IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

CXR-DBICNet: A Deep Neural Network based on EfficientNetV2 for Chest X-Ray Image Classification

Suresh Mohan[1], **Subhashini Narayan**[2]

Independent Researcher, Periyar 7th Street, West Gandhi Nagar, Avadi, Chennai, India[1]

Assistant Professor, SITE, VIT University, Vellore, India[2]

Abstract: The purpose of this research is to develop a novel deep neural network model for identifying COVID disease in patients using chest X-ray images, which remain an affordable and readily available means of diagnosing lung infections. For this purpose, we propose a novel model called CXR-DBICNet (chest X-ray deep binary image classification network). We use EfficentNetV2 as our base layer and build our model on top of that so that it is optimized for binary classification. We use the pre-learned weights and customize them for use with chest X-ray images, which will enable us to extract meaningful features and classify the current problem. We use various layers of convolutions and fully connected layers to achieve the desired model. We evaluated the CXR-DBICNet model using a dataset from Mendeley's data containing 3271 normal and 1281 COVID images. In the experiment, we achieved a binary accuracy of 96.93%, a precision of 97.50%, a recall of 95.00%, and an F1-score of 96.00%. The results were then compared with five different state-of-the-art EffcientNetV2-based models. The evaluations suggest that the proposed CXR-DBICNet model achieves a superior level of binary classification performance as compared to previous studies, indicating that this model could be effectively used for the clinical preliminary diagnosis of COVID in infected patients.

Keywords: deep learning, convolutional neural networks, pre-learned model, binary classification, EfficientNetV2, chest X-ray

I. INTRODUCTION

Artificial intelligence (AI) has expanded its capabilities, evolved exponentially, and found its way into nearly every industry in a short amount of time. The availability of data, reduced storage costs, and improved computing power have all contributed to this growth. The government and business enterprises have also begun to invest heavily in AI research and development across a wide range of disciplines and sub-disciplines, complementing the efforts of academic institutions. This has led to advancements in machine learning and other AI techniques in recent years, such as deep learning, reinforcement learning, and generative models, thereby enabling the creation of more powerful AI systems. As AI grows, it is inevitable that technology will revolutionize many aspects of our lives, including healthcare. It's a positive factor that AI is growing quickly in the medical field, with many applications and prognostic value in many areas of health care. AI is being used to improve drug discovery and development, clinical decision support, personalized medicine, and the use of high-tech medical instruments. Artificial intelligence (AI) systems can examine medical images, including X-rays, CT scans, and MRIs, to assist in disease diagnosis and therapy planning [1]. For example, AI algorithms can find patterns in medical images that aren't obvious to the human eye. This helps doctors find diseases like pneumonia, COVID, cancer, etc. in their early stages. AI could change the way medicine is done by giving doctors new and better ways to diagnose and treat diseases, improve patient outcomes, and lower healthcare costs.

The worldwide spread of the Coronavirus disease (COVID) virus has caused a public health emergency with far-reaching consequences. Therefore, studies on the virus's multiple threats are urgently needed, as are studies into how to contain it and reduce its impacts. In addition to understanding the virus and creating effective vaccinations and therapies, a rapid and accurate diagnosis is crucial. X-rays are a vital part of the medical community since they are accessible to the general population and very inexpensive [2]. Radiographs may be used to diagnose COVID, evaluate the extent of the disease, and direct therapy. Medical images, such as X-rays, may be analyzed by deep learning algorithms for indications of COVID, which can cause pneumonia and other problems. By automating the X-ray processing process using deep learning algorithms, radiologists' workloads may be lightened, and they can devote more time to the most critical cases. As a result, it may facilitate a quicker and more precise diagnosis [3]. Deep learning is a branch of machine learning that takes inspiration from how the brain works. The approach utilizes multilayered artificial neural networks to handle and evaluate massive datasets [4] Some applications of deep learning algorithms have been very fruitful, including those in the fields of computer vision, natural language processing, and voice recognition.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

The artificial neuron, which is based on a biological neuron, is the basic building block of a deep learning system. These neurons take in information, process it, and then produce a result. The inputs to a deep learning system are sent through many layers of neurons before the final output is generated. When training a neural network, the neurons in each layer receive data as inputs, process that data, and then send their outputs on to the layer above them. With so many levels to learn from, the system may learn very intricate connections between inputs and outputs. Deep learning has the ability to automatically extract characteristics from raw, unprocessed data. Deep learning algorithms are able to learn meaningful representations of the data directly from the raw inputs, while typical machine learning methods need handcrafted features. As a result, deep learning systems have become far more accurate and widely used [5].

Convolutional neural networks (CNNs) are a type of deep learning algorithm commonly used for image classification and computer vision tasks. They are designed to automatically learn and extract features from image data, making them well-suited for recognizing patterns and objects in images. CNN has the ability to learn hierarchical representations of the input data, where the lower-level layers learn simple features such as edges and textures, and the higher-level layers learn more complex features such as objects and parts. For this reason, CNNs are well suited for image classification tasks since they can be trained to automatically detect and discriminate between distinct classes of objects in the input data [6]. EfficientNet is a cutting-edge architecture for convolutional neural networks (CNNs) that was developed by Google AI researchers in 2019 [7] As the most effective CNN architecture yet developed, it outperforms previous models like ResNet and MobileNet. The foundation of the network is a simple but efficient scaling strategy for increasing the size of CNNs via the use of a fixed set of parameters. The second iteration of Google's EfficientNet design, released in 2021, is known as EfficientNetV2 [8].

It is a system that can be easily expanded and improved upon, and it provides precise results. EfficientNetV2's usage of the compound scaling mechanism, which simultaneously increases the network's depth, breadth, and resolution, is a significant advancement. As a result, the network can scale to handle a broad variety of computer vision tasks without sacrificing accuracy. Recent experiments have shown that EfficientNetV2 is many times more efficient than prior state-of-the-art models, while still achieving state-of-the-art accuracy on a number of widely used benchmark datasets. The objective of this research is to improve EfficientNetV2 by enhancing its transfer learning capabilities and introducing more advanced processing layers. We reviewed several research papers and articles pertaining to deep neural networks. Research into convolutional neural networks, which are used for image processing, is a well-established field that has amassed a significant body of evidence over the years. EfficientNet and EfficientNetV2 are new technologies that was just invented, and there are now many studies being done on it. After carefully analyzing a large number of frameworks and putting some of them into practice using various online tools, we came to a conclusion that the functionality of some of the models could be improved. In order to prove that our theory is correct, we devised a computer program and put it through its trials using many well-known datasets found online. Following a number of iterations of the execution, we found that the strategy that we had presented did, in fact, yield superior results. Our inquiry has resulted in the production of this research article, in which we present both the findings of our experiment and the recommended framework that we have developed. The experiment was conducted to identify and differentiate the normal chest X-ray (CXR) images from COVID CXR images. Samples of normal and COVID CXR images are presented in Fig. 1 Sample CXR images of normal and COVID infected patients.

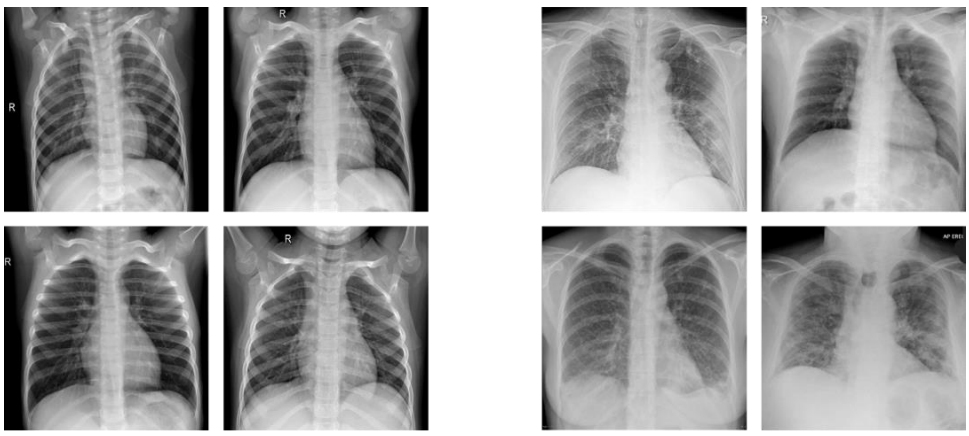

Fig. 1 Sample CXR images of normal and COVID infected patients

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

This research paper is composed of the following parts: Section 2 provides a detailed report of the existing literature. Section 3 provides the details of the existing methodology. Section 4 provides a detailed explanation of the proposed methodology. Section 5 provides details on the performance measures used. Section 6 provides details on the environment setup, dataset, pre-processing, training, and validation of the proposed model. Section 7 provides model evaluation with various metrics. Section 8 provides a detailed comparison of this proposed model with other contemporary methodologies. Section 9 provides the conclusion, followed by the reference.

