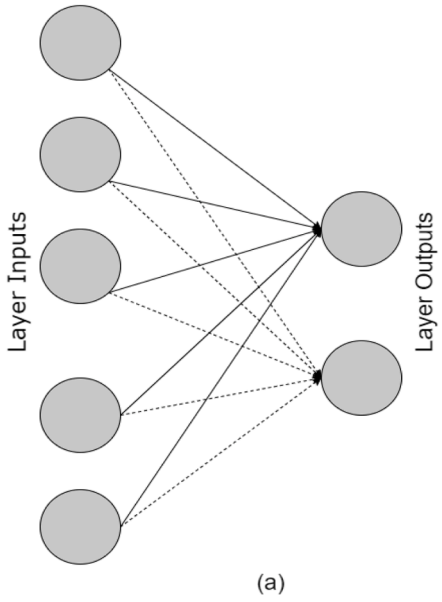
For computer vision, Image Net pre-trained CNNs have been deployed. These models learn varied feature representations containing varying depth levels. Deeper models may not be best for medical images that are limited in quantity as there may be overfitting and generalization loss.

Thus, performance and generalizability can be improved in recurrent CXR-precise pre-training and fine-tuning phases. The minority classes are rewarded by the class weights which prevent biasing error and reduces overfitting. In the proposed work, five deep learning models such as MobileNET, EfficientNET, FITNET, IRCNN, and ResNet are deployed and integrated by majority voting, simple averaging and weighted averaging ensemble which are discussed in the results section.

A hybrid relevance vector machine and logistic regression (RVM-L) model is proposed (Zhu, Ding, Yu, Wang, & Ma, 2021) and experimental details show that in comparison with existing approaches, RVM-L based early warning technique can achieve the prediction accuracy upto 96%. This model can be used to improve the public's awareness of preventive measures, helping society organizing management efforts, and effectively guiding the development of public opinion.

A hybridized algorithm is proposed inZivkovic et al. (2021) between Cauchy exploration strategy beetle antennae search(CESBAS) and

Fig. 3. Second stage pre-training of CNN models in the proposed approach.



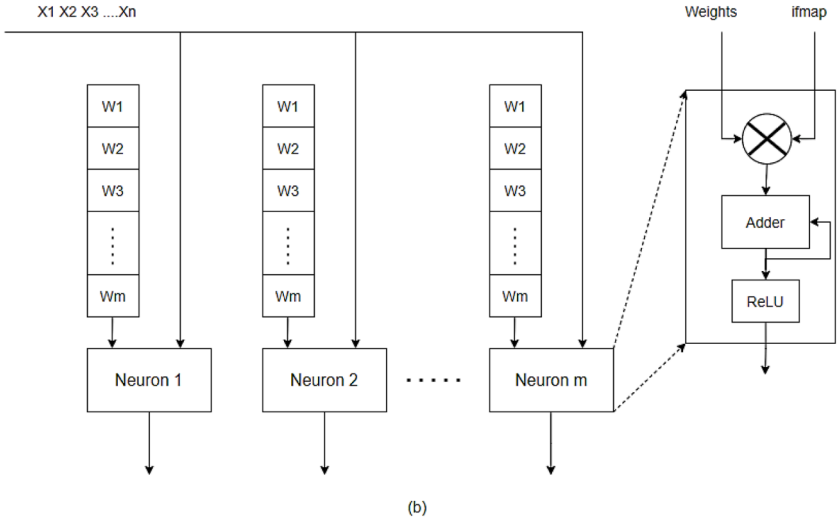


Fig. 4. (a) An FC layer (b) A Semi Parallel Implementation for an FC layer with each neuron.

adaptive neuro-fuzzy inference system (ANFIS) to improve the current time-series prediction. A maximum R2 score of 0.9793 is achieved and conclude that their proposed hybrid model would be beneficial to limit the number of infected people, therefore the health organization does not get overwhelmed by the COVID patients who would need intensive care in hospitals.

3.3. Fully connected network

An FC neural network is a multi-layered network where each layer is composed of 'N' neurons. In Feed-forward FC layers every neuron is connected to the next layer's subsequent neurons. Each connection is given a weight that quantifies the strength of the connection. FC networks can learn non-linear abstractions of the data.

Further processing of FC layer parameters are described in the appendix section.

The main computation of an FC layer involves lots of vector multiplications which increase the area power timing. A practical convolutional computation may look like the computation in Fig. 4. (a) fully connected network and Fig. 4 (b) shows a Semi Parallel Implementation for an FC layer with each neuron having its weights stored in the registers

3.4. Convolutional layer

The convolutional layers consist of neurons enumerated in 3 dimensions: height H, width W, and channel C. Each convolutional layer transforms 3D input pixels (a set of C_{in} 2D maps) to 3D output activation maps (a set of C_{out} activation maps). This transformation is carried out by a 4D filter (a set of C_{out} 3D filters). Each set of 3D filters convolves with the 3D input pixels to give out a single 2D $H_{out} \times W_{out}$ plane of the output computed pixels. In the end, a 1D bias is added to the 3D output pixels. As illustrated in the appendix B the simple 2D convolution is performed.

The existing hardware architecture that is needed for computing the $X_1 \ldots X_{25}$ pixels requires 25 clock cycles and the output layer requires 9 neurons. For each cycle, it needs 25 multiplication operations excluding the addition and bias weight operation which is computationally

Table 1
Optimized computation by resource utilization for Convolutional Computations.

The first row of the output				The second row of the output			The third row of the output		
Clock cycles	Neuron #1	Neuron #2	Neuron #3	Neuron#4	Neuron#5	Neuron#6	Neuron#7	Neuron#8	Neuron#9
1	$X_1 \times W_1$	$X_2 \times W_1$	$X_3 \times W_1$	$X_{16} \times W_7$	$X_{17} \times W_7$	$X_{18} \times W_7$	$X_{21} \times W_7$	$X_{22} \times W_7$	$X_{23} \times W_7$
2	$X_2 \times W_2$	$X_3 \times W_2$	$X_4 \times W_2$	$X_{17} \times W_8$	$X_{18} \times W_8$	$X_{19} \times W_8$	$X_{22} \times W_8$	$X_{23} \times W_8$	$X_{24} \times W_8$
3	$X_3 \times W_3$	$X_4 \times W_3$	$X_5 \times W_3$	$X_{18} \times W_9$	$X_{19} \times W_9$	$X_{20} \times W_9$	$X_{23} \times W_9$	$X_{24} \times W_9$	$X_{25} \times W_9$
4	$X_6 \times W_4$	$X_7 \times W_4$	$X_8 \times W_4$	$X_6 \times W_1$	$X_7 \times W_1$	$X_8 \times W_1$	$X_{16} \times W_4$	$X_{17} \times W_4$	$X_{18} \times W_4$
5	$X_7 \times W_5$	$X_8 \times W_5$	$X_9 \times W_5$	$X_7 \times W_2$	$X_8 \times W_2$	$X_9 \times W_2$	$X_{17} \times W_5$	$X_{18} \times W_5$	$X_{19} \times W_5$
6	$X_8 \times W_6$	$X_9 \times W_6$	$X_{10} \times W_6$	$X_8 \times W_3$	$X_9 \times W_3$	$X_{10} \times W_3$	$X_{18} \times W_6$	$X_{19} \times W_6$	$X_{20} \times W_6$
7	$X_{11} \times W_7$	$X_{12} \times W_7$	$X_{13} \times W_7$	$X_{11} \times W_4$	$X_{12} \times W_4$	$X_{13} \times W_4$	$X_{11} \times W_1$	$X_{12} \times W_1$	$X_{13} \times W_1$
8	$X_{12} \times W_8$	$X_{13} \times W_8$	$X_{14} \times W_8$	$X_{12} \times W_5$	$X_{13} \times W_5$	$X_{14} \times W_5$	$X_{12} \times W_2$	$X_{13} \times W_2$	$X_{14} \times W_2$
9	$X_{13} \times W_9$	$X_{14} \times W_9$	$X_{15} \times W_9$	$X_{13} \times W_6$	$X_{14} \times W_6$	$X_{15} \times W_6$	$X_{13} \times W_3$	$X_{14} \times W_3$	$X_{15} \times W_3$

intensive. So the need for computation less, low latency with less computation power with better accuracy is the demand for image recognition and classification with the least Misclassification Rate (MSR). This work strives for implementing a hardware architecture that could act as a CNN accelerator for massive computing which best suits COVID-19 diagnosis. The higher accuracy of prediction with less hardware complexity makes this system most sutable for sustainable smart city building.

