Utilization of a Deep Convolutional Neural Network for the Binary Classification of Chest X-Ray Pneumonia

Haydr Nataq Taha Al-Azzawi
Department of Computer Communication, College of Engineering, Islamic University of Lebanon, Beirut, Lebanon
haydrnatiq2@gmail.com

Ahmad Ghandour
Department of Computer Communication, College of Engineering, Islamic University of Lebanon, Beirut, Lebanon
ahmad.ghandour@iul.edu.lb

Haider Ali
Department of Cybersecurity and Cloud Computing, Technical Engineering, Uruk University, Baghdad 10001, Iraq
Haider_ali@uruk.edu.iq

Ahmad Taher Azar
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia | Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia | Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt
aazar@psu.edu.sa

Najla Althuniyan
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia | Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia
nthuniyan@psu.edu.sa

Ibraheem Kasim Ibraheem
Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad 10001, Iraq
ibraheemki@coeng.uobaghdad.edu.iq (corresponding author)

Yousif I. Hammadi
Department of Medical Instrumentation Techniques Engineering, Bilad Alrafidain University College, Diyala 32001, Iraq
yousif.ibrahim.hammadi@gmail.com

Amjad J. Humaidi
Control and Systems Engineering Department, University of Technology, Baghdad 10001, Iraq
amjad.j.humaidi@uotechnology.edu.iq

Zeeshan Haider
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia | Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia
zhaider@psu.edu.sa

Saim Ahmed
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia | Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia
sahmed@psu.edu.sa

Received: 30 November 2024 | Revised: 25 December 2024 | Accepted: 1 January 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: https://doi.org/10.48084/etasr.9788

ABSTRACT

Pneumonia remains a significant global health concern, necessitating efficient diagnostic tools. This study presents a novel Convolutional Neural Network (CNN) architecture, CuDenseNet, designed for the binary classification of Chest X-Ray (CXR) images as either having pneumonia or normal (healthy). Unlike models that rely on transfer learning from pre-trained architectures, CuDenseNet is trained from scratch and incorporates three parallel DenseNet paths of varying depths, enhancing feature extraction and classification accuracy. The model was evaluated on a combined dataset of 11,708 CXR images, achieving exceptional performance metrics of 99.1% accuracy, 99.7% precision, 99.1% recall, and an AUC of 99.7%. The comparative analysis demonstrates that CuDenseNet outperforms state-of-the-art pre-trained models such as VGG19 and ResNet50 while providing superior adaptability. These results underscore the potential of CuDenseNet as a robust and reliable tool for automated pneumonia diagnosis, with significant implications for clinical applications and future research in medical imaging.

Keywords-deep learning; convolutional neural networks; chest X-ray images; pneumonia classification; image processing; Computer-Aided Diagnosis (CAD); health informatics; AI

I. INTRODUCTION

Machine Learning (ML) and Artificial Intelligence (AI) are revolutionizing nearly every aspect of modern life, from healthcare and transportation to finance and entertainment. These technologies allow machines to process vast amounts of data, identify patterns, and make decisions with minimal human intervention. By automating routine and complex tasks, AI and ML enhance productivity, reduce errors, and enable new capabilities that were previously unattainable. For instance, AI-driven diagnostics in healthcare have improved the accuracy and speed of disease detection. ML models also power personalized recommendations in e-commerce and streaming platforms, delivering more relevant and engaging user experiences [1-4].

Deep Learning (DL), a subset of ML, plays a transformative role in healthcare by enabling advanced data analysis and decision-making capabilities [5-7]. DL models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), excel at analyzing complex and high-dimensional data like medical images, genomic sequences, and Electronic Health Records (EHRs). For example, DL algorithms have been pivotal in improving the accuracy of diagnostic tools, such as identifying tumors in radiology scans, detecting diabetic retinopathy, and predicting cardiovascular risks. These systems often achieve or exceed the performance of human experts, offering faster and more consistent evaluations, which is critical in resource-limited settings or high-pressure environments.

Pneumonia represents a significant global health challenge, being a leading cause of mortality, particularly among children under the age of five and the elderly. The diagnosis and management of pneumonia are critical for reducing its associated mortality and morbidity. In low-resource settings, the lack of accessible, cost-effective diagnostic tools exacerbates the challenge, as timely detection and intervention can significantly improve patient outcomes. Traditional diagnostic techniques for pneumonia, such as sputum culture, blood tests, and clinical examinations, often lack the speed and accuracy needed for early detection. Chest X-rays (CXRs) are widely employed for diagnosis due to their availability and cost-effectiveness. However, their interpretation relies heavily on skilled radiologists, making the process subjective and prone to errors, especially in resource-limited settings with fewer specialists. DL, particularly CNNs, has emerged as a transformative approach in medical imaging. Its capability to process complex, high-dimensional data enables automated and highly accurate diagnosis from CXR images. Unlike traditional methods, DL models can:

- Reduce diagnostic errors: They offer consistency in interpretation, minimizing human-related variability.
- Enhance speed: Automated analysis enables rapid screening, which is crucial during disease outbreaks or pandemics.
- Address resource gaps: In settings with limited access to radiologists, DL systems can provide diagnostic support, democratizing healthcare access.

Given these factors, DL is particularly well-suited to tackle the challenges of pneumonia diagnosis, offering a scalable, efficient, and reliable solution. This study's proposed CuDenseNet architecture exemplifies such potential by demonstrating superior performance metrics compared to existing methods, emphasizing its practical and clinical relevance in addressing the global burden of pneumonia.

A. Problem Statement

As an alternate diagnosis procedure for pneumonia, X-ray imaging technology has significant advantages over traditional testing techniques. These benefits include low operating costs, widespread availability of X-ray competencies, non-invasiveness, compact computing complication, and apparatus straightforwardness. X-ray images may be a better alternative for extensive, easy, low-cost, and speedy identification of a pandemic such as COVID-19, heart failures, bone fractures, etc. [8]. Computer-controlled pneumonia detection systems have grown in popularity in recent years as a tool to improve the efficacy and cogency of medical services. DL techniques go further than traditional ML approaches for picture analysis in health utilizations such as following, labeling, and organization.

Authors in [9] assessed the CNNs' efficiency in automating the classification of chest radiographs into binary data. Authors in [10] presented a unique hybrid method for identifying pulmonary disorders like pneumonia and lung nodules using modular artificial neural networks and fuzzy logic. Authors in [11] proposed and validated a mix of radionics and ML features for early and rapid COVID-19 diagnosis from CXR in the presence of viral or bacterial pneumonia and at varied sickness severity levels. In [12], Transfer Learning (TL) is used to generalize information from a large set of unlabeled Synthetic Aperture Radar (SAR) scene photos to labeled target data. Authors in [13], motivated by the fact that X-ray imaging systems are more prevalent and cheaper than CT scan systems, a DL-based CNN model, Truncated Inception Net, was proposed to screen COVID-19 positive CXRs from other non-COVID and/or healthy cases. Authors in [14] used CXR scans from people with normal bacterial pneumonia, confirmed COVID-19 infection, and normal occurrences to automatically diagnose the Coronavirus disease.

Pre-trained CNN models include VGG19 [15], MobileNet v2 [16], Inception [17], Xception [18], and Inception ResNet v2 [17]. To this end, authors in [19] present an ensemble model developed through the joint effort of three separate TL models: EfcientNet, GoogLeNet, and XceptionNet. Using DCNN-based pre-trained TL models and CXR pictures, authors in [20] presented an automatic Computer-Aided Diagnosis (CAD) prediction model consisting of CXRs from healthy individuals, COVID-19 patients, and those infected with bacterial and viral pneumonia. The authors used pre-trained models of the ResNet50, ResNet101, ResNet152 [21], InceptionV3, and InceptionResNetV2 architectures to improve prediction accuracy on three separate binary datasets. In [22], features were extracted from CXR images using the pre-trained network DenseNet169, whereas XGBoost (eXtreme Gradient Boosting) was used to categorize some of the chosen features. The proposed approach was trained and tested using the ChestX-ray8 dataset [23]. AI was studied in [24] to see if it could be helpful in the quick and precise diagnosis of pneumonia from CXRs. This study aims to offer a reliable method for automatic pneumonia detection from digital CXRs using pre-trained DL algorithms to optimize detection accuracy. MobileNetv2, SqueezeNet [25], ResNet18, ResNet101, and DenseNet201 were evaluated experimentally. Authors in [26] deployed CheXNet for pneumonia classidfication from CXR images. The work presented in [27] applies a DL method based on a pre-trained AlexNet structure for categorizing CXR scans received from various public databases.