II. LITERATURE SURVEY

Many contemporary researchers have used EfficientNet and EfficientNetV2 to detect lung infections using CXR images. COVID-DSNet is a deep CNN, EfficientNet based multi-classification model to detect infections using CXR images [9]. They were able to achieve a maximum of 95.97% accuracy on a 5-fold cross-validation. The authors [10] propose a multi-channel model based on EfficientNet B0, B1, and B2 pre-trained models. The features from these models are fused and passed into a non-linear, fully connected layer, and then finally into a stacked ensemble learning classifier. The model achieves 98% accuracy for COVID detection. The authors [11] propose a hierarchical classifier model based on EfficientNet, which they validated using a dataset of 13569 CXR images and compared it with five competing architectures. They also propose a cross-dataset evaluation to validate the effectiveness of the proposed model and achieve 93.9% overall accuracy. The authors [12] propose a model based on EfficientNet that could perform binary and multi-class classification, and they achieve an accuracy of 99.62% and 96.70%, respectively. They used 10-fold execution and compared their results with other state-of-the-art models for both binary and multi-class classification. The authors [13] propose a novel approach using the fixed boundary-based two-dimensional empirical wavelet transform method. It is a multi-scale CNN model that is evaluated on two datasets containing 1225 and 9000 CXR images. The authors achieved an accuracy of 96% and 97.17%, respectively, while performing a five-fold cross-validation.

CoviXNet, a novel model based on EfficientNet, is proposed by [14]. They had compared their model with other existing models and were able to attain 99.47% and 96.61% accuracy using two-class and three-class classifications, respectively. The authors [15] propose a large-scale stacked ensemble meta-classifier and feature fusion approach to detect COVID. EfficientNet is used as the base layer, and global average pooling is done to extract the required details. Principal component analysis and support vector machines were used at the higher layers, and then a logistic regression classifier was applied to achieve 99% accuracy and 99% recall rates. The authors [16] propose a model based on EfficientNet and MixNet and use transfer learning algorithms to evaluate it. Two CXR datasets consisting of 15000 and 17905 images were used to perform five-fold validation. They consistently achieved an accuracy of 95% over multiple trials. The authors [17] propose a novel ensemble model using a pre-trained transfer learning approach and evaluate it on two CXR datasets. 99.21% multi-class and 98.95% binary classification accuracy are achieved. The authors [18] propose a DeepCCXR architecture based on the EfficientNet-B5 model. The top layers are customized for binary and multi-class classification. Softmax is used for activation in addition to global average pooling and fully connected layers to strengthen their model. They validated their model on nine datasets and achieved a maximum accuracy of 93%. The authors [19] propose an ensemble of five EfficientNet models that are fine-tuned and implemented on the Montgomery and Shenzhen CXR datasets. This ensemble model based on EfficientNet B0, B1, B2, B3, and B4 achieves 97.44% accuracy, which is significantly higher than the individual accuracy of each of these models.

DFFCNet combines the advantages of a deep feature fusion module and a multi-disease classification module and then integrates them with support vector machines to provide an improved classification model [20]. In addition to performing multiple-way data augmentation on CXR images, spatial attention and channel attention are introduced. Four improvements are proposed. Improved data augmentation, using EfficientNetV2 as the backbone structure, adding a convolutional block attention module to the network, and feature fusion were done. Overall, 99.89% accuracy is achieved in multi-class classification, outperforming eight other state-of-the-art techniques. The authors [21] classify CXR images using EfficientNetV2-M and a transfer learning technique. An accuracy of 82.15% was achieved when using the National Institutes of Health (NIH) dataset, and an accuracy of 82.20 was achieved when using the Soon Chun Hyang University Hospital (SCH) dataset. The authors [22] propose the LightEfficientNetV2 model in addition to fine-tuning seven CNN in COVID detection, including EfficientNetV2. The model performs five-fold cross-validation on three different datasets. EfficientNetV2 achieved 97.73% accuracy after fine-tuning, and the LightEfficientNetV2 model achieved 97.48 % accuracy on CT images. EfficientNetV2 is also used in other problem domains in combination with other techniques. Eff2Net is an EfficientNetV2-based CNN to detect skin diseases using images [23]. The efficient channel attention block is used in both the MB-Conv and fused MB-Conv layers of the EfficientNetV2 to improve the standard squeeze and excitation block. The authors were able to achieve an accuracy of 84.70% using this method. The authors [24] propose a method to use EfficientNetV2 as convolution blocks for UNet++ to improve training speed and parameters. They have used this model on an iris database for iris segmentation. Conv3, MB-Conv, and fused MB-Conv operations are

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

performed at the base layers before applying UNet++. The authors [25] propose an improved EfficientNetV2 model fused with U2-Net to detect plant diseases. They had evaluated using EfficientNetV2-S, EfficientNetV2-M, and EfficientNetV2-L models using RGB color images. The U2_Net is used for effective background removal. The model had achieved a maximum of 98.28% accuracy in the evaluation. The model proposed by [26] is based on a three-stage training of the EfficientNetV2 architecture, which was previously trained with ImageNet. The first three layers are frozen before retraining the layers for improved efficiency in handling the glaucoma in RGB images. The number of parameters trained and the confusion matrices are shown to support the effectiveness of re-training the EfficientNetV2 model to suit specific use cases with a 96.6% F1-score. The authors [27] use EfficientNetV2 to evaluate the time-to-progression and overall survival prognosis in the treatment of hepatocellular carcinoma. Transfer learning and data augmentation were done before training the dataset, and X-tile was used as a biomarker. 82.2% average accuracy was achieved.

III. METHODOLOGY

This section provides details on the underlying methods that were used to build our proposed model.

EfficientNet is a baseline network made by using a search for neural architecture with multiple goals that try to improve both floating point operations per second (FLOPS) and accuracy. Latency is not optimized, but FLOPS are optimized. An effective network created by this search is known as EfficientNet-B0. Mas-Net and EfficientNet-B0 have similar architectures, although EfficientNet-B0 is significantly larger due to its larger FLOPS (400M). Inverted bottleneck MBConv serves as the foundation of EnfficientNet-B0, to which squeeze-and-excitation optimization is applied. Early layers of depth wise convolutions are sluggish, whereas later layers are efficient. The massive depth wise convolutions in EfficientNet are another training barrier. Compared to ordinary convolutions, depth wise convolutions contain fewer parameters and FLOPs, but they are frequently unable to fully exploit contemporary accelerators. To more effectively utilize mobile or server accelerators, Fused-MBConv is used. It substitutes a single normal conv3x3 for MBConv conv3x3 and conv1x1. In EfficientNet-B4, the original MBConv for fused-MBConv is gradually swapped out in order to compare these two building blocks.

Fused-MBConv, when used in stages 1-3, accelerates training while only slightly increasing parameters and FLOPs. However, when all blocks are replaced with fused-MBConv, it dramatically slows down training while also increasing parameters and FLOPs. A neural architecture search is performed to find the ideal combination of MBConv and fused-MBConv building blocks automatically. The Fig. 2 Representation of MBConv compared with Fused-MBConv networks depicts the details of the MBConv and fused- depicts the details of the MBConv and fused-MBConv networks.