3.5. Proposed hardware implementation for covid-19 diagnosis

In this section, different ensemble methods are deployed which can aid to identify COVID using various deep learning models. First, the Chest X-Ray (CRX) images are preprocessed by restructuring the images using fuzzy color techniques, and then images are stacked to structure it with the original images. The structured images are classified using various deep learning methods such as MobileNET, EfficientNET, FITNET, IRCNN, and ResNet. The output of these classifiers is ensembled using majority voting, simple averaging, and weighted averaging method to detect COVID abnormal cases from X-ray images. The general design of the proposed model is shown in Fig. 1.

3.6. Proposed data flow for convolutional computations

The proposed hardware implementation focus on developing an efficient computatational processing element that could exploit the data/signal statics and correlation among them and thereby reduce the computations. The ideas of pipelining and parallel processing which could explore the hardware resource utilization are also being considered to reduce the power consumption for complex computation and the latency of the FC neural network, which is being the basic requirement of COVID-19 diagnosis.

Convolutions by the same hardware as the FC layer can be computed by assigning one neuron to one output pixel in the output vector. As with the case of semi-parallel FC layer implementation, the number of parallel neurons is equal to the number of output pixels ($Hout \times Wout = 9$). Also, the input pixels are broadcasted to all the neurons while weights are stacked in by each neuron register. Each input pixel is processed in a way similar to the computation performed in the FC layer. Figure shown in Appendix B represents the basic 2D CNN computations required for generating a 3×3 output matrix. It is observed that a set of 9 neurons has been used to compute all of the output pixels. The convolution of the first row of the filter(i.e. W_1, W_2, and W_3) with the first row of the input pixels (i.e. X_1, X_2, X_3, X_4, and X_5) requires 5 clock cycles when $N = 3$ and $W_f = 3$. Therefore, $Hf \times (N + Wf - 1)$ clock cycles are required for a convolution of a row of the filter with its corresponding inputs. This clearly shows the requirement of 25 clock cycles to compute the output vector. It is also possible to increase the Utilization Factor (UF) and thereby a considerable increase in the latency of computation.

It has been arrived at till now that using a fully parallel implementation of the proposed data flow yields unacceptably low UF as neurons for the large proportion of clock cycles are idle. The fully parallel implementation also takes up a lot of silicon area and power consumption. Hence, the data flow diagram for CNN is analyzed carefully in the view of optimizing it for low latency, less computation, and less clock cycle. In this view the whole computation can be done using 9 neurons in the output layers with 15 clock cycles, this could be done by placing the input pixels on an ON—CHIP memory and the weights are generated using an Linear Feedback Shift Register (LFSR). The computations are rescheduled such that X_{21} to X_{25} are computed in the 1to 5 clock cycle with neurons 7 to 9 because they perform '0'computations during this period. Similarly, X_{16} to X_{20} has also been rescheduled to neuron 4 to 6 and input X_{16} to X_{20} has also been rescheduled to neuron 7 to 9 in clock cycles 6 to 10. This data rescheduling operation can be done because the data are uncorrelated and thereby a parallel architecture is designed in the hardware. This rescheduling has reduced the clock cycle by 40%. In general, the total latency of this approach is calculated using $Hf \times (N + Wf - 1) \times Cin \times Cout \times Hout \times Wout / N$. The detailed data flow representation is shown in Tables 1 and 2.

Further deep investigating of the data flow table shown in Table C.1 of Appendix-C indicates that there is much worthless computation that could be optimized which is shown in green and blue shades. Reusing of neurons in the computation window will further optimize the clock cycle better so with this the CNN computation are rescheduled as shown in Table 1. This requires only a 9 clock cycle, provided the input is to be stored in on-chip memory.

Table 2
Performance Metrics of Fine-tuned second-stage pre-trained models for COVID-19 detection.

Models	Technique	Acc	S	SP	P	F_1	MCC	K	DOR	AUC
IRCNN	Baseline	0.847	0.833	0.867	0.842	0.847	0.694	0.694	30.79	0.928 (0.886, 0.970)
	Fine-tuned	0.854	0.902	0.888	0.884	0.865	0.736	0.736	44.4	0.917 (0.872, 0.962)
Mobile Net	Baseline	0.854	0.902	0.875	0.839	0.855	0.698	0.666	35.31	0.932 (0.891, 0.95
	Fine-tuned	0.875	0.902	0.819	0.833	0.866	0.724	0.7222	42.17	0.904 (0.856, 0.952)
FITNET	Baseline	0.868	0.847	0.888	0.847	0.844	0.736	0.736	44.4	0.921 (0.877, 0.965)
	Fine-tuned	0.875	0.902	0.902	0.895	0.863	0.737	0.736	46.47	0.930 (0.888, 0.971)
ResNet-18	Baseline	0.833	0.916	0.847	0.884	0.865	0.714	0.708	41.83	0.930 (0.888, 0.971)
	Fine-tuned	0.895	0.861	0.902	0.897	0.878	0.751	0.752	51.54	0.981 (0.864, 0.957)
Efficient Net	Baseline	0.847	0.847	0.791	0.814	0.862	0.673	0.694	30.06	0.915 (0.868, 0.960)
	Fine-tuned	0.868	0.847	0.930	0.930	0.892	0.793	0.791	83.2	0.947 (0.913, 0.985)

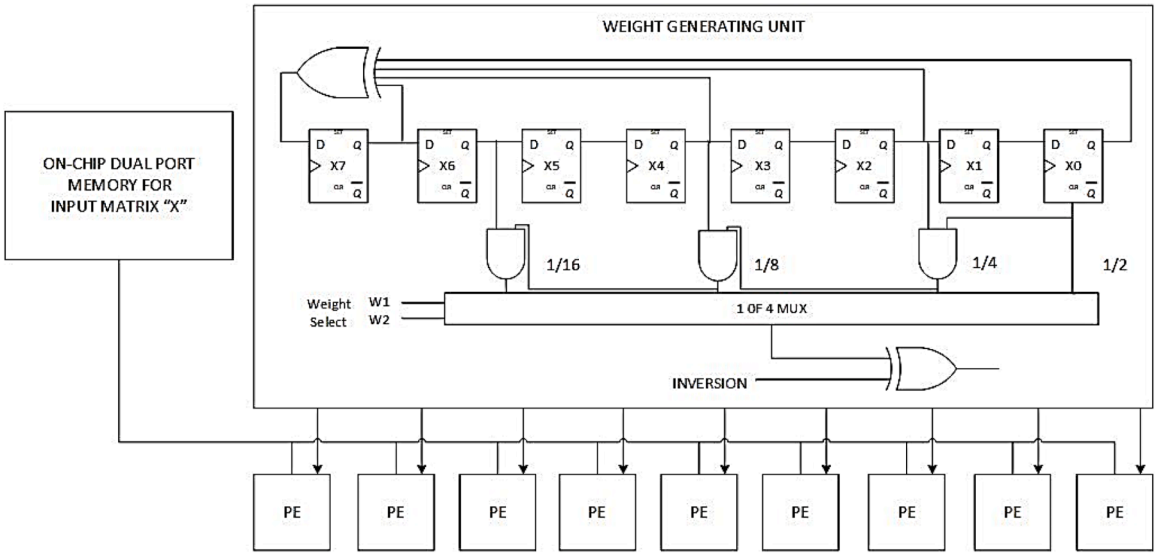


Fig. 5. Overall architecture for proposed CNN computations.