Sequential CNNs create both phases of the architecture in [28]. Phase 1 classifies data as COVID or NON-COVID, while phase-2 classifies NON-COVID pictures as normal or having pneumonia. Two models were proposed in [29] for binary and three-class classifications. The authors recommend a CNN for COVID-19 identification in the case of binary categorization. Authos in [31] utilized five pre-trained models (InceptionV3, NASnet [30], Xception, MobileNetV2, ResNet101) integrated through TL into the EL model for Covid-19 and pneumonia classification in a system called InstaCovNet-19. Authors in [32] studied the application of DL for detecting COVID-19 based on CXR images. Authors in [33] specifically employed CNNs and ANNs. This research suggests two distinct methods, each involving two separate systems, for identifying tuberculosis in two separate data sets. Authors in [34] tried to offer a DL solution that utilizes the integration of Xception-NN and LSTM that can achieve automated detection of individuals with pneumonia in CXRs. Authors in [35] offered a DL architecture to investigate pneumonia and lung cancer. Its efficacy was further verified by comparison to other pre-trained DL methods like AlexNet [36], VGG16, VGG19, and ResNet50. Authors in [37] used data from a database containing the results of 35,038 posterior-anterior CXRs and finalization conducted on adults between 2005 and 2015. GoogLeNet-CNN was trained on 3 GPUs to reliably diagnose cardiomegaly and other conditions based on CXRs. With 19 layers in the base model and relatively small ConvL filters (3-by-3), the model proposed in [38] was developed to distinguish pneumonia, COVID-19, and healthy cases using VGG-19 as a base, and the weights were adjusted using the ImageNet setup. The CNN model in [39] uses separable convolution for all ConvLs except the first. Its separability distinguishes it from the classical convolution. Depth and point convolution include separable convolution. Feature extractors and a classifier comprise most of the suggested CNN model design in [40]. It uses four ConvLs in a sequential design with the classifier located at its end. A new hybrid approach for CXR-based pneumonia diagnosis was proposed in [41] that used an adaptive median filter and the RF (ACNN-RF).

Motivated by the shortcomings of the studies mentioned above, it was found that the training from scratch has not been fully investigated, and most of the state-of-the-art methodologies are based on pre-trained models. The proposed models are very simple, therefore they cannot be generalized due to unrealizable results. Consequently, in this work, a novel CNN structure, based on the DenseNet model, is introduced and trained from scratch. In other words, the structure is a Customized DenseNet (CuDenseNet) model. The conventional

DenseNet consists of a single path from the input to the output, while the CuDenseNet consists of three parallel DenseNet paths, each with a different depth. Hence, to introduce CuDenseNet confidently, conventional DenseNet should be introduced first.

B. Paper Contribution

This paper introduces several significant contributions to the field of medical imaging and automated pneumonia diagnosis, establishing CuDenseNet as a valuable advancement in DL for medical imaging, paving the way for further research and application in related domains. The main contributions of this paper are:

- **Novel CNN Architecture**: The study presents CuDenseNet, a custom-built CNN specifically designed for binary classification of CXR images. Unlike traditional approaches, CuDenseNet is trained from scratch, enabling greater adaptability and targeted performance.
- **Innovative Multi-Path Design**: The CuDenseNet architecture incorporates three parallel DenseNet paths of varying depths, which enhance feature extraction and improve classification accuracy compared to standard single-path networks.
- **High-Performance Metrics**: The proposed model achieved exceptional results demonstrating its effectiveness and reliability for pneumonia detection.
- **Comparison with State-of-the-Art Models**: The study provides a thorough comparative analysis, showing that CuDenseNet outperforms pre-trained models such as VGG19 and ResNet50, offering superior adaptability and efficiency.
- **Comprehensive Dataset Utilization**: The model was evaluated on a large combined dataset of 11,708 CXR images, ensuring robustness and generalizability.
- **Practical and Clinical Relevance**: The study highlights CuDenseNet's potential as a useful tool for automated pneumonia diagnosis, with implications for enhancing clinical workflows and supporting healthcare professionals in resource-limited settings.

II. THE PROPOSED MODEL

The structure of the proposed model is based on the conventional DenseNet model. The proposed model consists of three parallel paths of different depths of DenseNets, as shown in Figure 1. The first path, which is the longest, involves 96-ConvLs, the second path has 40-ConvLs, and the third path consists of 31-ConvLs. Figure 2 shows the Dense-Block, which consists of six layers, starting with BchNL followed by ReLU, ConvL, BchNL, ReLU, and ConvL, respectively. Figure 3 shows the main part from which the paths are constructed. The first ConvL in the Dense Block has a kernel size of 1-by-1, while the second ConvL is adjusted with a kernel size 3-by-3.

Figure 3 shows the construction of the first path, in which the first layer is the input layer, followed by ConvL(w_i, h_i, f_i, w_o, h_o, f_o), where w_i, h_i, and f_i are the input width, height, and number of filters, respectively, and w_o, h_o, and f_o are the output width, height, and number of filters, respectively. w_i, h_i, f_i, w_o, h_o, and f_o are 150, 150, 3, 75, 75, and 64, respectively. After that, PoL is followed, which will reduce the output FM dimensions by a factor of 2. Thus, the output FM is (w_o =38, h_o =38, f_o =64). Note that the output number of filters will not be affected. The previous operations can be considered as the input block. The following are the main model operations, which are combinations of dense blocks and transition blocks. The input dimensions to the Dense-Block is equal to the previous dimensions. However, the number of filters will summate the last two consecutive Dense-Blocks, while the outputs are: w_o = 38, h_o = 38, f_o = 32). The previous procedure continues to span a total of 14 Dense-Blocks. Then, the first transition block is reached (see Figure 4). The Transition-Block input layer is BchNL, followed by ReLU and ConvL(w_i, h_i, f_i, w_o, h_o, f_o). Next will be the AvgPL with stride = 2 steps. Therefore, the output dimensions of the transition block will be divided into two parts, w_o = w_i/2 and h_o = h_i/2, while the number of filters will not be changed.

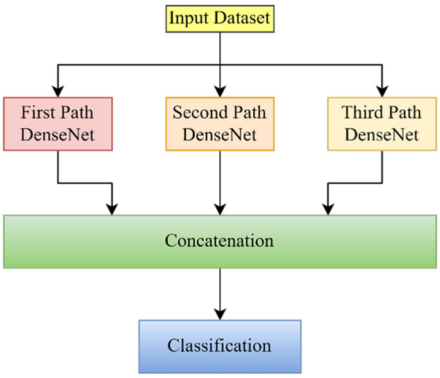

Fig. 1. General proposed methodology structure.

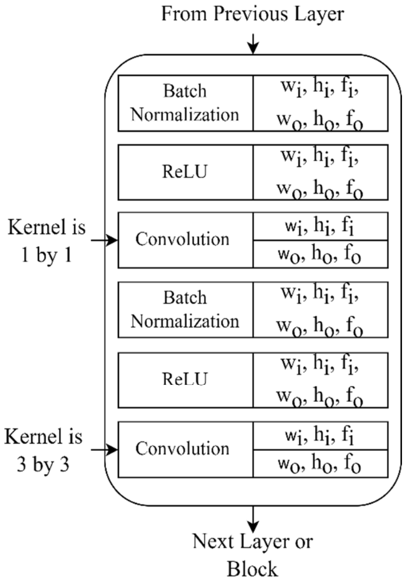

Fig. 2. Dense-block structured with six layers.