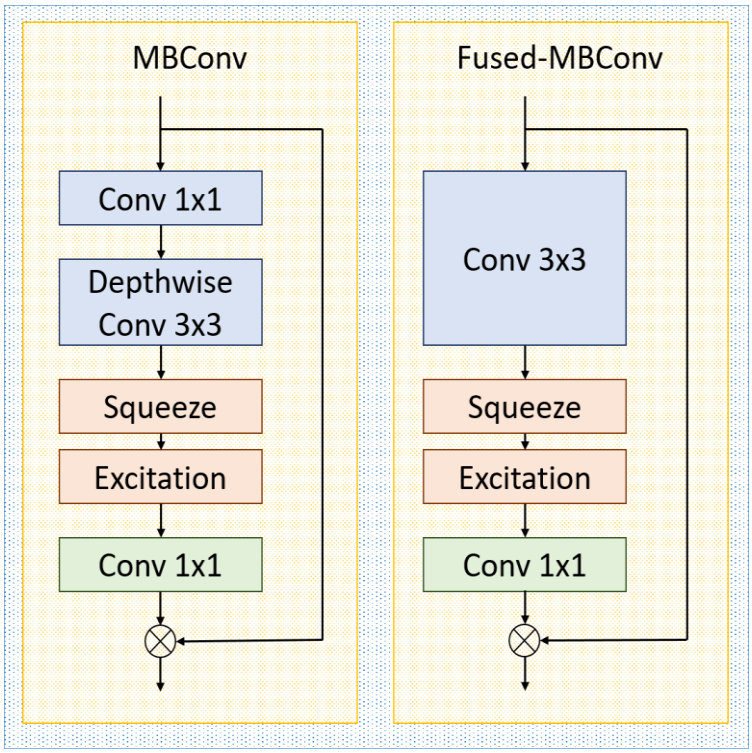

Fig. 2 Representation of MBConv compared with Fused-MBConv networks

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

EfficientNetV2 is an improvement over EfficientNet that has better efficiency and speed despite remaining smaller than the existing methods. To achieve this, improved search and scaling for neural network architecture are used. As previously described, the fused-MBConv was used in search space.

The early layers use both MBConv and fused-MBConv, with a smaller expansion ratio. Smaller kernels, preferably 3x3, are used, and the last stride in the EfficientNet is removed due to parameter size and memory overhead.

The maximum image size is reduced to 480, as large images take up a lot of memory and other resources. Also, more layers are added towards the end to enable scaling and make the network more efficient. In the early training epochs, the network is trained using smaller images and less regularization, so that simple representations can be learned by the network.

This ensures that the network is able to generalize its knowledge. Next, the images are scaled gradually with stronger regularization. We had used EfficientNetV2 as the base layer for this proposed model.

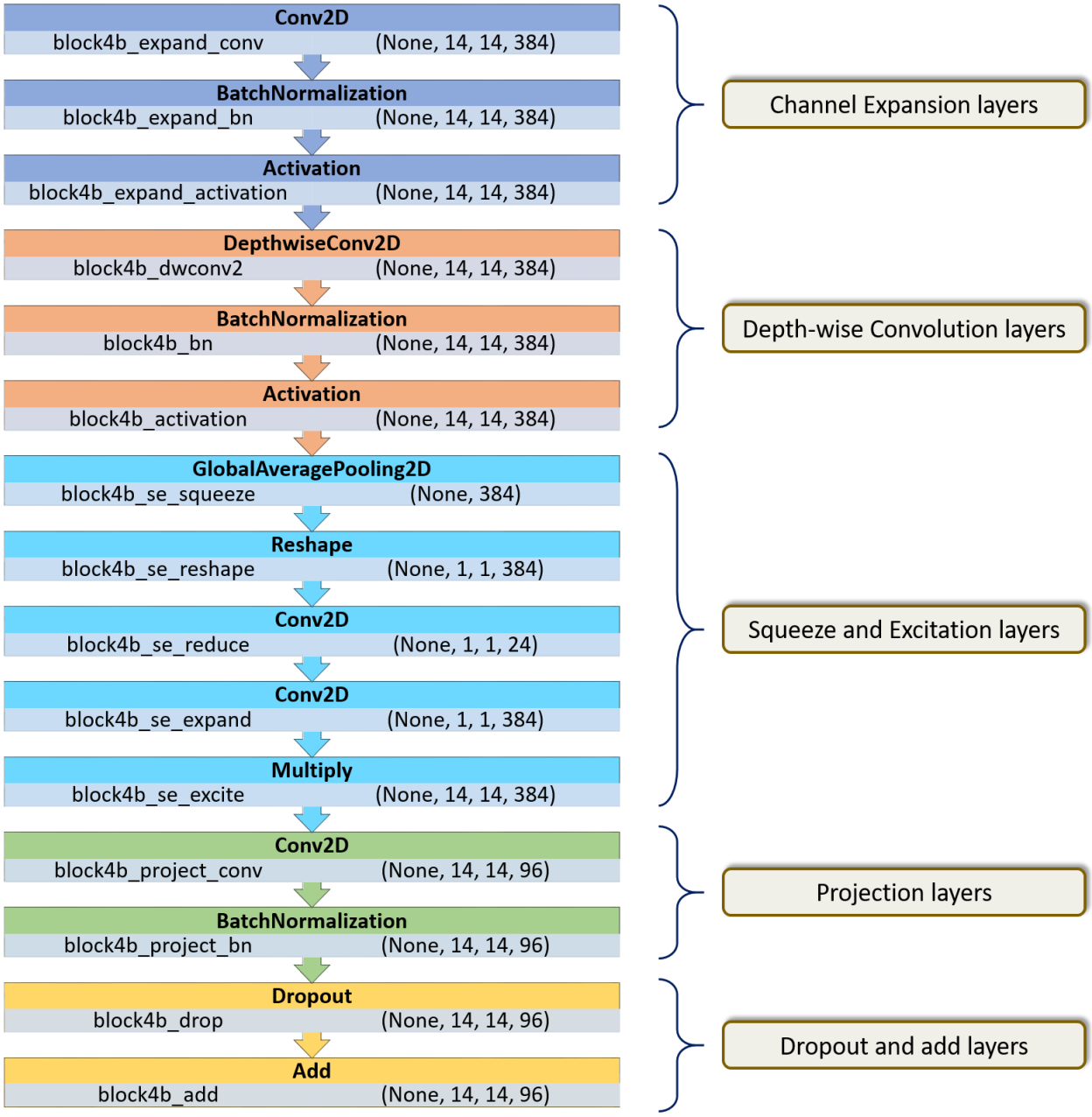

Fig. 3 Illustration of MBConv with Squeeze and Excitation block

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

We used the structure of the MBConv layers to build upon our model. A typical MBConv block contains 15 layers. They could be broadly categorized into the channel expansion layers, the depth-wise convolution layers, squeeze and excitation layers, projection layers, dropout and add layers. Adding all these layers and executing them in multiple iterations had provided an in-depth analysis of the given images. The channel expansion layers constitute of a Conv2D, batch normalization and activation layers. Depthwise convolution layers constitute of DepthwiseConv2D layers together with batch normalization and activation layers. Next is the squeeze and excitation layers which contains GlobalAveragePooling2D and Reshape layers followed by two Conv2D and a multiply layer. The projection layer consists of a Conv2D and a batch normalization layer. Then towards the end we have dropout and add layers. These layers are responsible of making sure that the model expands, squeeze, and excite at the routine intervals so that all the features are learnt and there is reduced possibility of over-fitting and under-fitting. The Fig. 3 Illustration of MBConv with Squeeze and Excitation block provides the illustration of the MBConv layer of the 4th block of the EfficientNetV2 B0 model.

IV. PROPOSED METHODOLOGY

This section provides details on the design of the proposed model. We propose a novel deep neural network model for CXR image binary classification which we would refer to as Chest X-Ray Deep Binary Image Classification Network (CXR-DBICNet). First, the input CXR images are preprocessed in which the images are prepared into an acceptable size and format as required by the model. All the images were resized into 224 x 224 resolution to get a uniform size as depicted in equation 1.

$$R = Resize(D[224{,}224]_n) = \{r_1, r_2, r_3, \dots r_n\} \tag{1}$$

The dataset is divided into three subsets each for training, validation, and testing in the ratio of 64%, 16%, and 20% respectively. This is to provide sufficient data for each of the phases. Then to mitigate the risk of overfitting, data augmentation was performed on the dataset using rotation, shear, brightness, and zoom methods which are depicted in the equations 2 through 5. The data augmentation is applied only on the training and validation sets.

$$Ro = Rotation((R_n)^{\theta}) = \{r_1^{\theta}, r_2^{\theta}, r_3^{\theta}, \dots, r_n^{\theta}\} \tag{2}$$

$$S = Shear((R_n)^{\delta}) = \{r_1^{\delta}, r_2^{\delta}, r_3^{\delta}, \dots, r_n^{\delta}\} \tag{3}$$

$$B = Brigntness((R_n)^{\rho}) = \{r_1^{\rho}, r_2^{\rho}, r_3^{\rho}, \dots, r_n^{\rho}\} \tag{4}$$

$$Z = Zoom((R_n)^{\pi}) = \{r_1^{\pi}, r_2^{\pi}, r_3^{\pi}, \dots, r_n^{\pi}\} \tag{5}$$

We had used the EfficientNetV2 B0 pre-trained model to reduce the amount of data and compute needed to train a model from scratch. Machine learning pre-trained architectures are neural network models that have been trained on huge quantities of data for a given purpose and may be fine-tuned or transferred to different tasks. Pre-trained image recognition models can recognize edges, textures, and forms, which may help them recognize objects in fresh images. When compared to training a model from scratch, fine-tuning a pre-trained model on a smaller labeled dataset improves performance on a new job with less data and computation.