Weights are generated by the weighted LFSR to reduce memory accesses, in the case of convolutional implementations. It is passed on to all neurons as shown in Fig. 5. Passing weights from a neuron output vector pixel of one row to a neuron output that of another row requires $(S-1)$ *x Hin x Hf* delay elements which is being implemented using the D Flipflop.

3.7. Generalizing the proposed data flow

The utilization factor may be precalculated for the setup by the expression

$$UF = (Hf \times Wf)/(Hin \times Win) \times 100 \tag{4}$$

UF can be improved by reusing a subset of neurons populating in the same row of the output neuron $(N <= Wout)$ to process for all the output activation pixels. This set of neurons is referred to as a 1-D tile. As can be seen from Table 1 a convolving row of a filter map with its corresponding input pixels requires $N + Wf - - 1$ clock cycles. Now, the enhanced UF is expressed as given in Eq. (5) in comparison with Eq. (4),

$$UF = Wf/(N + Wf - 1) \times 100. \tag{5}$$

Where N indicates several re-useable neurons in the same output computation row. Despite the increase in UF and the use of a fewer number of neurons, the number of memory accesses of filter weights increases considerably. These issues can be compensated by using 'p' parallel 1D tiles to compute '*p*' out of the Cout vector in parallel. Such parallelism allows for a reduction of latency and memory accesses by a factor of p. The input pixels are broadcasted to all the 'p' 1D tiles thus improving the latency 'p' times and also easing out the bandwidth requirement of the data buses through broadcasting. The number of clock cycles to compute the convolutional layer can be expressed as given in Eq. (6):The number of

$$CCs = 3 \times (N+2) \times Cin \times Cout \times (Wout \times Win)/(N \times p). \tag{6}$$

where P is the total number of output pixels to be computed.

As can be seen in Table 2, an input pixel is read at each clock cycle while 3 filter weights are read every $(N + 2)$ clock cycle. Therefore, the number of memory accesses (frequency of access) needed for input and filter weights are as follows:

$$MA\ input\_pixels = 3 \times (N+2) \times Cin \times Cout \times (Wout \times Win)/N. \tag{7}$$

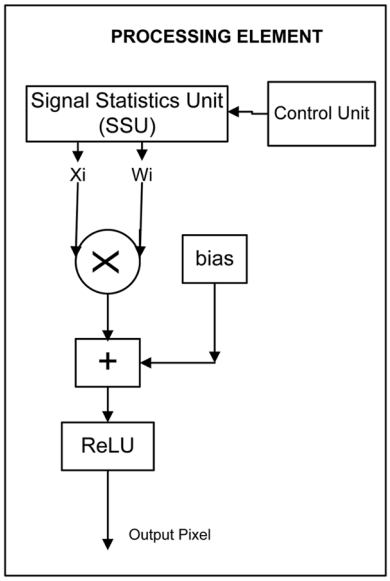


Fig. 6. Proposed Processing Element architecture.

$$MAfilter\_weights = 3 \times 3 \times Cin \times Cout \times (Wout \times Win)/(N \times p). \tag{8}$$

Looking at the expressions (7) and (8), it is inferred that altering N does not alter MAimaps but MAfilters increase as N is decreased.

Proposed processing Element architecture receives a broadcasted input pixel and a filter weight from the weight generator and performs their multiplication and then accumulates it with the corresponding value in the registers holding the psum and this process repeats till the end as shown in Fig. 6. After the computations for each pixel, ReLU is applied and the output pixels are stored in the off-chip memory. The weight generator is responsible to provide each neuron the appropriate weight is shown in Fig. 5. The proposed CNN architecture makes the computation as data aware and thereby it incorporates the smartness in the computation and thereby reduces the power consumption and with increased frequency of operation. This feature helps to design the smart city by preserving the green environment

Table 3
Top-1, top-2, and top-4 fine-tuned Ensemble models performance for COVID-19 identification.

Ensemble method	Top-N models	Accuracy	Sensitivity	Specificity	Precision	F_1	MCC	Kappa	DOR	AUC
Majority voting	1	0.932	0.961	0.926	0.945	0.958	0.938	0.955	102.22	0.949 (0.962, 0.956)
	2	0.941	0.9612	0.945	0.9586	0.979	0.764	0.763	57.63	0.961 (0.829, 0.934)
	4	0.948	0.955	0.932	0.94	0.967	0.928	0.977	65.02	0.958 (0.837, 0.940)
Simple averaging	1	0.955	0.968	0.972	0.951	0.945	0.937	0.971	74.32	0.938 (0.972, 0.984)
	2	0.931	0.961	0.952	0.948	0.979	0.964	0.963	57.63	0.946 (0.967,9831)
	4	0.961	0.975	0.968	0.977	0.981	0.964	0.963	56.01	0.955 (0.908, 0.982)
Weighted averaging	1	0.999	0.992	0.984	0.989	0.989	0.989	0.989	105.6	0.987 (0.981, 0.984)
	2	0.972	0.975	0.980	0.976	0.9	0.906	0.985	93.87	0.949 (0.953, 0.985)
	4	0.988	0.988	0.988	0.988	0.988	0.977	0.977	64.02	0.945 (0.958, 0.982)

4. Experimental analysis

The experiments are performed on the windows system with Intel Xeon CPU E3–1275, v6 3.80 GHz processor, and NVIDIA GeForce 1050 T_i all experiments are performed. Tensorflow backend uses Keras DL framework. To accelerate the performance of GPU CUDA and CUDNN libraries were used. Numerous stages of learning is performed in the proposed CNN-based deep learning models and were trained in this study: (i) ResNet-18 ii) Mobile Net-V2 iii) FitNet iv) IRCNN and v) EfficientNet. In ensemble learning, the models are chosen with a knowledge of growing the representation power, architectural diversity, when integrated and used.

An application-specific hardware architecture which incorporates the pipeline, parallel processing, reusability of computational resources by carefully exploiting the data flow and resource availability. The processing element (PE) and the CNN architecture are modeled using Verilog, simulated, and synthesized using 90 nm tech file. The simulated results show a 40% reduction in the latency and number of clock cycles. The computations and power consumptions are minimized by designing the PE as a data-aware unit.