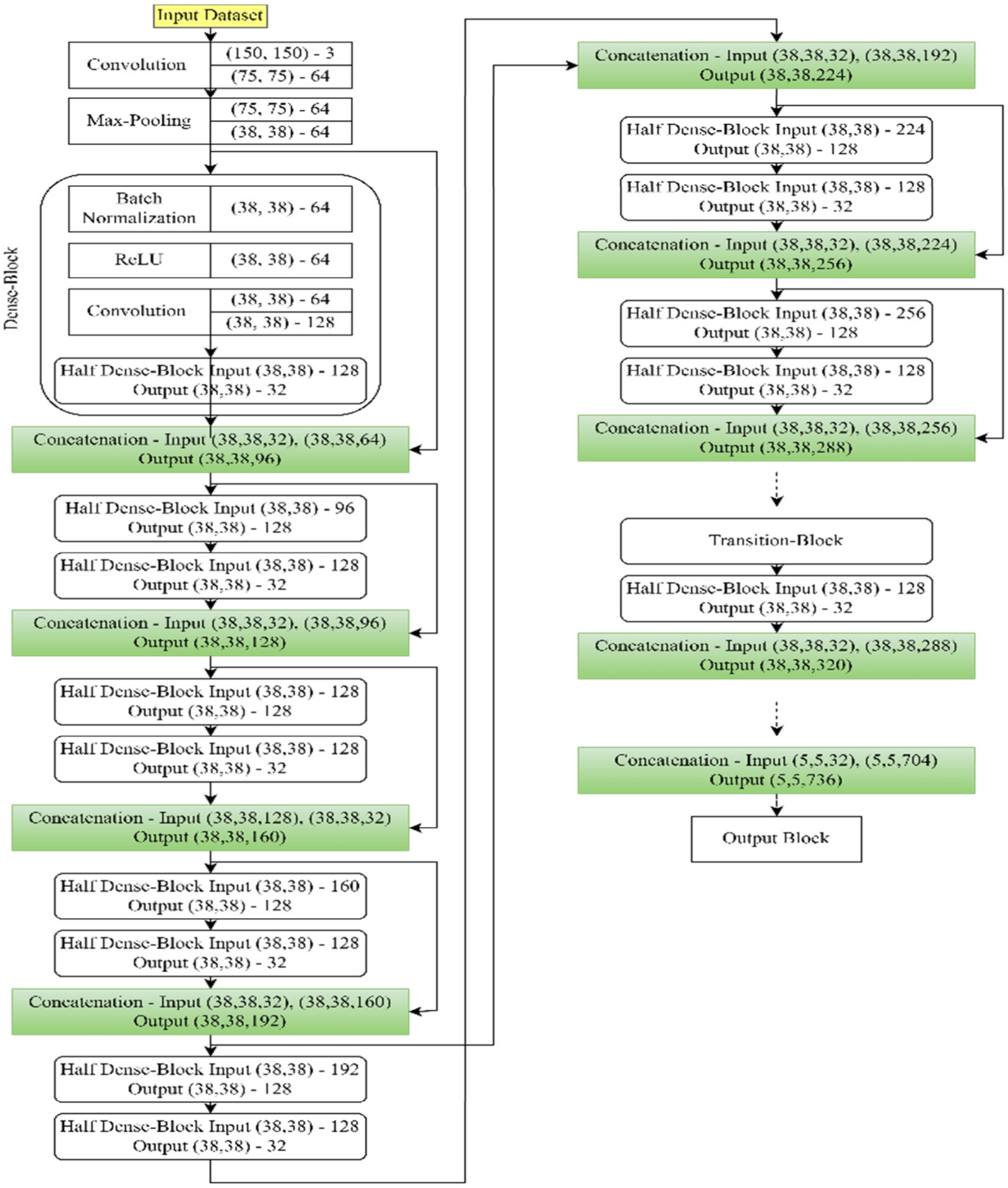
Input Dataset
Convolution (150, 150) - 3 (75, 75) - 64
Max-Pooling (75, 75) - 64 (38, 38) - 64
Dense-Block
Batch Normalization (38, 38) - 64
ReLU (38, 38) - 64
Convolution (38, 38) - 64 (38, 38) - 128
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,32), (38,38,64) Output (38,38,96)
Half Dense-Block Input (38,38) - 96 Output (38,38) - 128
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,32), (38,38,96) Output (38,38,128)
Half Dense-Block Input (38,38) - 128 Output (38,38) - 128
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,128), (38,38,32) Output (38,38,160)
Half Dense-Block Input (38,38) - 160 Output (38,38) - 128
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,32), (38,38,160) Output (38,38,192)
Half Dense-Block Input (38,38) - 192 Output (38,38) - 128
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,32), (38,38,192) Output (38,38,224)
Half Dense-Block Input (38,38) - 224 Output (38,38) - 128
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,32), (38,38,224) Output (38,38,256)
Half Dense-Block Input (38,38) - 256 Output (38,38) - 128
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,32), (38,38,256) Output (38,38,288)
Transition-Block
Half Dense-Block Input (38,38) - 128 Output (38,38) - 32
Concatenation - Input (38,38,32), (38,38,288) Output (38,38,320)
Concatenation - Input (5,5,32), (5,5,704) Output (5,5,736)
Output Block

Fig. 3. First path construction of the proposed model.

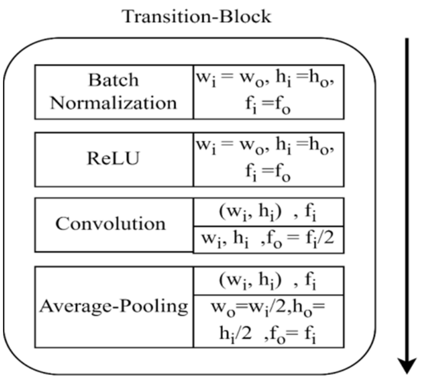
Transition-Block
Batch Normalization
$w_i = w_o$, $h_i = h_o$, $f_i = f_o$
ReLU
$w_i = w_o$, $h_i = h_o$, $f_i = f_o$
Convolution
(w_i, h_i) , f_i
w_i, h_i ,$f_o = f_i/2$
Average-Pooling
(w_i, h_i) , f_i
$w_o = w_i/2$, $h_o = h_i/2$,$f_o = f_i$

Fig. 4. Transition--Block structure, showing input and output dimensions.

The output block contains the (Global Average Pooling Layer GAPL) and and the DnsL, also known as the classification layer, as shown in Figure 5. The GAPL has a special function. To understand it, it is necessary to understand the flatten-layer function. Every tensor can be flattened into a one-dimensional version while retaining its original values using the Flatten Layer function. For instance, if you have a tensor with the values (samples, 55, 55, 64), you can write it as (samples, 55 * 55 * 64). Overfitting to the training data is possible in such a structure. DropLs can prevent overfitting in practice. For once, GAPO breaks the mold. Only the spatial dimensions are pooled down to an average value of 1, while all other dimensions are left alone. For instance, the tensor (samples 55, 55, 64) would produce the following result: (samples 1, 1, 64).

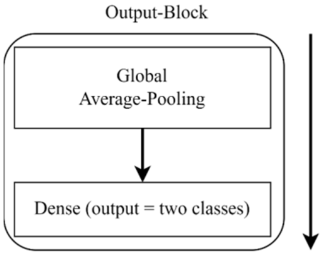

Fig. 5. Output-Block structure.

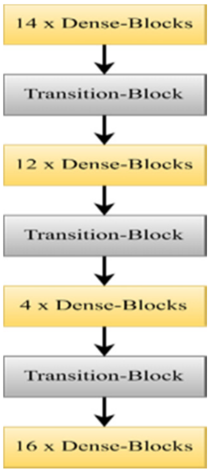

Fig. 6. Whole first-path architecture.

Consequently, the second path consisted of 6-Dense-Blocks, Transition-Block, 4-Dense-Blocks, Transition-Block, 6-Dense-Blocks, Transition-Block, and 2-Dense-Blocks. In contrast, the third path (last) is structured as: 4-Dense-Blocks, Transition-Block, 3-Dense-Blocks, Transition-Block, 2-Dense-Blocks, Transition-Block, and 4-Dense-Blocks, as shown in Figure 7. The first path has 95-ConvOs, the second has 39-ConvOs, and the third has 30-ConvOs. Further, there are three transition blocks in each path and one GAPL at the end of each path. These three paths will be concatenated and forwarded to the classification layer (Dense). Hence, there are 173 convolutional operations in the suggested model, 164 in the Dense-Blocks, and 9 in the Transition-Blocks.