Feature extraction is the process of extracting important features from new data by applying the representations learned by a previous network. These features are subsequently processed by a classifier that has been trained from scratch. There are two parts to the convolutional networks that are used to classify images. From a series of pooling and convolutional layers, they progress to a densely connected classifier. The initial component is the model's convolutional base. With convolutional networks, the process of feature extraction involves feeding the new data into the network's convolutional base and then training a classifier on top of the network's processed output.

The convolutional base could be reused but not the densely connected classifier because the representations learned by the convolutional base are more generic. It is critical that the layer depth in the model determine both the generality level and, thus, the reusability of the representations extracted by specific convolution layers.

Initial layers in the model extract more abstract features like "visual edges" and "colors and textures", whereas deeper layers in the model extract more concrete features. Therefore, as our CXR image dataset is very different from the dataset that the original model was trained on, it is possible that for feature extraction it is best to use the initial few layers of the

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

model rather than the entire convolutional structure. This is because our dataset is likely to have more information that is relevant to the task at hand. As we are using the CXR images, it is best that we use the initial few layers and determine our custom deeper layers.

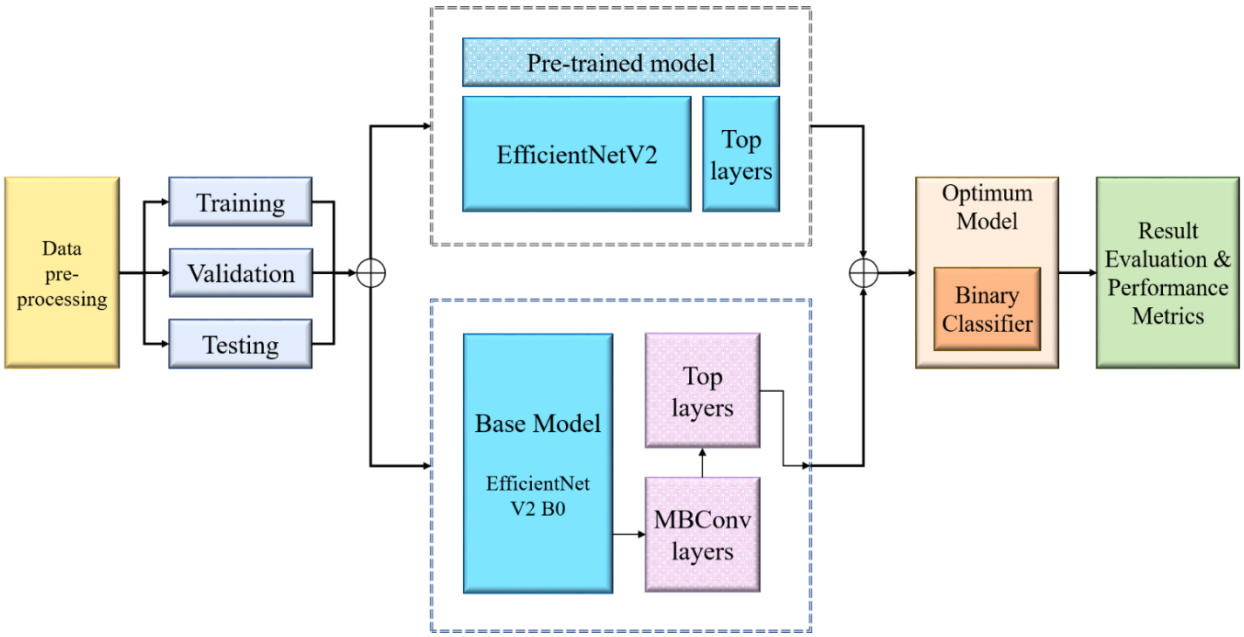

Fig. 4 Design of the proposed CXR-DBICNet model

The Fig. 4 Design of the proposed CXR-DBICNet model illustrates the overall design of the proposed CXR-DBIC model. The data is pre-processed after which it is split into training, validation, and testing sets. The training set is then provided as input to the pre-trained model based on the EfficientNetV2 B0. To fine-tune the model, the top layers that are specific to the dataset on which the EfficientNetV2 is trained is removed and a customized layers that are specific to CXR binary classification is added to the network. Then the binary classifier is added to the optimum model after which the results and the performance are subjected to evaluation and the metrics are computed. The EfficientNetV2 B0 base layers are taken and is added with the top layers. The Fig. 5 3D image of the top layers in the proposed CXR-DBICNet model provides a detailed view of the top layers in our proposed model. A dense layer, or fully connected layer, is a special kind of layer used in deep neural networks. Each neuron in a dense layer is coupled to every neuron in the layer below it. As its input, a dense layer receives a tensor, to which it first applies a linear transformation and then a non-linear activation function. During training, the network learns the weights of the linear transformation matrix. As we had added non-linearity to the network by using the activation function, this made it possible for the model to learn complex relationships between inputs and outputs. Since our proposed method stacks many dense layers, we have added regularization and dropout to prevent overfitting and make the model work better.

We also used batch normalization to make the training process more stable and to help the network come together better. The sigmoid activation function is widely utilized in deep learning neural networks. The activation function is used to bring non-linearity into the network, enabling the model to learn complicated correlations between inputs and outputs, and it is applied element-wise to the output of each neuron in the network. The output of the sigmoid function is always a positive number between 0 and 1, and it follows a distinctive S-shaped curve. Since the purpose of this problem is to make a prediction between the "Normal" and "COVID" classes, this technique is used in the problem's output layer.

The sigmoid function converts an integer input to a probability of the positive class expressed as a number between 0 and 1. Since we are working with a huge and complicated model, we employ the Adam optimizer since it has the potential to provide quicker convergence and higher performance than other optimization techniques. It is a hybrid of the gradient descent and Root Mean Square Propagation (RMSProp) optimizers, making it a gradient-based optimization technique. To optimize parameters, the Adam optimizer keeps a weighted average of the gradient and the squared gradient. This lets the optimizer have a different rate of learning for each parameter, which can change over time based on the information about historical gradients.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