4.1. Dataset

The images are collected from the COVID-19 Radiography Database (Melchiorre et al., 2013). A research team from various countries has created a database of chest X-ray images for COVID-19 positive cases along with images of Normal and Viral Pneumonia. This COVID-19, normal, and other lung infection dataset is released in various phases. In the first phase, 219 COVID-19, 1341 normal, and 1345 viral pneumonia chest x-ray images are released. In the second phase, the COVID-19 class images are increased to 1200. In the third phase, there are 3616 COVID-19 positive cases along with 10,192 normal images. In addition, there are 1345 viral Pneumonia images. All images are in Portable Network Graphics (PNG) file format with a resolution of 299×299 pixels. A stepwise training approach is initially performed. Pre-trained models like ImageNet and custom CNN are retrained with a large collection of images. The features of normal and infected lung images help to train the model (Togacar, Ergen, & Comert, 2020). Here 90% of datasets are divided for training and 10% for testing during the training phase. From the training dataset, 10% is randomly allocated for validation. A stratified K-fold cross-validation with $K = 5$ is performed.

4.2. Deep learning models parameter settings

Table C.2 of Appendix-C shows the optimization of hyperparameters for the single and ensemble deep learners. Adam optimizer is chosen for all deep learners. The batch size, max. epoch, global learning rate, validation frequency, drop out rate and learn rate factor is initialized after an optimal 5-fold cross-validation accuracy is obtained for the models. The classification layer weight vector for the input, hidden and output layers are also obtained based on the optimal cross-validation accuracy.

It is important to evaluate the performance of the classifiers using various metrics (Zhou et al., 2021) such as accuracy(Acc), sensitivity (S), specificity (SP), precision (P), F-score, Matthews correlation coefficient (MCC), Diagnostic Odds Ratio (DOR), Kappa (K), and Area under curve (AUC).

Table 4
Ensemble model classification results on chest x-rays.

Dataset		Normal	Pneumonia	COVID-19
Balanced dataset	Precision	0.982	0.976	0.984
	Recall	0.977	0.988	0.975
	F1	0.965	0.972	0.985
Imbalanced dataset	Precision	0.906	0.864	0.877
	Recall	0.897	0.853	0.881
	F1	0.902	0.858	0.879

5. Results and discussion

The performance of the different models are analyzed individually in ther first stage and second stage of CXR specific training. It is observed that only IRCNN and MobileNet perform better without fine-tuning. Therefore, all pre-trained models are fine tuned iteratively with their model parameters for increasing the performance as shown in the results.

The performance of the single learner models is improved by fine-tuning the models deployed in the ensemble approaches for COVID-19 identification: (i) Majority Voting; (ii) weighted and (iii) Simple averaging. The results are shown in Table 3. There is no considerable difference statistically in the AUC results ($P > 0.05$) of the ensemble model. The top-1 weighted averaging method performs better than Top-2 and Top-4 methods based on DOR, AUC, accuracy, F1 score, MCC, specificity, precision, and Kappa when compared to other models. SLSQP

Table 5
Fine-tuned ensemble model on mean squared error (MSE) cross-validation replicates.

Model	R^2	R.S.S	d.f	F-Value	P-Value
M_{MSE1}	0.3535	7.9759	2147	41.73	4.442×10^{-15}
M_{MSE2}	0.1904	9.8519	4145	9.759	5.107×10^{-7}
M_{MSE3}	0.5564	5.3238	6143	32.14	2.2×10^{-6}
M_{MSE4}	0.9931	0.0778	14,135	1540	2.2×10^{-6}

*R^2= Percentage of variation in a response variable, *R.S.*S*= Residual Sum of Squares, *d.*f*=degrees of freedom. The table shows the individual model ANOVA on Mean Squared Error (MSE) cross-validation replicates. The first model M_{MSE1} (Majority voting model) contains highly significant evidence for the variance in MSE influenced by the choice of the learning approach. The second model M_{MSE2} (Simple Averaging model) contains evidence for significant contribution to the variance in MSE by choice of attribute mapping approach. The third model contains both learning techniques and mapping approaches, but without interactions between techniques and attributes M_{MSE3} (Weighted Averaging model), contained a significantly better fit to either M_{R1} or M_{R2} model that contained only learning approaches or mapping methods. Finally, the model M_{R4}, which contained interaction terms between techniques and methods, had a marginally significantly better fit than the model M_{R3}.

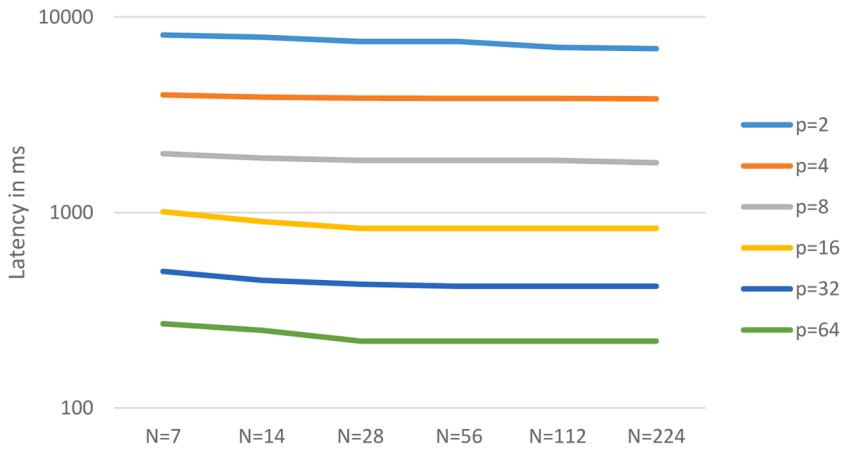


Fig. 7. Latency dependency on N and p.

technique is used to iterate the minimization and the optimal weights are converged for model predictions.

Table 4 shows the classification results of various infection types like Normal, Pneumonia, and COVID-19. Performance measures like precision, recall, and F1-score are compared among the balanced and Imbalanced datasets. The dataset initially had a huge imbalance with 10,192 normal images, 3616 COVID-19 images and 1345 Pneumonia images. The models trained on imbalanced data can cause wrong predictions during inference time due to overfitting which are evident in Table 6 results. Imbalance Ratio (IR) is a statistical metric calculated which is the ratio of majority to minority samples. A low IR specifies the minimum difference between the class labels and those samples will be undersampled.

A random undersampling is done on the majority class label in the preprocessing phase to overcome the imbalance in class distribution. In addition, a class weighting mechanism is implemented to penalize the model whenever a positive sample is misclassified.

Anova test is performed on top-1, top-2 and top-4 fine-tuned ensemble models of our proposed model along with the CNN model FitNet which has resulted in better performance in existing models. The results are shown in Table 5.

Fig. 7 shows that the latency depends on the N (reused Neurons in the same row) and P (output neuron).

All the values of latency and Memory access are for a parallelism factor $p = 1$. Table C.3 shown in appendix-c shows the variation of latency and memory access with the variation of N. This work developed a novel dataflow to accelerate deep convolutional neural networks which have better performance compared to the devised architectures in the recent 5–6 years. The techniques exploit inherent data reuse and repetitions in the processing of convolutions and FC layers. Also, algorithms such as deep compression can be deployed in conjunction with any accelerator to further expedite the processing. With the increasing use of Artificial Intelligence in recent times, it is necessary to devise energy-efficient and robust ASIC implementations that allow deploying such robust systems in battery-operated and real-time applications. These features best suit the hardware developed for the COVID-19 prediction chipset.