A. Confusion Matrix and Performance Criteria

Classifier performance may be precisely evaluated using a Confusion Matrix (CM). The entire set of diagonals represents the successfully predicted results. The CM's off-diagonals show the misclassified results. To this end, the optimal classifier will have a CM consisting entirely of diagonal entries and zeros for all other elements. After the classification procedure, the actual and anticipated values are produced in a CM. Using the matrix numbers, one may evaluate the system's efficiency. Table I shows the CM for the two-class classifier [42].

TABLE I. CONFUSION MATRIX ORGANIZATION

		Predicted	
		Positive	Negative
Actual	Positive	True-Positive	False-Negative
	Negativee	False-Positive	True-Negative

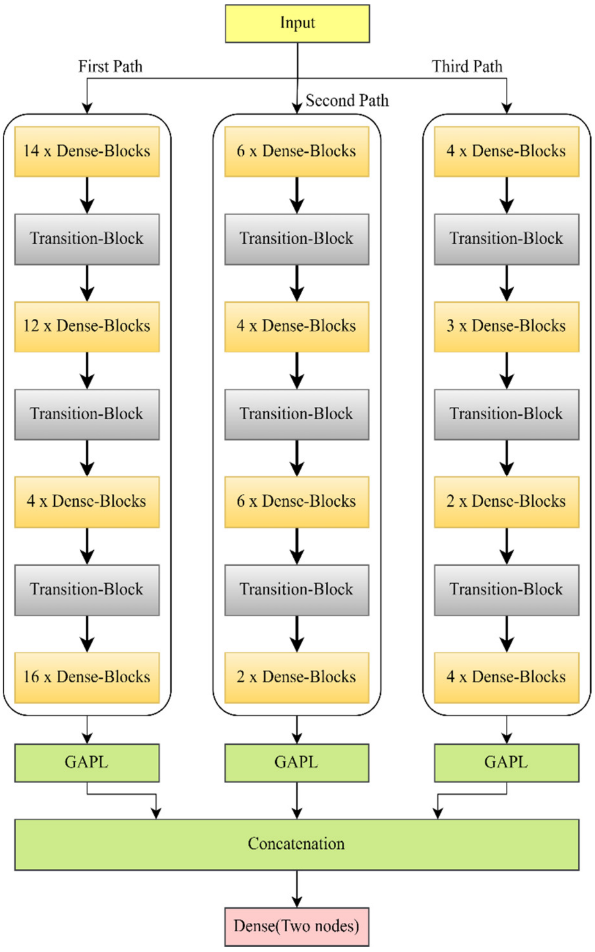

Fig. 7. The whole system structure involving three densely connected paths.

One way to evaluate various systems' efficacy is by accuracy (ACC). It considers the sum of the classifier's correct predictions [40]. In (1)-(4), TP, TN, FP, FN represent True Positive, True Negative, False Positive, and False Negative classifications, respectively.

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \quad (1)$$

Recall (Rc) or Sensitivity measures how often positive inputs are accurately identified [42]:

$$\mathrm{Rc} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (2)$$

The accuracy of an algorithm is measured by with a certain level of certainty, or

Precision (Prc) measures how many examples an algorithm predicts correctly as positive [42]:

$$\mathrm{Prc} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (3)$$

Area Under the Curve (AUC) is the area aunder the TP rate-FP rate Figure. The bigger its value, the better the model.

B. Methodology Workflow and Dataset Description

The methodology workflow can be seen in Figure 8.

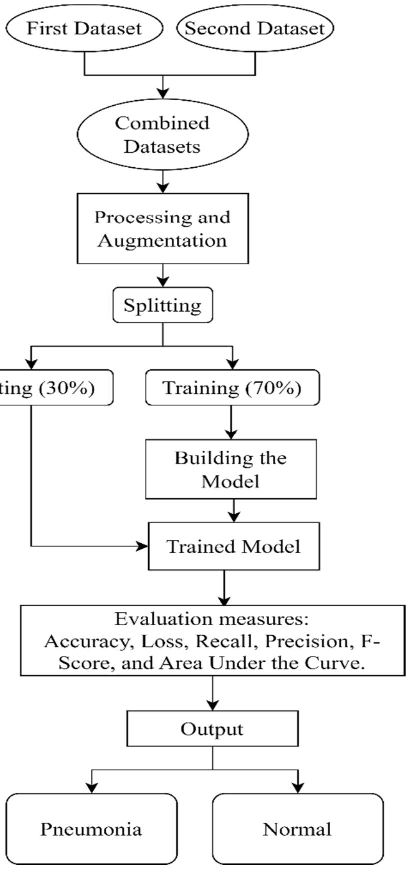

Fig. 8. Implemented methodology workflow.

The dataset employed in this work is a combination of two separate datasets, each containing two distinct classes, labelled as Pneumonia and Normal. The first data set contains 5852 (1581 Normal/Healthy and 4271 Pneumonia) CXR samples [43] and the second 5856 (1583 Normal and 4273 Pneumonia) [44], in a total of 11708 CXRs. The total Normal CXRs in the combined dataset are 3164, while the total number of Pneumonia CXRs are 8544, as shown in Table II. Table II and Figure 8 show the imbalance between the two classes as 72.98% of the combined datasets belong to the pneumonia class and only 27.02% belong to the normal class, (Figure 10). This imbalance will be treated in the subsequent section, which is an essential step to get an unbiased model. Figure 11 shows 25 randomly selected samples from the combined dataset.

TABLE II. UTILIZED DATASET STATISTICS

Dataset	Normal	Pneumonia	Total
First	1581	4271	5852
Second	1583	4273	5856
Total	3164	8544	11708

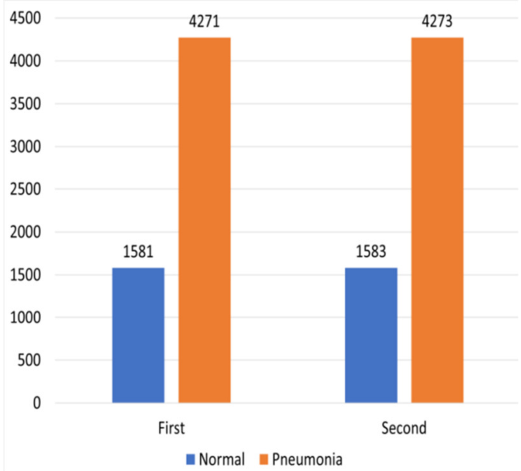

Fig. 9. Distribution of normal and pneumonia classes in the datasets.

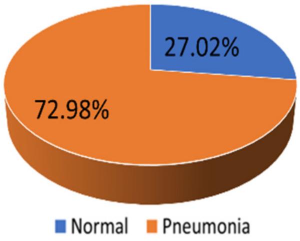

Fig. 10. Total distribution of normal and pneumonia classes.

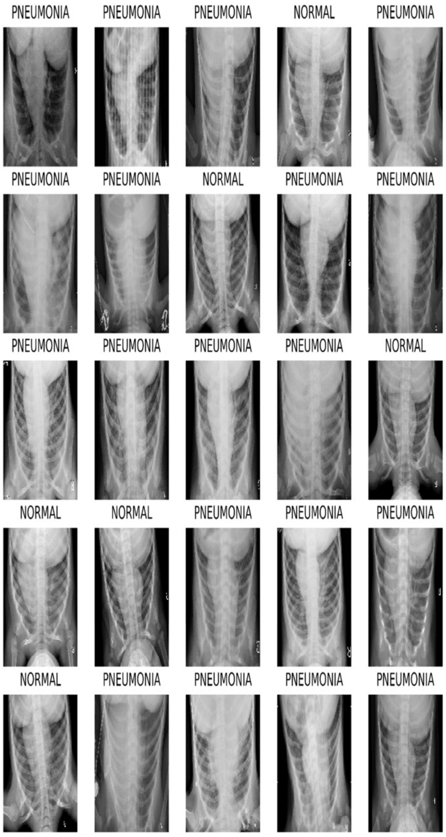

Fig. 11. Samples of the combined dataset.