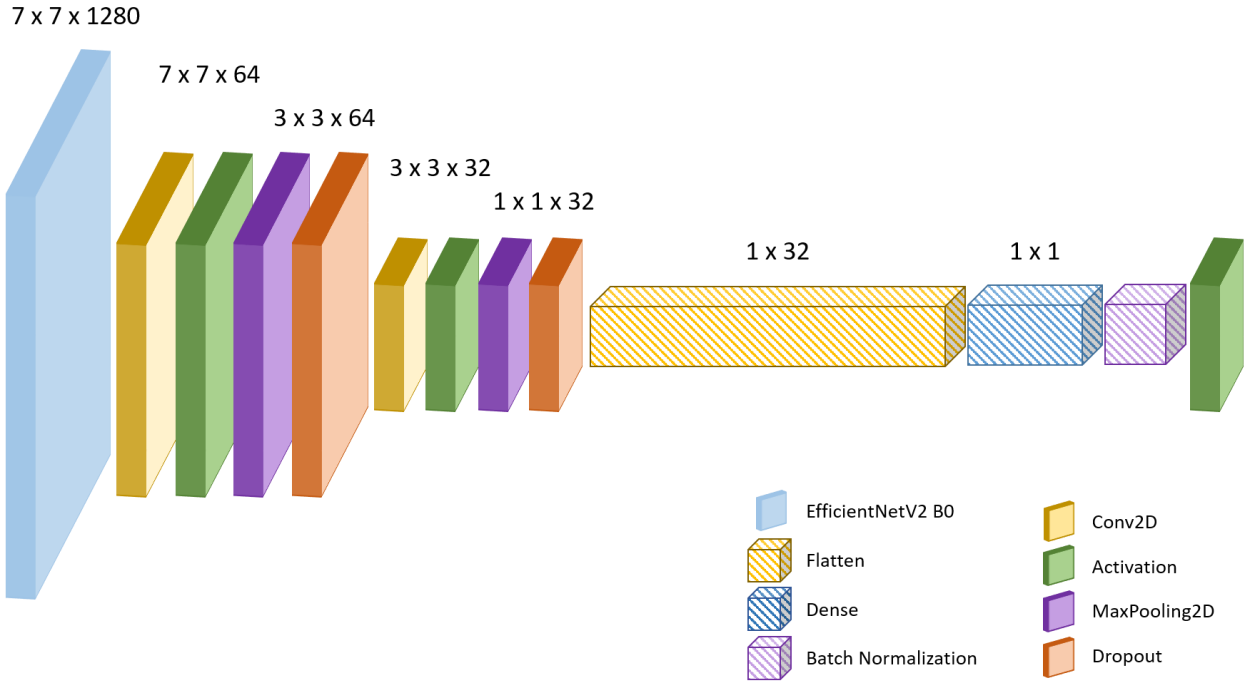

Fig. 5 3D image of the top layers in the proposed CXR-DBICNet model

A convolution operation is where two functions are combined to produce a third function that can be expressed in such a way that these functions affect each other. A generic operation generates a single output, r(t), where r is the time and both a and t are real number values. When an operation is noisy, we typically take the average of several measurements, each of which has its own weight, denoted by w(y). This could be expressed as a convolution operation, as denoted in equation 6. The function r is the input, and the function w is the feature map. A single-dimensional discrete convolutional operation is denoted in equation 7. Since neural networks use multidimensional data having multiple axes, a two-dimensional feature map is denoted using equation 8, and a cumulative convolutional operation is denoted using equation 9. A cross-correlation that is normally used in a convolutional network is denoted by the equation 10 and 11.

$$f(t) = \int r(y)w(t-y)(xy) \tag{6}$$

$$f(t) = (r * w)(t) \tag{7}$$

$$f(t) = \sum_{y=-\infty}^{\infty} r(y)(t-y)w \tag{8}$$

$$F(i,j) = \sum_{a}\sum_{b} \beta(a,b).\gamma(i-a, j-b) \tag{9}$$

$$F(i,j) = \sum_{a}\sum_{b} \beta(i-a, j-b).\gamma(a,b) \tag{10}$$

$$F(i,j) = \sum_{a}\sum_{b} \gamma(a,b).\beta(i+a, j+b) \tag{11}$$

The detailed execution structure of the proposed model is depicted in Table 1 Details of the various layers in the proposed model. The CXR images are subjected as input to the model. This is provided to the EfficientNetV2 layer. The data is sent to the Conv2D layer where the convolution operation is performed. After this layer, the activation and MaxPooling2D layers are invoked to ensure that the various features of the image are captured. The dropout layer is

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

added to the model to ensure that there is no overfitting. This block of 4 layers is then repeat for the second run but with a reduced Conv2D value. Then the flatten layer is added to convert the resultant two-dimensional values in to a continuous single dimensional value. Then the dense layer with the activation function is used to classify the output. The EfficientNetV2 B0 output is passed to the new top layers in which the convolution is performed at 64 and 32 consecutively after which it is flattened. This flattened output is then provided to the dense layer with sigmoid activation function to perform the binary classification.

Table 1 Details of the various layers in the proposed model

Layer Name	Layer Type	Input Shape	Output Shape	Parameters
efficientnetv2-b0	Functional	224, 224, 3	7, 7, 1280	5919312
conv2d	Conv2D	7, 7, 1280	7, 7, 64	81984
activation	Activation (ReLU)	7, 7, 64	7, 7, 64	0
max_pooling2d	MaxPooling2D (2,2)	7, 7, 64	3, 3, 64	0
dropout	Dropout (0.8)	3, 3, 64	3, 3, 64	0
conv2d_1	Conv2D	3, 3, 64	3, 3, 32	2080
activation_1	Activation (ReLU)	3, 3, 32	3, 3, 32	0
max_pooling2d_1	MaxPooling2D (2,2)	3, 3, 32	1, 1, 32	0
dropout_1	Dropout (0.8)	1, 1, 32	1, 1, 32	0
flatten	Flatten	1, 1, 32	32	0
dense	Dense (1)	32	1	33
batch_normalization	BatchNormalization	1	1	4
activation_2	Activation (sigmoid)	1	1	0

V. PERFORMANCE MEASURES

We used multiple performance measures to evaluate our proposed model. **Accuracy**: The number of overall accurate forecasts is an indicator of accuracy. The ratio of the total number of true positives and true negatives to the total number of observations is used to compute this. **Precision**: The total number of correct positive predictions is an indicator of precision. It is calculated by dividing the total of true positives by the total of positive observations. **Sensitivity**: Sensitivity (recall) is a measurement of the proportion of accurate positive predictions among all positive forecasts. The ratio of all true positives to the total of true positives and false negatives is used to compute it. **Specificity**: The indicator of how many accurate negatives were produced from all the negatively anticipated values is called specificity. The ratio of all true negatives to the total of true negatives and false positives is used to compute it. **Support**: The support of a given class refers to how frequently it actually appears in the dataset. **F-measure**: When comparing two data sets, the F-measure is used to identify whether the data set has higher precision but lower recall, or vice versa. This is done by taking the harmonic mean of the recall and precision. It is determined by dividing the product of recall and precision by two and adding the result to recall and precision.

Learning Curve: The learning curve is a graphical depiction of a relationship between two linked parameters. The number of epochs that were completed with the intention of reaching the goal and the accuracy of the learning method were represented by the learning curve in this study. When the epoch, which is recorded on the horizontal axis, increases, the precision, which is measured on the vertical axis, typically increases as well. It is typically understood to mean that the amount of accuracy that is attained increases with the number of epochs applied to the learning procedure. We were able to ascertain that this specific curve does not, however, show a continual rise. Instead, at a certain point, the accuracy was not significantly affected by adding more epochs. Despite the fact that our initial presumption was that we would face a severe learning curve, we later realized that it would be more gradual. Our research led us to the conclusion that improving the accuracy would not be significantly aided by lengthening the epochs. Additionally, running more epochs becomes less effective at a certain point. The second observation relates to the curve's slope (gradient) angle. It didn't appear to be dependent on how well the learning approach performed overall. It expresses the anticipated rate of performance growth as the number of epochs rises.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

Confusion Matrix: Using a table known as a confusion matrix, it is possible to evaluate the efficacy of a learning technique. Visualizing and summarizing performance data is facilitated by a confusion matrix. It is a straightforward tabular representation capable of displaying a variety of properties. In this work, the confusion matrix was utilized to depict the link between actual true and false outcome values and projected true and false outcome values. A true-positive (TP) occurs when both the actual and expected results are true. A false-positive (FP) occurs when the actual result is false but the projected result is true. A false-negative (FN) occurs when the actual result is correct but the projected result is incorrect. A true-negative (TN) is a situation in which both the actual and projected outcomes are false.

VI. EXPERIMENT & RESULTS

This section provided details on the experiment that we conducted using CXR images on our proposed CXR-DBICNet to predict COVID infections.

The experiment was performed on Kaggle. The system configuration with Python 3.7.12 packaged by conda-forge, Keras version 2.10.0, Tensorflow version 2.10.0, and sklearn version 1.0.2 was used.

DATASET

We used the dataset from Mendeley's data in our experiment. The dataset complied by [28] is a compilation of 15 separate datasets accessible to the public; its authors categorized and organized the images into COVID and normal categories, among others. Inception V3 architecture was utilized to produce image embedding, and subsequently, unsupervised learning methods were employed to detect flawed images and remove them. This data set included both COVID x-rays (1281) and regular x-rays (3271). We divided the dataset into three parts: training data, validation data, and test data as provided in Table 2 Data distribution. Each part had both normal and COVID CXR images. Twenty percent of these images were used for testing purposes. Out of the remaining 80%, 20% of the samples were utilized for validation, whereas the other 80% were used for training. The images were arbitrarily selected and programmatically allocated to each of these groups.