There are few perceptions to be analyzed in the studies such as (i) the information size and variation used in training; (ii) different deep learning designs learning capacity notifying their determination; (iii) modifying the models for improved execution, and the (iv) advantages of learning in a group. Ensemble models enhanced qualitative and quantitative performance in the identification of COVID-19 samples. Also, the predictions of the majority models are combined to ignore the mislabeling of individual models and reducing the training data prediction variance. It is evident from the results that the top-1 fine-tuned weighted averaging ensemble model increased the performance in comparison to other models. The results show that the detection is enhanced because of the ensemble of CXR-specific repeated pre-training for fine-tuning the models.

Table 6
Comparison of proposed model with state-of-the-art deep learning models.

Models	Accuracy(in %)	Precision(in %)	Recall(in %)	F1(in %)
Alexnet (Younis, 2021)	76	75	60	67
ResNet-50 (Younis, 2021)	88	71	86	77
VGG-2 (Younis, 2021)	89	88	87	87
LeNet-5 (Younis, 2021)	88	88	87	86
VGG-1 (Younis, 2021)	84	83	83	83
VGG-3 (Younis, 2021)	81	80	81	80
Inception V3 (Younis, 2021)	94	91	91	95
IRCNN	85	87	84	87
MobileNet	87	89	88	87
FitNet	87	88	90	86
ResNet-18	89	89	93	84
Deep-LSTM (Akbarian et al., 2020) ensemble	94	97.59	95	96.78
CSEN (Yamac et al., 2021)	95	90	93	89
RVM-L (Zhu et al., 2021)	96	95	96	96
SSDMNV2[31]	92.64	93	93	93
Proposed Model	99	98	98	98

Table 7
Computation time of the ensemble models.

Ensemble Method	Top-N models	Computation Time (in Seconds)
Majority Voting	1	5.64
	2	8.4
	4	9.7
Simple Averaging	1	4.54
	2	4.94
	4	6.75
Weighted Averaging	1	3.72
	2	5.62
	4	8.78

The proposed model is compared with prior deep learning models in Table 6 for COVID-19 prediction. CNN models like one block VGG, two-block VGG, three-block VGG, four-block VGG, LetNet-5, AlexNet, and Resnet-50 for identifying the COVID and SARS_MERS (Zhu et al., 2021). As per their results, LSTM approach achieved better results of 99% accuracy. MobileNet and InceptionV3 architectures were used in this proposed work and it produces better classification results with an accuracy of 96.49% respectively (Yamac et al., 2021). Based on LSTM, authors designed a nested ensemble model using deep learning methods and they proposed the Deep-LSTM ensemble model and achieved an

Table 8
Hardware resources required for CNN architecture implementation.

Sl. No	Parameters	Existing Architecture		Proposed Architecture
		AlexNet	VGG-16	VGG-16
1	Technology	45nm	45nm	45nm
2	Gate Count (NAND-2)	1852k	565k	485K
3	#MAC	168	192	178
4	Supply voltage (Volts)	1v	1v	1v
5	Power(mW)	278	236	196
6	Total Latency(ms)	115.3	4309.5	2678.3
7	Throughput(fps)	34.7	26.8	43.2
8	No. of clock cycles required	25	15	9
9	Performance (Gops)	46.1	21.4	70.3
10	Performance Efficiency	55%	26%	93%

accuracy of 97.59% (Bhattacharya et al., 2021). Convolution Support Estimation Network (CSEN) based classification has been proposed in this work for feature extraction with the deep NN solution for X-ray images and achieved accuracy of 92.64% and precision, recall and F1 score of 93% respectively (Nagrath et al., 2021).

Table 7 shows the computation time of the various ensemble models in the optimized CNN models. The weighted averaging ensemble results in minimum computational time. The limitations of this analysis are: (i) the freely accessible COVID-19 information dataset is insignificant and may not envelop a wide scope of sickness design fluctuation. (i) reduced the number of dataset samples. To overcome this problem, joint datasets can be used for the integration; (ii) better generalization capabilities of the deep learner ensemble have not been analyzed due to the limitation in the samples. (ii) regular convolutional parts are deployed for the examination, however, unique convolutional bits can minimize feature dimensionality resulting in improved execution, decreased memory, and prerequisites for computation; finally, (iv) Ensemble models involve notably high time, computational resources, and memory for effective implementation. Conversely, recent developments in registering provisions, storage, and cloud innovation will prove to be worthy in the future. The memory access and latency of the CNN hardware architecture have been reduced by 45%, this immensely supports the hardware building for COVID-19 prediction and diagnosis.

The proposed architecture has been coded using Verilog HDL, simulated using Modelsim, and synthesized used RTL synthesizer in Cadence with 45 nm technology node. The synthesized results are updated in Table 8. These results clearly show that there is around a 40% reduction in computation time in terms of clock cycles and the power consumption has been reduced by 17%

Therefore, our proposed model is a feasible solution and has shown its advantages of battling against the pandemic. Our approach produces promising results with the superiority of adaptive learning, contributing to fully understanding the current situations and predicting future trends about COVID-19.It is worth noting that the ensemble learning based model has shown outstanding performance in determining the positive number of COVID-19 cases. The reduced latency and memory access builts a robust system with high speed and low power consumption which helps the green environment thereby upholding the SDG internationally. The higher accuracy and precision of the simulated results shows a robust reliable, highly tracable system building which could be a great support for a smart health care development.

6. Conclusion

COVID-19 identification is very crucial in the era of the pandemic. This work tries to come up with a novel framework using five deep

Table C.2
Optimization of hyper-parameters.

HYPERPARAMETER	Setting		Data augmentation
	ResNet, FitNet, IRCNN, EffectiveNet, and FitNet	Majority Voting Ensemble, Simple Averaging Ensemble, and Weighted Averaging Ensemble	
Optimizer	ADAM	ADAM	Both axis side
Batch Size	10	10	random
Max Epoch	200	100	reflection
Global Learning Rate	4	4	Rescaling
Dropout rate	0.5	0.8	randomly b/w
Validation Frequency	68	68	[0. 5 *to* 1.50]
Learn Rate Factor	10	10	Rotating
classification layer weight vector	[0.75 0.15 1.18]	[0.75 0.15 1.18]	randomly b/w [−40° 40°]

Table C.3
Dependence on N for latency and memory accesses.

Value of N	Processing Latency (ms)	$MA_{filters}$ MB)	MA_{input} pixelMB)
7	16,334.6	4384.7	13,154
14	14,615.8	2192.2	11,692.6
28	13,784.7	1096.1	11,027.8
56	13,507.1	548.2	10,805.8
112	13,436.9	273.9	10,749.5
224	13,422.6	138.8	10,738

Table C.1
Initial stage of proposed dataflow for convolutional computations.