C. Classe Imbalance Processing

The Class Imbalance issue troubles the majority of ML/DL classification tasks. It happens when one or more classes (majority classes) appear more frequently than the other classes (minority classes). So, there is a bias toward the majority class. In most ML techniques, it is assumed that the data are evenly distributed across all classes. When a model is trained on a dataset with an uneven distribution of classes, the results tend to favor the majority classes. The model improves its performance when it has access to a larger pool of training data, but struggles to pick up on relevant patterns that would help it learn the minority classes when fewer instances are available.

Let us examine this in greater detail. Assuming we had applied a standard cross-entropy loss to each category. One may remember that the i^th^ training data case's contribution to the cross-entropy loss is [45]:

$$L_{\text{CrossEntropy}}(x_i) = -\big(y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i))\big) \qquad (5)$$

where y_i is the label, x_i is an input feature, and $f(x_i)$ is the model's output probability that the feature is positive. Observe that only one of these variables contributes to the loss for any given training scenario, as $y_i = 0$ or $(1 - y_i) = 0$. At the same time, the other component is scaled by zero, thus becoming zero. For all N training samples, the average cross-entropy loss is:

$$L_{\text{CrossEntropy}}(D) = -\frac{1}{N}\Big(\sum\nolimits_{\text{Positive Samples}} \log(f(x_i)) + \sum\nolimits_{\text{Negative Samples}} \log(1 - f(x_i))\Big) \qquad (6)$$

where *D* is the training dataset. With this formulation, we can see that the loss will be influenced by the negative class if there is a significant imbalance, such as when there are relatively few positive training cases. When we calculate the contribution of each class (positive or negative) across all training samples, we get:

$$\text{Frequency}_{\text{positive}} = \frac{\text{Tot.Number of Positive Samples}}{N} \qquad (7)$$

$$\text{Frequency}_{\text{Negative}} = \frac{\text{Tot.Number of Negative Samples}}{N} \qquad (8)$$

Figures 10 and 11 demonstrate that the Pneumonia cases are more than the Normal instances. To ensure that the overall contribution from each class is equal, one method is to multiply each example by a weight factor that is specific to that class [45-48]:

$$w_{\text{Ne}} \times \text{Frequency}_{\text{Negative}} = w_{\text{Po}} \times \text{Frequency}_{\text{positive}} \qquad (9)$$

where w_{Ne} and w_{Po} are the negative and positive weights of negative and positive classes, respectively. Equating the positive weight to the negative frequency and vice versa is easy:

$$w_{\text{Po}} = \text{Frequency}_{\text{Negative}} \qquad (10)$$

$$w_{\text{Ne}} = \text{Frequency}_{\text{positive}} \qquad (11)$$

In this approach, both positive and negative labels will be equally represented. Once the weights are calculated, the weighted loss for each training case is:

$$L^{w}_{\text{CrossEntropy}}(x) = -\big(w_{\text{po}} y \log(f(x)) + w_{\text{Ne}}(1 - y) \log(1 - f(x))\big) \qquad (12)$$

Hence, no biasing/skewing appears in the trained model. Therefore, the next step of the methodology can be conducted easily.

D. Data Augmentation and Splitting

Data augmentation artificially increases the amount of data by creating new data points from preexisting data.

1) Preprocessing Steps

The following preprocessing steps were applied to the dataset to standardize the inputs:

- Normalization: Pixel values of all images were scaled to the range [0, 1], ensuring uniform input features across the dataset.
- Resizing: Each image was resized to 120×120 pixels, maintaining a consistent input size for the DL model.
- Flipping: Images were flipped horizontally and vertically to augment the data and simulate variations in image orientation.

2) Data Augmentation

Data augmentation was employed during the training phase to improve model robustness and mitigate the impact of class imbalance. Specific augmentation techniques included:

- Random rotations and shifts: These transformations simulated positional variations and ensured the model could generalize to different orientations.
- Intensity adjustments: Random modifications to brightness and contrast were applied to account for variations in imaging conditions.

3) Addressing Class Imbalance

The significant imbalance between normal and pneumonia cases required targeted handling to prevent model bias:

- Weighted loss function**:** A weighted binary cross-entropy loss was employed, with class weights calculated as the inverse of class frequencies.
- Weight factors: Based on the dataset distribution and (10) and (11), weights of 1.84 and 0.69 were assigned to the normal and pneumonia classes, respectively.

Moreover, the whole dataset was split into 70% and 30% groups for the training and testing phases, respectively.

E. Model Training and Hyperparameter Settings

Essential parameters should be set before initiating the training of the model. For instance, the loss function is Binary Cross-Entropy since the model should classify binary values, (Normal / Pneumonia). The classification layer in the model is sigmoid for the same reason. The selected optimizer was the RMSprop (Root Mean Squared Propagation). RMSprop is an optimization method based on gradients used to train neural networks. As the data move through a complex function, like a neural network, the gradients disappear or blow up (vanishing gradients problem). RMSprop solves this problem by normalizing the gradient using a moving average of squared gradients. This normalization keeps the step size (momentum) in check by making the step smaller for large gradients so they don't explode and bigger for small gradients so they don't disappear. Simply put, RMSprop does not treat the learning rate as a hyperparameter, but uses an adaptive learning rate. This means that the learning rate changes as time goes on.

The performance metrics were Acc, Pr, Rc, and AUC. During the training operation, the system will monitor the validation loss. The full number of epochs was set to 100 with 327 steps per each epoch. However, the system may reach the optimal state before spanning all 100 epochs. Therefore, some criteria must be set to stop the training phase when achieving the best situation. Such criteria (callbacks) are the Reduce-Learning-Rate (RLR), Model-Check-Point (MCP), and Early-Stopping (ES). The learning rate initiates at 0.001 and decreases gradually by a factor of 0.05 with steps of 2 epochs until reaching the minimum learning rate, which is 1×10^{-9}.

MCP is a Keras callback used to store model weights or the complete model at a specified frequency or whenever a quantity (training loss) is optimal compared to the previous epoch/batch. During training, MCP records the model's weights or the complete model. It lets us specify a quantity to track, such as loss or accuracy on a training or validation dataset. When the monitored metric is optimal compared to the previous epoch or batch, it can automatically store the model weights or the entire model. Model weights or the full model are saved in two formats: 'tf' and 'h5'. The MCP callback is configured to activate whenever the model's validation loss improves compared to its previous value. Table III lists all the parameters and hyperparameter settings.

TABLE III. SETTINGS OF THE PROPOSED MODEL

Hyperparameter	Settings
Splitting	Train: 70%, Test: 30%
Initial learning rate	0.001
Optimizer	RMSprop
Activation function	Sigmoid (for binary classifications)
Loss function	Binary Cross Entropy
Epoch number	100
Batch size N_B	25 (due to memory concerns)
Steps per epoch	Number of training samples/ N_B = 327
Total trainable parameters	6,451,329

On the other hand, the structured system has to be set in terms of initialization and number of trainable parameters. For instance, in each path's first layer, the ConvL, has been initialized differently. The first path’s kernel was initialized with zero mean and unity-standard deviation. The second path’s ConvL initialized with mean = 0.002 and standard deviation of 1.002. The last path’s first ConvL was initialized with mean = 0.003 and a standard deviation of 1.003. This different initializations were conducted to ensure that each path will be uncorrelated with the others and the collected information will vary from one path to another. Furthermore, the kernel size settings of the ConvLs were adjusted as the first ConvL (after the input) has 7-by-7 width and height, stride = 2, and the same padding. This applies to the other two paths of the system. Moreover, the kernel sizes of the Dense-Block are 1-by-1 and 3-by-3 for the first and second halves of the Dense-Block ConvLs, respectively.