Table 2 Data distribution

Dataset	Percent	Normal	COVID	Total
Training	64%	2080	833	2913
Validation	16%	528	200	728
Testing	20%	663	248	911
Total	100%	3271	1281	4552

DATASET PREPROCESSING

The dataset comprises COVID and normal images that we enhanced using several techniques. First, the images were randomly separated into three groups, as previously mentioned. To further mitigate data overfitting, we used four distinct data enhancement techniques. Twenty degrees of random rotation are applied to the images.

The brightness of the image is adjusted between 0.6 and 1.3 points. The images are then chopped at a counterclockwise angle of 0.3 degrees. The images are then zoomed between 0.8 and 1.0. Using these procedures, the training set and the validation set comprise both original and enhanced images. The Table 3 Summary of data augmentation methods used provides a summary of the data enhancement performed on the images.

Table 3 Summary of data augmentation methods used

Augmentation Method	Value
Rotation range	20 degrees
Brightness range	0.6 to 1.3
Shear range	0.3 degrees
Zoom range	0.8 to 1.0

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

CXR-DBICNet MODEL

We built a neural network model based on our proposed CXR-DBICNet and trained it with the dataset. This state-of-the art model is built on the EfficientNetV2 B0 model. The model is built with various blocks and layers in it. The most prominent among the blocks is the MBConv block that is typically made up of 15 layers. The EfficientNetV2 B0 model has multiple such layers as described earlier. For the base model, we provided the input shape as (224, 224, 3) and added standard pre-trained weights that was trained on ImageNet. The layers in the base model is set to trainable so that the model learns about the current scenario and adjusts its weights accordingly. This would facilitate an improved performance for the current model. We removed the top layer from the model and then applied our custom model on that.

We included Conv2D, MaxPooling2D, Dropout, Flatten, and Dense layers with activation function wherever applicable in sequential order. For the Conv2D layers we used ReLu activation function and for the dense layer we used sigmoid activation function. The first Conv2D layer size is 64 and the subsequent Conv2D layer size is 32 before it is flattened. The dropout ratio is set to 0.8 in both the instances. The MaxPooling2D layer is added and a pool size (2, 2) is provided. BatchNormalization is used and the activation is performed.

To monitor the performance of the model, performance metrics were initiated. Binary accuracy, AUC, precision, and recall were added as performance metrics. The optimizer is set to Adam with a learning rate of 1e-1. The loss is set to binary cross-entropy and the model is compiled. Once the model is successfully compiled, the model is trained using the fit method. The training set and the validation set that we created earlier is provided to the model.

Epochs measure the number of times that the training dataset is sent through the model's processing pipeline throughout its execution. The results of each epoch are different, and to get the best outcomes possible, it is common practice to factor in each epoch's performance relative to the overall average of all of these epochs. In response, the input data is segmented into batches to facilitate the learning process and the subsequent updating of the parameters. The model will do an update on its internal parameters and weights at the beginning of each of these epochs, which may result in an improvement in the model's overall performance. To ensure that this model would be consistent throughout its execution, we employed a total of 50 epochs.

TRAINING AND VALIDATION

This description represents the training and validation metrics achieved by the proposed model over a total of 50 epochs. During each epoch, the model was trained on a training set and then evaluated on a validation set to obtain the corresponding training and validation metrics. The model is evaluated using the binary accuracy, precision, loss, and recall metrics.

The binary accuracy scores are used to evaluate the model’s performance. The trend in binary accuracy during validation and training phases are depicted in Fig. 6 Model binary accuracy during training and validation phases.
The training accuracy represents the accuracy achieved by the model on the training set during each epoch, while the validation accuracy represents the accuracy achieved on a separate validation set. The validation set is used to evaluate the generalization performance of the model, i.e., how well it can predict new data that it has not seen during training. At the beginning of the training, the model achieved an accuracy of 69.28% on the training set and 72.53% on the validation set.

As training progressed, the model's accuracy on the training set increased steadily, reaching a peak of 96.50% at epoch 50. On the other hand, the validation accuracy had a lot of fluctuations but generally showed an increasing trend. It reached its peak at epoch 50, with a value of 98.21%. From epochs 1 to 5, the model's training accuracy improved steadily, indicating that the model was learning from the data. During this period, the validation accuracy also showed improvement. However, from epochs 6 to 9, the validation accuracy began to level off.

The model's performance continued to fluctuate, with the validation accuracy increasing at some epochs and decreasing at others. At epoch 19, both the training and validation accuracies peaked at 93.51% and 96.98%, respectively. At epoch 10, the model experienced a sharp drop in validation accuracy, which could indicate a problem with the data or the model itself. However, the model recovered and reached a new peak in validation accuracy at epoch 33, with a value of 98.35%. In general, the model's accuracy improved with each epoch until epoch 50, where it achieved its highest accuracy.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