The first row of the output				The second row of the output			A third row of the output		
Clock cycle	Neuron #1	Neuron #2	Neuron #3	Neuron#4	Neuron#5	Neuron#6	Neuron#7	Neuron#8	Neuron#9
1	$X_1 \times W_1$	$X_1 \times 0$	$X_1 \times 0$	$X_{16} \times W_7$	$X_{16} \times 0$	$X_{16} \times 0$	$X_{21} \times W_7$	$X_{21} \times 0$	$X_{21} \times 0$
2	$X_2 \times W_2$	$X_2 \times W_1$	$X_2 \times 0$	$X_{17} \times W_8$	$X_{17} \times W_7$	$X_{17} \times 0$	$X_{22} \times W_8$	$X_{22} \times W_7$	$X_{22} \times 0$
3	$X_3 \times W_3$	$X_3 \times W_2$	$X_3 \times W_1$	$X_{18} \times W_9$	$X_{18} \times W_8$	$X_{18} \times W_7$	$X_{23} \times W_9$	$X_{23} \times W_8$	$X_{23} \times W_7$
4	$X_4 \times 0$	$X_4 \times W_3$	$X_4 \times W_2$	$X_{19} \times 0$	$X_{19} \times W_9$	$X_{19} \times W_8$	$X_{24} \times 0$	$X_{24} \times W_9$	$X_{24} \times W_8$
5	$X_5 \times 0$	$X_5 \times 0$	$X_5 \times W_3$	$X_{20} \times 0$	$X_{20} \times 0$	$X_{20} \times W_9$	$X_{25} \times 0$	$X_{25} \times 0$	$X_{25} \times W_9$
6	$X_6 \times W_4$	$X_6 \times 0$	$X_6 \times 0$	$X_6 \times W_1$	$X_6 \times 0$	$X_6 \times 0$	$X_{16} \times W_4$	$X_{16} \times 0$	$X_{16} \times 0$
7	$X_7 \times W_5$	$X_7 \times W_4$	$X_7 \times 0$	$X_7 \times W_2$	$X_7 \times W_1$	$X_7 \times 0$	$X_{17} \times W_5$	$X_{17} \times W_4$	$X_{17} \times 0$
8	$X_8 \times W_6$	$X_8 \times W_5$	$X_8 \times W_4$	$X_8 \times W_3$	$X_8 \times W_2$	$X_8 \times W_1$	$X_{18} \times W_6$	$X_{18} \times W_5$	$X_{18} \times W_4$
9	$X_9 \times 0$	$X_9 \times W_6$	$X_9 \times W_5$	$X_9 \times 0$	$X_9 \times W_3$	$X_9 \times W_2$	$X_{19} \times 0$	$X_{19} \times W_6$	$X_{19} \times W_5$
10	$X_{10} \times 0$	$X_{10} \times 0$	$X_{10} \times W_6$	$X_{10} \times 0$	$X_{10} \times 0$	$X_{10} \times W_3$	$X_{20} \times 0$	$X_{20} \times 0$	$X_{20} \times W_6$
11	$X_{11} \times W_7$	$X_{11} \times 0$	$X_{11} \times 0$	$X_{11} \times W_4$	$X_{11} \times 0$	$X_{11} \times 0$	$X_{11} \times W_1$	$X_{11} \times 0$	$X_{11} \times 0$
12	$X_{12} \times W_8$	$X_{12} \times W_7$	$X_{12} \times 0$	$X_{12} \times W_5$	$X_{12} \times W_4$	$X_{12} \times 0$	$X_{12} \times W_2$	$X_{12} \times W_1$	$X_{12} \times 0$
13	$X_{13} \times W_9$	$X_{13} \times W_8$	$X_{13} \times W_7$	$X_{13} \times W_6$	$X_{13} \times W_5$	$X_{13} \times W_4$	$X_{13} \times W_3$	$X_{13} \times W_2$	$X_{13} \times W_1$
14	$X_{14} \times 0$	$X_{14} \times W_9$	$X_{14} \times W_8$	$X_{14} \times 0$	$X_{14} \times W_6$	$X_{14} \times W_5$	$X_{14} \times 0$	$X_{14} \times W_3$	$X_{14} \times W_2$
15	$X_{15} \times 0$	$X_{15} \times 0$	$X_{15} \times W_9$	$X_{15} \times 0$	$X_{15} \times 0$	$X_{15} \times W_6$	$X_{15} \times 0$	$X_{15} \times 0$	$X_{15} \times W_3$

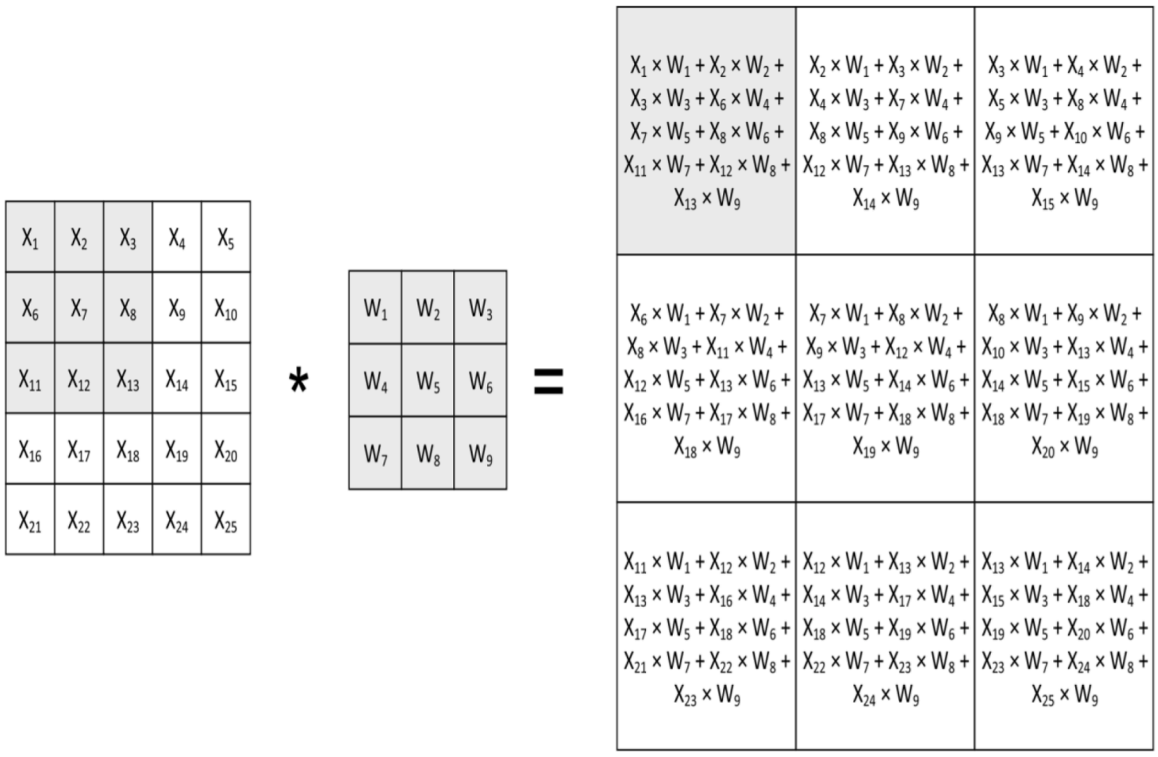


Fig. B.1. Basic Convolutional Operation.

learning models namely ResNet, FitNet, IRCNN, EffectiveNet, and Fitnet. They are pretrained individually using a recurrent CXR specific approach. The models are fine-tuned with the initialization parameters. Each of these models are integrated using various ensemble approaches like majority voting, simple averaging and weighted averaging. It is observed that weighted averaging ensemble results in maximum accuracy, precision, recall and F1-score of 0.99, 0.98,0.98 and 0.98 respectively with 64% reduction in clock cycles. The hardware architecture developed is made as a dedicated chipset which minimizes the computation time, energy, latency and improves the performance efficiency by 93% in comparison to state-of-the-art techniques. As future work, the data collected from different public automated health centres can be stored in a secured cloud environment which helps in extracting information in the national and international level. The individual identified as COVID positive can be tracked, guided, treated using IOT/mobile network solution. The noticeable progress of healthcare services and technologies, named Smart Healthcare, have a direct contribution with the improvement of smart cities in general. Thus, the proposed model can scientifically reduce the infection rate in a smart sustainable healthy city environment.