III. RESULTS AND DISCUSSION

Although the proposed model is based on the standard DenseNet model, the system was trained from scratch. Therefore, the training dataset may not be sufficient and data augmentation operations were utilized. The implementation was achieved on the Kaggle.com website, which used its supporting GPUs, RAM, and CPUs. Figure 12 shows the model loss performance, showing that the lowest validation loss occurrs at epoch 14, which was 0.04028. However, at this epoch, the training loss was 0.00393. The accuracies at this point are 0.99853 and 0.99114 for the training and validation, respectively.

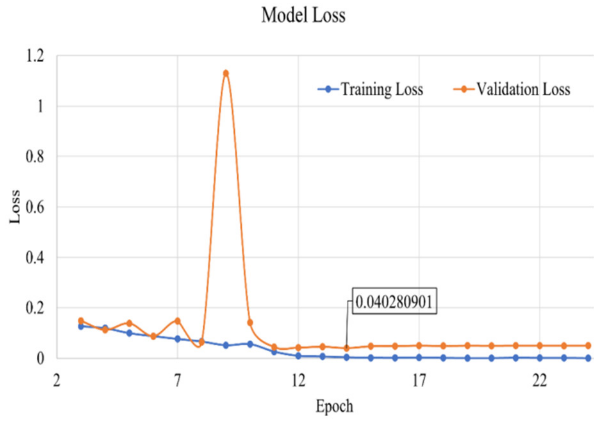

Fig. 12. Model loss performance plot after excluding first and second epochs.

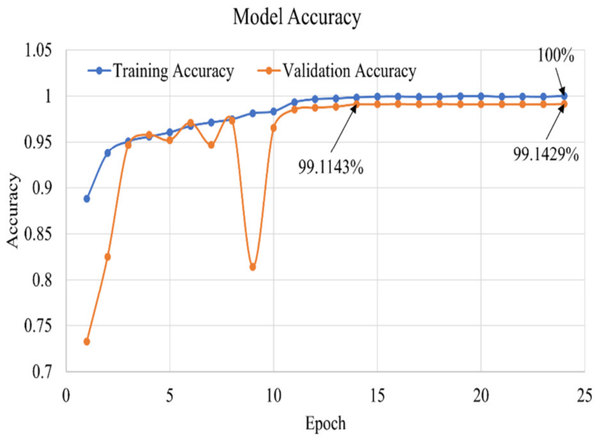

Fig. 13. Model accuracy performance.

Figure 13 shows the accuracy of the proposed structure. The validation accuracy is highest at epoch-24 (0.99143).

Nevertheless, the system's overall epoch setting was 100, but the training operation stopped early at epoch 24. The system did not improve for the next 10 epochs, as it was set in the ES callback. At epoch-24, the measure metrics for the training phase, namely loss, accuracy, precision, recall, and AUC were 0.00083, 1.00, 1.00, 1.00, and 0.999999, respectively. While for the validation phase, loss, accuracy, precision, recall, and AUC were: 0.04984, 0.99143, 0.99686, 0.991399, and 0.995498, respectively, as shown in Table IV. But at epoch-14, the measures for the training were 0.00393, 0.99853, 0.99966, 0.99832, and 0.99999 for the loss, accuracy, precision, recall, and AUC, respectively and the corresponding validation metrics were 0.04028, 0.99114, 0.99686, 0.99102, and 0.99675, respectively, as shown in Table V. Most metrics at epoch 24 performed better, but not the loss, which was the monitoring metric during training. See Table VI for a detailed comparison.

TABLE IV. METRIC MEASURES AT EPOCH NUMBER 24

Metric	Value
Train loss	0.00083
Validation loss	0.04984
Train ACC	1.00
Validation ACC	0.99143
Train Prc	1
Validation Prc	0.99686
Train Rc	1
Validation Rc	0.991399
Train AUC	0.999999
Validation AUC	0.995498

TABLE V. METRIC MEASURES AT EPOCH NUMBER 14

Metric	Value
Train loss	0.00393
Validation loss	0.04028
Train ACC	0.99853
Validation ACC	0.99114
Train Prc	0.99966
Validation Prc	0.99686
Train Rc	0.99832
Validation Rc	0.99102
Train AUC	0.99999
Validation AUC	0.99675

TABLE VI. COMPARISON OF METRIC MEASURES AT EPOCHS 14 AND 24

Metric	Value			
	Training		Validation	
	Epoch 14	Epoch 24	Epoch 14	Epoch 24
Loss	0.00393	0.00083	0.04028↓	0.04984↑
ACC	0.99853	1	0.99114↓	0.99143↑
Prc	0.99966	1	0.99686	0.99686
Rc	0.99832	1	0.99102↓	0.991399↑
AUC	0.99999	0.99999	0.99675↑	0.995498↓

The learning rate hyperparameter (Table III), was set to 0.001. As mentioned above, the learning rate should be reduced during the training operation to avoid overfitting problems. This is achieved, as shown in Figure 14. The validation loss was improved during the first 10 epochs and did not stop improving during two consecutive epochs, therefore, the learning rate did not reduce. Nevertheless, during epochs 9 and 10, the validation loss did not show improvement. Thus, the learning rate was reduced at epoch-11 to 5×10^{-5}. The next reduction occurred at epoch 17 because the validation loss did not enhance during the previous two consecutive epochs. More reductions were conducted, epochs 19, 21, and 23. The training operation should be terminated since no improvement was completed in the validation loss from epoch-14 for 10 consecutive epochs to epoch-24. Because the ES callback patience/waiting was set to 10 epochs, the system training process had to stop at this point.

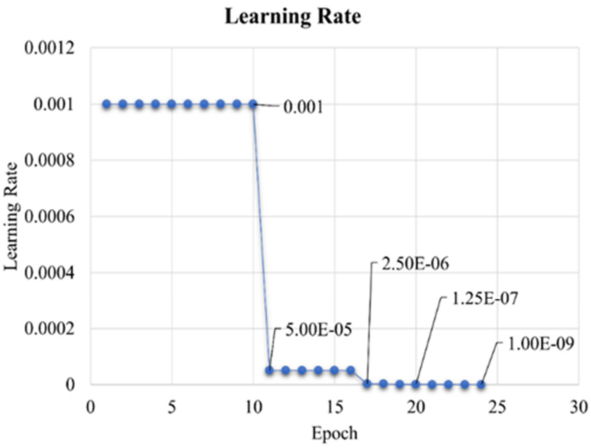

Fig. 14. Learning rate reduction during training.

At this point, the system has reached its best situation without spanning 100 epochs. Consequently, this best system situation was stored, which is at epoch-14, to be used for the prediction operation in the future. Accordingly, the system was used to predict the classes of the CXRs on the testing dataset. This dataset contained 3513 samples, which were fed to the trained model. The results can be seen in Table VII. The loss, accuracy, precision, recall, and AUC values for the new CXRs can be considered quite satisfactory when compared to those obtained in the literature. Although the trained model was never tested on the test dataset, the results matched those obtained for the training-validation set, as listed in Table VII.

TABLE VII. PREDICTION RESULTS COMPARISON BETWEEN NEVER-SEEN AND TEST-SET CXRs

Metric	Training – epoch 14	Validation – epoch 14	Testing
Loss	0.00393	0.04028	0.0261
ACC	0.99853	0.99114	0.9968
Prc	0.99966	0.99686	0.9949
Rc	0.99832	0.99102	1.0000
AUC	0.99999	0.99675	0.9978

The proposed network in this work achieved improved results compared to other known models. Since the work in [22] relied on a pre-trained network of 169 layers called DenseNet169, our network's results can be considered as an improvement. In addition, the work in [49] had a 0.924 classification accuracy, whereas [14] had a 0.934 accuracy, as can be seen in Figure 15.

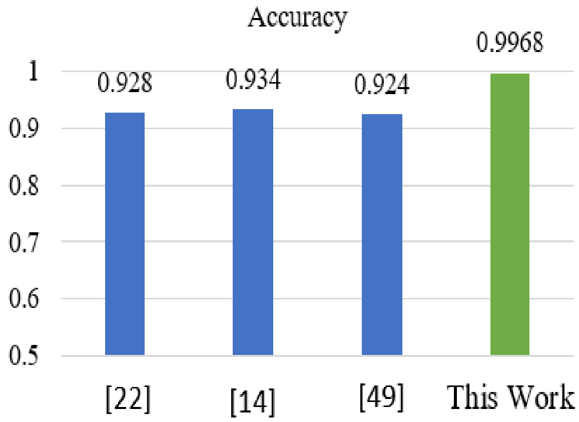

Fig 15. Accuracy comparison with other works.