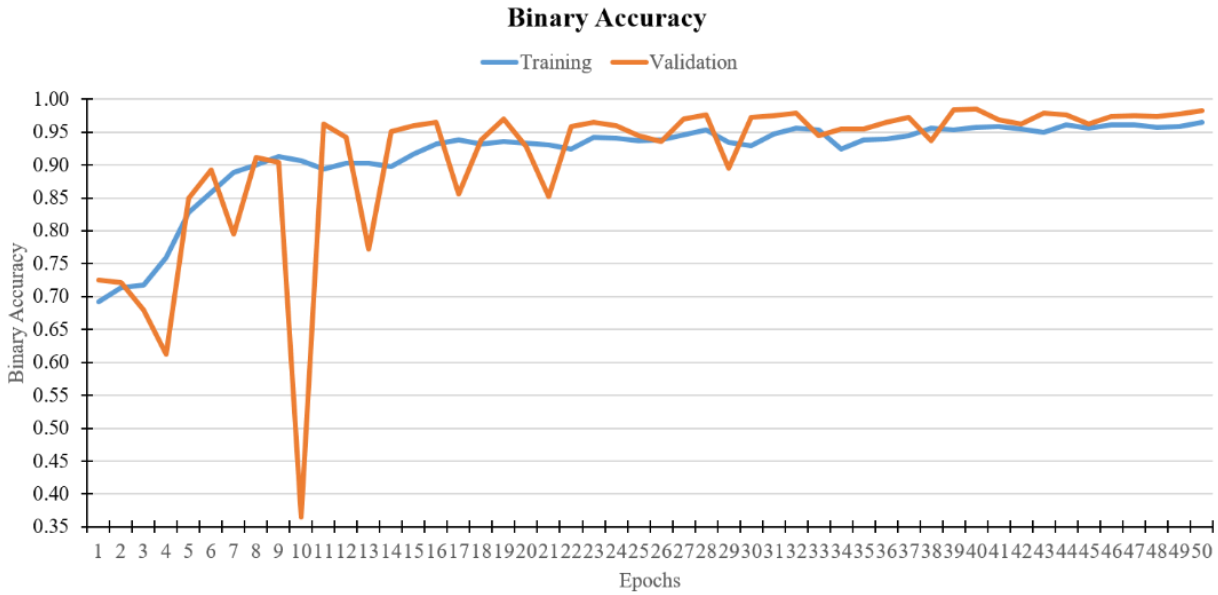

Fig. 6 Model binary accuracy during training and validation phases

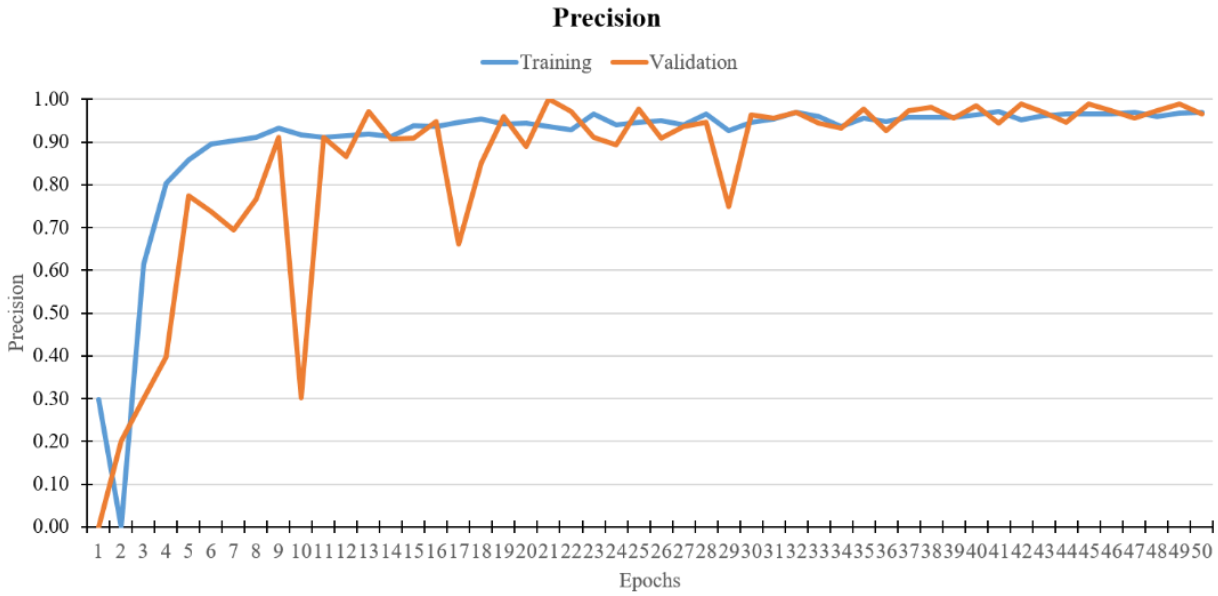

Fig. 7 Model precision during training and validation phases

Next, the precision scores are used to evaluate the model's performance. The trend in precision during validation and training phases are depicted in Fig. 7 Model precision during training and validation phases. Precision is a metric that measures the proportion of true positives among all positive predictions. Looking at the trend of precision scores over the 50 epochs, it is observed that both training and validation precision scores start from very low values in the initial epochs and gradually increase as the model is trained further. The training precision score shows a relatively smooth increase throughout the training process, reaching a high of 0.9692 in the last epoch. On the other hand, the validation precision score shows a more fluctuating pattern, with some sudden drops in some epochs followed by recovery in others. From an initial value of 0.2987 for the training precision, the model experiences a significant improvement in performance over the first four epochs, reaching a precision of 0.8037. Subsequently, the model's training precision continues to increase gradually, reaching a peak value of 0.9710 at epoch 41.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

The validation precision also follows a similar pattern, with an initial value of 0.0, increasing to 0.2000 at epoch 2 and gradually increasing thereafter. In general, the model's performance appears to be relatively stable throughout most of the training process, with occasional fluctuations in both training and validation precision. There are also some cases where the training precision is significantly higher than the validation precision, which could indicate overfitting.

However, the model's overall performance appears to be consistently high. Next, the recall scores are used to evaluate the model's performance. The trend in recall during validation and training phases are depicted in Fig. 8 Model recall during training and validation phases. Recall is a metric that represents the proportion of actual positive samples that are correctly identified by the model as positive. The recall score ranges from 0 to 1, with 1 indicating perfect recall. Looking at the data, we can see that the initial recall scores on both the training and validation sets are very low, indicating that the model is initially not performing well and needs further training to improve its performance.

As we move forward in epochs, we can see that the training recall score starts to improve steadily. By epoch 4, the training recall score reaches 0.2113, which is a considerable improvement from the initial score. The training recall score continues to improve and reaches its maximum value of 0.9064 in epoch 50. On the other hand, the validation recall score shows a more irregular pattern. We can see that the validation recall score initially stays at 0 and then starts to fluctuate. We can also observe that there are some epochs where the validation recall score is higher than the training recall score, such as in epochs 6, 8, 10, 17, 23, 27, 28, 31, 32, 36, 43, 44, and 50.

This behavior indicates that the model may be overfitting the training data. Overall, the model's performance on the validation set improves as we move forward in epochs. The highest validation recall score is achieved in epoch 10 with a score of 1.0, indicating that the model correctly identifies all positive samples in the validation set. To summarize, the data shows that the model's training recall score steadily improves as we increase the number of epochs, while the validation recall score over the epochs.