Declaration of Competing Interest

None.

Funding/Acknowledgement

Waleed Al-Numay acknowledges financial support from the Researchers Supporting Project No. (RSP-2021/250), King Saud University, Riyadh, Saudi Arabia.

Appendix:A: Ensemble Models

Let N_1 be the number of output channels, $D_1K \times D_1K$ be the convolution kernel size K_1, and M_1 specifies the input channels. The parameters of a depthwise convolution are given in equation (A.1):

$$D_1K.D_1\ K(\alpha M_1.\delta) \tag{A.1}$$

and the parameters of a pointwise convolution is given in equation (A.2):

$$(\alpha M_1.\delta)\cdot\alpha N_1 \tag{A.2}$$

Fully Connected Network

The computation of an FC layer can be done as follows in equation (A.3):

$$Y = ReLU(w < \text{ce:inf} > \text{m} < /\text{ce:inf} > \times n \times x < \text{ce:inf} > \text{n} < /\text{ce:inf} > \times 1 + b_m \times 1), \tag{A.3}$$

where x represents the inputs, y the output,w the weights, and b the biases, while ReLU is the non-linear activation function given in equation (A.4):

$$ReLU(x) = max(0, x) \tag{A.4}$$

Appendix:B

Fig. B.1 illustrates a simple example of a 2D convolution. Each layer performs the operations as represented in the equations (B.1), (B.2), and (B.3).

$$Y(z,t,q) = B(q) + \sum_{k=1}^{Cin}\sum_{j=1}^{Hf}\sum_{i=1}^{Wf} X(zS+j, tS+i, k) \times W(j,i,k,q) \tag{B.1}$$

$$H_{out} = (H < \text{ce:inf} > \text{in} < /\text{ce:inf} > - H < \text{ce:inf} > \text{f} < /\text{ce:inf} > + S)/S, \tag{B.2}$$

$$W < \text{ce:inf} > \text{out} < /\text{ce:inf} > = (W < \text{ce:inf} > \text{in} < /\text{ce:inf} > - W < \text{ce:inf} > \text{f} < /\text{ce:inf} > + S)/S \tag{B.3}$$

Where *B, W, Y,* and *X* denote the bias, the weight matrix, the output map, and the input map respectively. Also, $1 \le z \le H_{out}$, $1 \le t \le W_{out}$ *and* $1 \le q \le C_{out}$. Stride S is the number of pixels of the input activation maps by which the filter hops after each convolution.

Appendix:C

Appendix:D

Abbreviations	Descriptions
AI	Artificial Intelligence
AUC	Area Under Curve
BCNN	Bayesian Convolutional Neural Networks
CNN	Convolutional Neural Network
CSEN	Convolution Support Estimation Network
CRM	Class-specific Relevance Mapping
CT	Computed Tomography
CXRs	Chest X-rays
CADx	Computer-Aided Diagnostic devices
DL	Deep Learning
DM	Diabetes Mellitus
DOR	Diagnostic Odds Ratio
FITNET	Function fitting Neural Network
FC	Fully Connected Network
HDs	Heart Disorders
HTN	Hyper Tension
HCLS	Hypercholesterolemia
IRCNN	Inception Recurrent Convolutional Neural Network
IoT	Internet of Things
K-NN	K-Nearest Neighbor
LSTM	Long Short-Term Memory
LRP	Layer-wise Relevance Propagation
LFSR	Linear Feedback Shift Register
MCR	Miss-Classification Rates
MCC	Matthews Correlation Coefficient
PE	Processing Element
ROI	Region-Of-Interest
RT-PCR	Reverse Transcription-Polymerase Chain Reaction
RCL	Recurrent Convolution Layer
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
SLSQP	Sequential Least- Squares Programming Method
TESM	Truth Estimate from Self Distances Method
TESD	Truth Estimate from Self Distances
UF	Utilization Factor

References

Ahmad, F., Farooq, A., & Ghani, M.U. (.2021). Deep ensemble model for classification of novel coronavirus in chest X-ray images. Computational intelligence and neuroscience, 2021. Article 8890226. 10.1155/2021/8890226.

Akbarian, S., Seyyed-Kalantari, L., Khalvati, F., & Dolatabadi, E. (2020). Evaluating knowledge transfer in neural network for medical images. 1-12. arXiv preprint arXiv: 2008.13574..

Ali, F., El-Sappagh, S., Islam, S. R., Kwak, D., Ali, A., Imran, M., et al. (2020). A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion, 63*(2), 208–222. https://doi.org/10.1016/j.inffus.2020.06.008

Ardakani, A., Condo, C., & Gross, W.J. (2016). Sparsely-connected neural networks: Towards efficient vlsi implementation of deep neural networks,3,1–14. arXiv preprint arXiv:1611.01427.

Bhattacharya, S., Maddikunta, P. K. R., Pham, Q. V., Gadekallu, T. R., Chowdhary, C. L., Alazab, M., et al. (2021). Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. *Sustainable Cities and Society, 65*, 1–18. https://doi.org/10.1016/j.scs.2020.102589

Chen, H. Y., & Su, C. Y. (2018). An enhanced hybrid MobileNet. In *Proceedings of the 9th International Conference on Awareness Science and Technology (iCAST)* (pp. 308–312). https://doi.org/10.1109/icawst.2018.8517177. IEEE.

Chen, Y. H., Krishna, T., Emer, J. S., & Sze, V. (2016). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits, 52*(1), 127–138. https://doi.org/10.1109/jssc.2016.2616357

Chowdhury, N.K., Kabir, M.A., Rahman, M., & Rezoana, N. (2020). ECOVNet: An ensemble of deep convolutional neural networks based on efficientnet to detect COVID-19 from Chest X-rays, 2, 1–21. arXiv preprint arXiv:2009.11850. 10.7717/peerj-cs.551.

Das, A. K., Ghosh, S., Thunder, S., Dutta, R., Agarwal, S., & Chakrabarti, A. (2021). Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network. *Pattern Analysis and Applications, 24*, 1111–1124. https://doi.org/10.1007/s10044-021-00970-4

Elgendi, M., Fletcher, R., Howard, N., Menon, C., & Ward, R. (2020). The evaluation of deep neural networks and x-ray as a practical alternative for diagnosis and management of covid-19,1-7. medRxiv. 10.1101/2020.05.12.20099481.

Ghoshal, B., & Tucker, A. (2020). Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection, 2, 1–14. arXiv preprint arXiv: 2003.10769..

Haghanifar, A., Majdabadi, M.M., Choi, Y., Deivalakshmi, S., & Ko, S. (2020). Covid-cxnet: Detecting covid-19 in frontal chest x-ray images using deep learning, 2, 1–21. arXiv preprint arXiv:2006.13807.

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., et al. (2016). EIE: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News, 44*(3), 243–254. https://doi.org/10.1145/3007787.3001163

Han, S., Mao, H., & Dally, W.J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,5,1–14. arXiv preprint arXiv:1510.00149.

Hussain, E., Hasan, M., Rahman, M. A., Lee, I., Tamanna, T., & Parvez, M. Z. (2021). CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images. *Chaos, Solitons & Fractals, 142*(110495), 1–12. https://doi.org/10.1016/j.chaos.2020.110495

Karim, M., Döhmen, T., Rebholz-Schuhmann, D., Decker, S., Cochez, M., & Beyan, O. (2020). DeepCOVIDExplainer: Explainable COVID-19 diagnosis based on chest X-ray images,2,1–10. arXivpreprintrXiv:2004.04582. 10.1109/bibm49941.2020.9313304.