TABLE VIII. MODEL PERFORMANCE COMPARISON

Model	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
Proposed CuDenseNet	99.1	99.7	99.1	99.7
DenseNet169 (pre-trained) [22]	92.8	93	92.5	93.2
ResNet50 (pre-trained) [49]	92.4	92.5	92	92.8
VGG19 (pre-trained) [14]	93.4	93.5	93	93.8
InceptionNet (pre-trained)	94.7	95	94.5	95.2

A. Limitations

While the proposed CuDenseNet model demonstrates outstanding performance, it is essential to consider certain limitations and areas for future improvement.

1) Dataset Bias

The combined dataset used in this study focuses on binary classification (normal vs. pneumonia). This potential bias limits the model’s exposure to other conditions such as tuberculosis, lung cancer, or COVID-19. The imbalance between the two classes (72.98% pneumonia, 27.02% normal) could also impact the model’s generalization ability.

2) Generalizability

The model’s performance has been validated on a specific dataset. Its applicability to other datasets, imaging modalities, and diverse patient demographics (e.g. variations in age, gender, or geographic regions) was not considered in this paper. This raises concerns about its robustness in real-world clinical settings.

3) Computational Constraints

Training the CuDenseNet model from scratch required substantial computational resources. This may limit the study's replicability for research groups or healthcare institutions with restricted access to high-performance computing infrastructure.

B. Future Research Directions

1) Multiclass Classification

Extending the model to classify multiple chest diseases would significantly enhance its clinical utility. A multiclass approach could provide a comprehensive diagnostic tool.

2) Transfer Learning

Investigating TL techniques could help leverage pre-trained models for other datasets or imaging modalities, particularly when labeled data is scarce. This approach could also reduce computational demands.

3) Cross-Dataset Validation

Future studies should evaluate the model’s performance on datasets from different sources to ensure its robustness and generalizability. Cross-institutional collaborations could help benchmark its reliability across various clinical settings.

4) Computational Efficiency

Developing lightweight and optimized model versions could reduce training and inference times. This would make the model more accessible for deployment in resource-limited settings.

IV. CONCLUSIONS

Medical imaging requiring efficient automated tools is critical for early disease diagnosis, especially in the case of pneumonia. This study introduces a novel DenseNet-based CNN model, CuDenseNet, tailored for binary classification of CXR images, trained from scratch. Unlike existing approaches relying on pre-trained models, CuDenseNet employs a custom architecture with three parallel DenseNet paths, enhancing feature extraction and classification accuracy. Key findings include:

- Exceptional performance metrics values: Accuracy: 99.1%, Precision: 99.7%, Recall: 99.1%, AUC: 99.7%
- Robust generalizability.

Compared to state-of-the-art models like VGG19 and ResNet50, CuDenseNet offers superior adaptability and independence from pre-trained weights. This model also matches or exceeds the performance of works such as [44] while introducing a novel three-path architecture for enhanced accuracy.

In conclusion, CuDenseNet demonstrates the effectiveness of training CNNs from scratch for pneumonia detection, providing a reliable tool with significant implications for clinical use and future AI research. Further studies may explore its scalability and application in broader medical imaging tasks.

ACKNOWLEDGMENTS

This paper is derived from a research grant funded by the Research, Development, and Innovation Authority (RDIA), Kingdom of Saudi Arabia, with grant number 13382-psu-2023-PSNU-R-3-1-EI-. The authors would like to acknowledge the support of Prince Sultan University, Riyadh, Saudi Arabia in paying the article processing charges of this publication. This research is supported by the Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. In addition, the authors wish to acknowledge the editor and anonymous reviewers for their insightful comments, which have improved the quality of this publication.

REFERENCES

[1] U. Hameed, M. U. Rehman, A. Rehman, R. Damaševičius, A. Sattar, and T. Saba, "A deep learning approach for liver cancer detection in CT scans," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 11, no. 7, Jan. 2024, Art. no. 2280558, https://doi.org/10.1080/21681163.2023.2280558.

[2] H. Inbarani H., A. T. Azar, and J. G, "Leukemia Image Segmentation Using a Hybrid Histogram-Based Soft Covering Rough K-Means Clustering Algorithm," *Electronics*, vol. 9, no. 1, Jan. 2020, Art. no. 188, https://doi.org/10.3390/electronics9010188.

[3] S. Al-Otaibi, M. Mujahid, A. R. Khan, H. Nobanee, J. Alyami, and T. Saba, "Dual Attention Convolutional AutoEncoder for Diagnosis of Alzheimer’s Disorder in Patients Using Neuroimaging and MRI Features," *IEEE Access*, vol. 12, pp. 58722–58739, Jan. 2024, https://doi.org/10.1109/ACCESS.2024.3390186.

[4] P. K. N. Banu, A. T. Azar, and H. H. Inbarani, "Fuzzy firefly clustering for tumour and cancer analysis," *International Journal of Modelling, Identification and Control*, vol. 27, no. 2, pp. 92–103, Mar. 2017, https://doi.org/10.1504/IJMIC.2017.082941.

[5] E. Emary, H. M. Zawbaa, A. E. Hassanien, G. Schaefer, and A. T. Azar, "Retinal blood vessel segmentation using bee colony optimisation and pattern search," in *International Joint Conference on Neural Networks*, Beijing, China, Jul. 2014, pp. 1001–1006, https://doi.org/10.1109/IJCNN.2014.6889856.

[6] S. R. Waheed, N. M. Suaib, M. S. M. Rahim, A. R. Khan, S. A. Bahaj, and T. Saba, "Synergistic Integration of Transfer Learning and Deep Learning for Enhanced Object Detection in Digital Images," *IEEE Access*, vol. 12, pp. 13525–13536, Jan. 2024, https://doi.org/10.1109/ACCESS.2024.3354706.

[7] A. Koubaa, A. Ammar, M. Alahdab, A. Kanhouch, and A. T. Azar, "DeepBrain: Experimental Evaluation of Cloud-Based Computation Offloading and Edge Computing in the Internet-of-Drones for Deep Learning Applications," *Sensors*, vol. 20, no. 18, Jan. 2020, Art. no. 5240, https://doi.org/10.3390/s20185240.

[8] D. M. Diamond, A. M. Campbell, C. R. Park, J. Halonen, and P. R. Zoladz, "The Temporal Dynamics Model of Emotional Memory Processing: A Synthesis on the Neurobiological Basis of Stress-Induced Amnesia, Flashbulb and Traumatic Memories, and the Yerkes-Dodson Law," *Neural Plasticity*, vol. 2007, no. 1, Jan. 2007, Art. no. 060803, https://doi.org/10.1155/2007/60803.

[9] J. A. Dunnmon, D. Yi, C. P. Langlotz, C. Re, D. L. Rubin, and M. P. Lungren, "Assessment of convolutional neural networks for automated classification of chest radiographs," *Radiology*, vol. 290, no. 2, pp. 537–544, Feb. 2019, https://doi.org/10.1148/radiol.2018181422.

[10] S. Varela-Santos and P. Melin, "A new modular neural network approach with fuzzy response integration for lung disease classification based on multiple objective feature optimization in chest X-ray images," *Expert Systems with Applications*, vol. 168, Apr. 2021, Art. no. 114361, https://doi.org/10.1016/j.eswa.2020.114361.

[11] M. Tamal, M. Alshammari, M. Alabdullah, R. Hourani, H. A. Alola, and T. M. Hegazi, "An integrated framework with machine learning and radiomics for accurate and rapid early diagnosis of COVID-19 from Chest X-ray," *Expert Systems with Applications*, vol. 180, Oct. 2021, Art. no. 115152, https://doi.org/10.1016/j.eswa.2021.115152.

[12] Z. Huang, Z. Pan, and B. Lei, "Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data," *Remote Sensing*, vol. 9, no. 9, Sep. 2017, Art. no. 907, https://doi.org/10.3390/rs9090907.

[13] D. Das, K. C. Santosh, and U. Pal, "Truncated inception net: COVID-19 outbreak screening using chest X-rays," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 3, pp. 915–925, Sep. 2020, https://doi.org/10.1007/s13246-020-00888-x.

[14] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 635–640, Jun. 2020, https://doi.org/10.1007/s13246-020-00865-4.

[15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015, https://doi.org/10.48550/arXiv.1409.1556.

[16] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, Apr. 17, 2017, https://doi.org/10.48550/arXiv.1704.04861.

[17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, Feb. 2017, pp. 4278–4284, https://doi.org/10.1609/aaai.v31i1.11231.

[18] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, Jul. 2017, pp. 1800–1807, https://doi.org/10.1109/CVPR.2017.195.

[19] N. Kumar, M. Gupta, D. Gupta, and S. Tiwari, "Novel deep transfer learning model for COVID-19 patient detection using X-ray chest images," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 1, pp. 469–478, Jan. 2023, https://doi.org/10.1007/s12652-021-03306-6.

[20] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1207–1220, Aug. 2021, https://doi.org/10.1007/s10044-021-00984-y.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[22] H. Nasiri and S. A. Alavi, "A Novel Framework Based on Deep Learning and ANOVA Feature Selection Method for Diagnosis of COVID-19 Cases from Chest X-Ray Images," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, Jan. 2022, Art. no. 4694567, https://doi.org/10.1155/2022/4694567.

[23] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, Jul. 2017, pp. 3462–3471, https://doi.org/10.1109/CVPR.2017.369.

[24] M. E. H. Chowdhury *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," *IEEE Access*, vol. 8, pp. 132665–132676, 2020, https://doi.org/10.1109/access.2020.3010287.

[25] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *IEEE International Symposium on Circuits and Systems*, Paris, France, Jun. 2010, pp. 253–256, https://doi.org/10.1109/ISCAS.2010.5537907.

[26] P. Rajpurkar *et al.*, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning." arXiv, Dec. 25, 2017, https://doi.org/10.48550/arXiv.1711.05225.

[27] A. U. Ibrahim, M. Ozsoz, S. Serte, F. Al-Turjman, and P. S. Yakoi, "Pneumonia Classification Using Deep Learning from Chest X-ray Images During COVID-19," *Cognitive Computation*, vol. 16, no. 4, pp. 1589–1601, Jul. 2024, https://doi.org/10.1007/s12559-020-09787-5.

[28] S. Hasija, P. Akash, M. Bhargav Hemanth, A. Kumar, and S. Sharma, "A novel approach for detection of COVID-19 and Pneumonia using only binary classification from chest CT-scans," *Neuroscience Informatics*, vol. 2, no. 4, Dec. 2022, Art. no. 100069, https://doi.org/10.1016/j.neuri.2022.100069.

[29] S. Karakanis and G. Leontidis, "Lightweight deep learning models for detecting COVID-19 from chest X-ray images," *Computers in Biology and Medicine*, vol. 130, Mar. 2021, Art. no. 104181, https://doi.org/10.1016/j.compbiomed.2020.104181.

[30] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake

City, UT, USA, Jun. 2018, pp. 8697–8710, https://doi.org/10.1109/CVPR.2018.00907.

[31] A. Gupta, Anjum, S. Gupta, and R. Katarya, "InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray," *Applied Soft Computing*, vol. 99, Feb. 2021, Art. no. 106859, https://doi.org/10.1016/j.asoc.2020.106859.

[32] T. Le Dinh, S.-H. Lee, S.-G. Kwon, and K.-R. Kwon, "COVID-19 Chest X-ray Classification and Severity Assessment Using Convolutional and Transformer Neural Networks," *Applied Sciences*, vol. 12, no. 10, Jan. 2022, Art. no. 4861, https://doi.org/10.3390/app12104861.

[33] S. M. Fati, E. M. Senan, and N. ElHakim, "Deep and Hybrid Learning Technique for Early Detection of Tuberculosis Based on X-ray Images Using Feature Fusion," *Applied Sciences*, vol. 12, no. 14, Jan. 2022, Art. no. 7092, https://doi.org/10.3390/app12147092.

[34] L. Kong and J. Cheng, "Based on improved deep convolutional neural network model pneumonia image classification," *PLOS ONE*, vol. 16, no. 11, Oct. 2021, Art. no. e0258804, https://doi.org/10.1371/journal.pone.0258804.

[35] A. Bhandary *et al.*, "Deep-learning framework to detect lung abnormality – A study with chest X-Ray and lung CT scan images," *Pattern Recognition Letters*, vol. 129, pp. 271–278, Jan. 2020, https://doi.org/10.1016/j.patrec.2019.11.013.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no.6, pp. 84-90, may 2017, https://doi.org/10.1145/3065386.

[37] M. Cicero *et al.*, "Training and Validating a Deep Convolutional Neural Network for Computer-Aided Detection and Classification of Abnormalities on Frontal Chest Radiographs," *Investigative Radiology*, vol. 52, no. 5, pp. 281–287, Dec. 2017, https://doi.org/10.1097/RLI.0000000000000341.

[38] F. Hajjej, S. Ayouni, M. Hasan, and T. Abir, "Automatic Detection of Cases of COVID-19 Pneumonia from Chest X-ray Images and Deep Learning Approaches," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, Jan. 2022, Art. no. 7451551, https://doi.org/10.1155/2022/7451551.

[39] M. B. Darici, Z. Dokur, and T. Olmez, "Pneumonia detection and classification using deep learning on chest x-ray images," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 8, no. 4, pp. 177–183, 2020, https://doi.org/10.18201/ijisae.2020466310.

[40] O. Stephen, M. Sain, U. J. Maduh, and D.-U. Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare," *Journal of Healthcare Engineering*, vol. 2019, no. 1, Jan. 2019, Art. no. 4180949, https://doi.org/10.1155/2019/4180949.

[41] H. Wu, P. Xie, H. Zhang, D. Li, and M. Cheng, "Predict pneumonia with chest X-ray images based on convolutional deep neural learning networks," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 3, pp. 2893–2907, Jan. 2020, https://doi.org/10.3233/JIFS-191438.

[42] F. Demir, "Deep autoencoder-based automated brain tumor detection from MRI data," in *Artificial Intelligence-Based Brain-Computer Interface*, V. Bajaj and G. R. Sinha, Eds. Cambridge, MA, USA: Academic Press, 2022, pp. 317–351.

[43] "chest_xray." https://www.kaggle.com/datasets/tanmaythapliyal/chest-xray.

[44] D. S. Kermany *et al.*, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, no. 5, pp. 1122-1131.e9, Feb. 2018, https://doi.org/10.1016/j.cell.2018.02.010.

[45] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, https://doi.org/10.1109/TKDE.2008.239.

[46] W. R. Abdul-Adheem, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Improved Active Disturbance Rejection-Based Decentralized Control for MIMO Nonlinear Systems: Comparison with The Decoupled Control Scheme," *Applied Sciences*, vol. 10, no. 7, Jan. 2020, Art. no. 2515, https://doi.org/10.3390/app10072515.

[47] Z. T. Allawi, I. K. Ibraheem, and A. J. Humaidi, "Fine-Tuning Meta-Heuristic Algorithm for Global Optimization," *Processes*, vol. 7, no. 10, Oct. 2019, Art. no. 657, https://doi.org/10.3390/pr7100657.

[48] I. K. Ibraheem and F. H. Ajeil, "Path Planning of an autonomous Mobile Robot using Swarm Based Optimization Techniques," *Al-Khwarizmi Engineering Journal*, vol. 12, no. 4, pp. 12–25, 2016, https://doi.org/10.22153/kej.2016.08.002.

[49] L. Wang, Z. Q. Lin, and A. Wong, "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Scientific Reports*, vol. 10, no. 1, Nov. 2020, Art. no. 19549, https://doi.org/10.1038/s41598-020-76550-z.