Lane, N. D., & Georgiev, P. (2015). Can deep learning revolutionize mobile sensing?. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (pp. 117–122). https://doi.org/10.1145/2699343.2699349

Li, S., Jiao, J., Han, Y., & Weissman, T. (2016). Demystifying resnet, 2, 1–18. arXiv preprint arXiv:1611.01186..

Lopez-Paz, D., Bottou, L., Schölkopf, B., & Vapnik, V. (2015). Unifying distillation and privileged information,3,1–10. arXiv preprint arXiv:1511.03643..

Melchiorre, M. G., Chiatti, C., Lamura, G., Torres-Gonzales, F., Stankunas, M., Lindert, J., et al. (2013). Social support, socio-economic status, health and abuse among older people in seven European countries. *PloS one, 8*(1), e54856. https://doi.org/10.1371/journal.pone.0242301

Melin, P., Monica, J. C., Sanchez, D., & Castillo, O. (2020). *Multiple ensemble neural network models with fuzzy Response Aggregation for Predicting COVID-19 Time Series: The case of mexico. In healthcare, 8* pp. 1–13). Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/healthcare8020181

Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society, 66*, Article 102692.

Otoom, M., Otoum, N., Alzubaidi, M. A., Etoom, Y., & Banihani, R. (2020). An IoT-based framework for early identification and monitoring of COVID-19 cases. *Biomedical Signal Processing and Control, 62*(102149), 1–12. https://doi.org/10.1016/j.bspc.2020.102149

Polsinelli, M., Cinque, L., & Placidi, G. (2020). A light CNN for detecting COVID-19 from CT scans of the chest. *Pattern Recognition Letters, 140*, 95–100. https://doi.org/10.1016/j.patrec.2020.10.001

Rajaraman, S., & Antani, S. (2020). Weakly labeled data augmentation for deep learning: A study on COVID-19 detection in chest X-rays. *Diagnostics, 10*(6), 1–17. https://doi.org/10.3390/diagnostics10060358

Rajaraman, S., Siegelman, J., Alderson, P. O., Folio, L. S., Folio, L. R., & Antani, S. K. (2020). Iteratively pruned deep learning ensembles for COVID-19 detection in chest X-rays. *IEEE, 8*, 115041–115050. https://doi.org/10.1109/access.2020.3003810

Rubin, G. D., Ryerson, C. J., Haramati, L. B., Sverzellati, N., Kanne, J. P., Raoof, S., et al. (2020). The role of chest imaging in patient management during the COVID-19 pandemic: A multinational consensus statement from the Fleischner society. *Radiology, 296*(1), 172–180. https://doi.org/10.1148/radiol.2020201365

Shalbaf, A., & Vafaeezadeh, M. (2021). Automated detection of COVID-19 using ensemble of transfer learning with deep convolutional neural network based on CT scans. *International Journal of Computer Assisted Radiology and Surgery, 16*(1), 115–123. https://doi.org/10.1007/s11548-020-02286-w

Shastri, S., Singh, K., Kumar, S., Kour, P., & Mansotra, V. (2021). Deep-LSTM ensemble framework to forecast Covid-19: An insight to the global pandemic. *International Journal of Information Technology, 13*(1), 1291–1301. https://doi.org/10.1007/s41870-020-00571-0

Shin, D., Lee, J., Lee, J., & Yoo, H. J. (2017). 14.2 DNPU: An 8.1 TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)* (pp. 240–241). https://doi.org/10.23919/date.2017.7927142. IEEE.

Shoeibi, A., Khodatars, M., Alizadehsani, R., Ghassemi, N., Jafari, M., Moridian, P., Sani, Z.A. (2020). Automated detection and forecasting of covid-19 using deep learning techniques: A review, 3, 1–20.arXiv preprint arXiv:2007.10785.

Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Deep learning applications for COVID-19. *Journal of Big Data, 8*(1), 1–54. https://doi.org/10.1186/s40537-020-00392-9

Smithson, S. C., Boga, K., Ardakani, A., Meyer, B. H., & Gross, W. J. (2016). Stochastic computing can improve upon digital spiking neural networks. In *Proceedings of the IEEE International Workshop on Signal Processing Systems (SiPS)* (pp. 309–314). https://doi.org/10.1109/sips.2016.61. IEEE.

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning* (pp. 6105–6114). PMLR.

Togacar, M., Ergen, B., & Comert, Z. (2020). COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. *Computers in Biology and Medicine,, 121*. https://doi.org/10.1016/j.compbiomed.2020.103805. Article 103805.

Toraman, S., Alakus, T. B., & Turkoglu, I. (2020). Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. *Chaos, Solitons & Fractals, 140*(110–122), 1–11. https://doi.org/10.1016/j.chaos.2020.110122

Vantaggiato, E., Paladini, E., Bougourzi, F., Distante, C., Hadid, A., & Taleb-Ahmed, A. (2021). Covid-19 recognition using ensemble-cnns in two new chest x-ray databases. *Sensors, 21*(1742), 1–20. https://doi.org/10.3390/s21051742

Vapnik, V., & Izmailov, R. (2015). Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research, 16*(1), 2023–2049. https://doi.org/10.1007/978-3-319-17091-6_1

Wang, S., Zhou, D., Han, X., & Yoshimura, T. (2017). Chain-NN: An energy-efficient 1D chain architecture for accelerating deep convolutional neural networks. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1032–1037). https://doi.org/10.1109/sips.2016.61. IEEE.

Yamac, M., Ahishali, M., Degerli, A., Kiranyaz, S., Chowdhury, M. E., & Gabbouj, M. (2021). Convolutional sparse support estimator-based COVID-19 recognition from X-Ray images. *IEEE Transactions on Neural Networks and Learning Systems, 32*(5), 1810–1820. https://doi.org/10.1109/tnnls.2021.3070467

Yan, B., Wang, J., Cheng, J., Zhou, Y., Zhang, Y., Yang, Y., et al. (2021). Experiments of federated learning for covid-19 chest x-ray images. In *Proceedings of the International Conference on Artificial Intelligence and Security* (pp. 41–53). https://doi.org/10.1007/978-3-030-78618-2_4. Springer, Cham.

Younis, M. C. (2021). Evaluation of deep learning approaches for identification of different corona-virus species and time series prediction. *Computerized Medical Imaging and Graphics, 90*. https://doi.org/10.1016/j.compmedimag.2021.101921. Article 101921.

Zhang, K., Zuo, W., Gu, S., & Zhang, L. (2017). Learning deep CNN denoiser prior for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3929–3938). https://doi.org/10.1109/cvpr.2017.300

Zhou, T., Lu, H., Yang, Z., Qiu, S., Huo, B., & Dong, Y. (2021). The ensemble deep learning model for novel COVID-19 on CT images. *Applied Soft Computing, 98* (106885), 1–9. https://doi.org/10.1016/j.asoc.2020.106885

Zhu, R., Ding, Q., Yu, M., Wang, J., & Ma, M. (2021). Early warning scheme of COVID-19 related internet public opinion based on RVM-L Model. *Sustainable Cities and Society, 74*, Article 103141.

Zivkovic, M., Bacanin, N., Venkatachalam, K., Nayyar, A., Djordjevic, A., Strumberger, I., et al. (2021). COVID-19 cases prediction by using hybrid machine learning and beetle antennae search approach. *Sustainable Cities and Society, 66*, Article 102669.