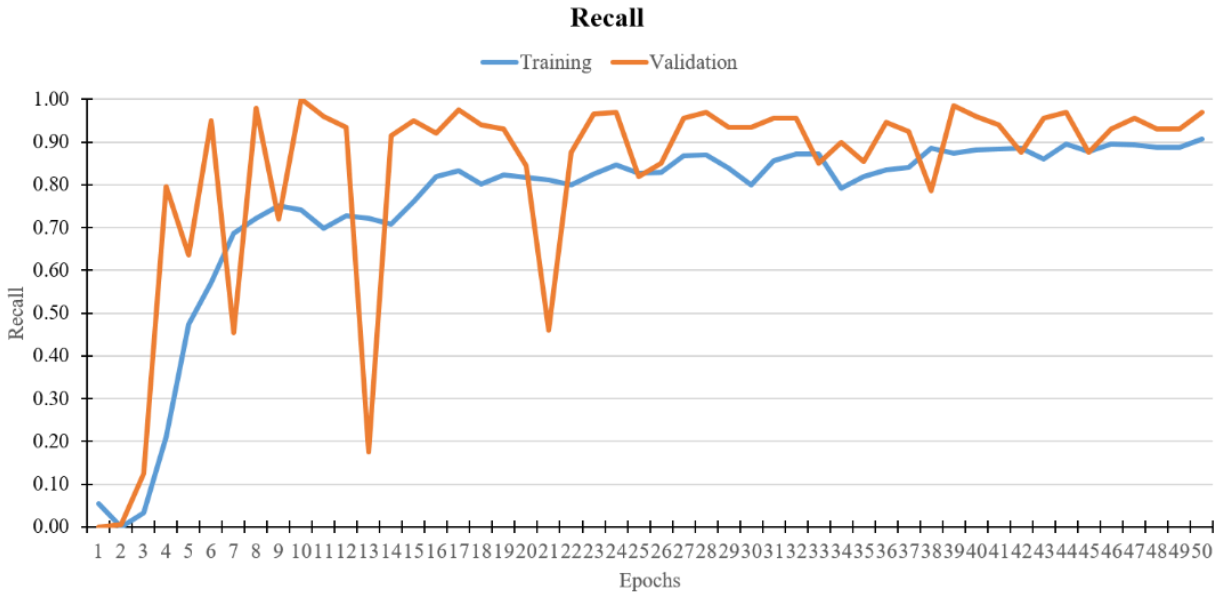

Fig. 8 Model recall during training and validation phases

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

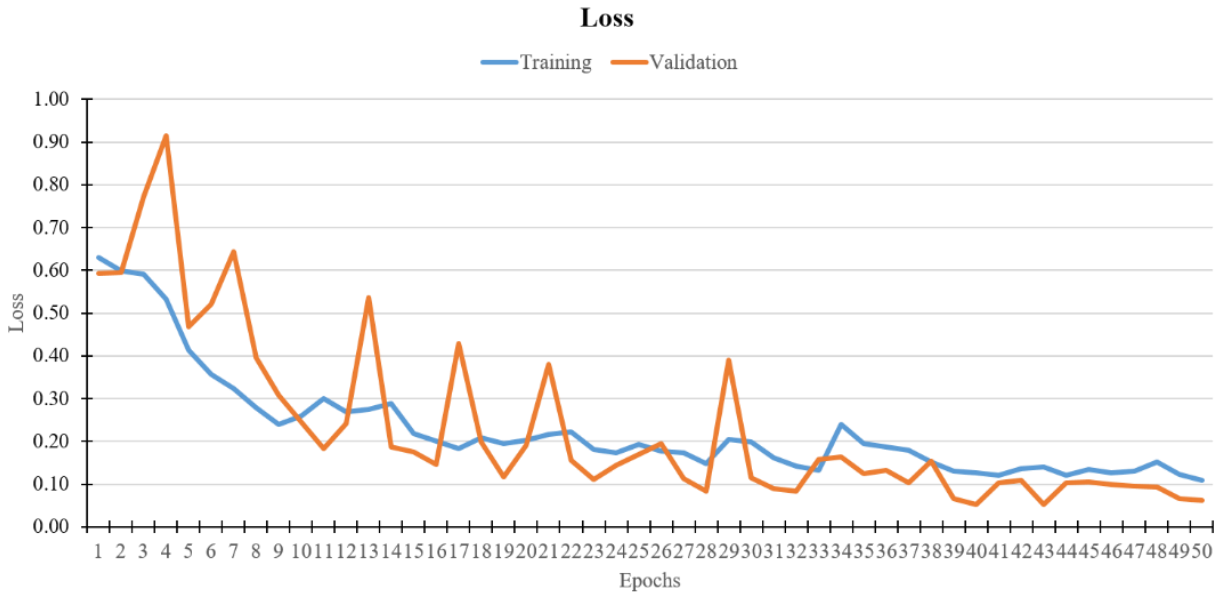

Fig. 9 Model loss during training and validation phases

Next, the loss scores are used to evaluate the model's performance. The trend in recall during validation and training phases are depicted in Fig. 9 Model loss during training and validation phases. The training loss is the error in the model's predictions on the training data, whereas the validation loss is the error on a separate validation dataset. To achieve the best model performance, the goal is to minimize both training and validation loss. We can see from the data that the training loss decreases steadily over time, indicating that the model's ability to fit the training data is improving over the 50 epochs. The validation loss, on the other hand, decreases initially but then begins to fluctuate and even increase in some epochs. Another trend we can see is that the validation loss is generally higher than the training loss, which is to be expected given that the model did not see the validation data during training and thus may be less well-suited to making predictions on it. In terms of specific epochs, we can see that the validation loss is greatest in epochs 3 and 4, implying that the model has over fitted to the training data at this point. The lowest validation loss occurs in epochs 28 and 40, indicating that the model is performing well on the validation data in these epochs. In summary, this data shows that the model's ability to fit the training data improves over time, but it may not generalize well to new data, as evidenced by the fluctuating and increasing validation loss.

TESTING & EVALUATION

The model is next subjected to evaluation to check it is yielding desired results using the evaluate method and the data from the test set created earlier is passed. Next the model is now tested if it could predict the results properly. The data from the test set is provided. The results are then plotted using the confusion matrix. Then the classification report is generated from the results. Finally, the ROC curve is generated for the results.

During model evaluation, the model has achieved a low loss score of 0.0926, which indicates that it is performing well and is able to minimize the difference between predicted and actual values. The binary accuracy score of 0.9693 indicates that the model is able to predict correctly in 96.93% of the cases. The AUC score of 0.9932 is a measure of how well the model is able to distinguish between positive and negative classes. A score of 1.0 indicates perfect discrimination, while a score of 0.5 indicates random guessing. A score of 0.9932 indicates that the model is able to distinguish between the two classes very well.

The precision score of 0.9783 indicates that out of all the samples that the model classified as positive, 97.83% were actually positive. The recall score of 0.9073 indicates that out of all the actual positive samples, the model correctly identified 90.73%. A high precision score is desirable when the cost of false positives is high, while a high recall score is desirable when the cost of false negatives is high. Overall, the model seems to be performing well based on the evaluation metrics. Even though the recall score is slightly lower, the binary accuracy, AUC, and precision are all quite high.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

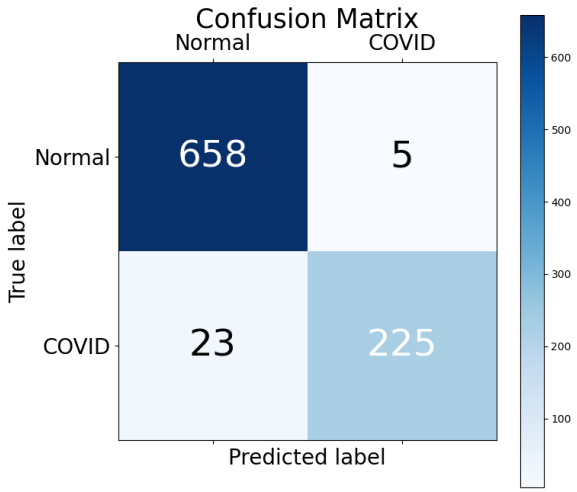

Fig. 10 Confusion matrix of the training results

The Fig. 10 Confusion matrix of the training results shows the confusion matrix, which is a detailed view on how well the training model works. The trained model is then given the test data to see how well it can identify and classify things. A total of 663 normal images were given. The trained model is able to identify 658 images properly and 5 images were improperly identified as COVID. Next, out of the total 248 COVID images provided, the model is able to identify 225 images correctly, however, 23 images were incorrectly identified as normal image. Normal images were correctly identified 99% of the time, while COVID images were correctly identified about 91% of the time.

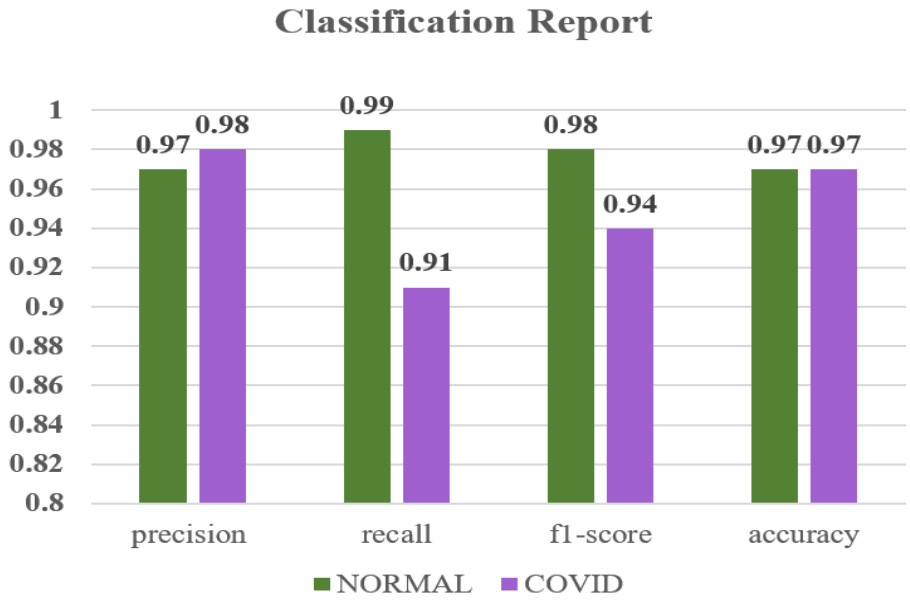

Fig. 11 Classification report of the training results

The Fig. 11 Classification report of the training results provides the classification report that gives a detailed analysis, broken down by class, of how well a model works for binary classification. In this case, the model was trained to classify CXR images into two classes: normal and COVID. The precision for the NORMAL class is 0.97, which means that the model correctly classified 97% of the X-ray images as normal. For the COVID class, the precision is 0.98, indicating that 98% of the images classified as COVID were indeed COVID. The recall for the normal class is 0.99, which means that 99% of the actual normal cases were correctly identified by the model. For the COVID class, the recall is 0.91, indicating that 91% of the actual COVID cases were identified by the model. The F1-score for the normal class is 0.98, and for the COVID class, it is 0.94.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

Accuracy is the overall performance of the model and is calculated as the ratio of correctly predicted samples to the total number of samples. In this report, the accuracy for both classes is 0.97, indicating that the model performed well overall. In this report, there were 663 normal samples and 248 COVID samples for support.

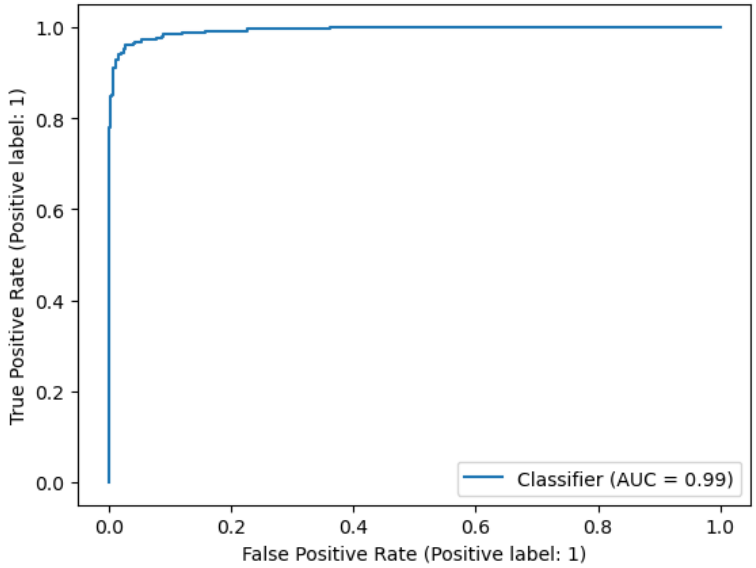

Fig. 12 ROC Curve of the training results

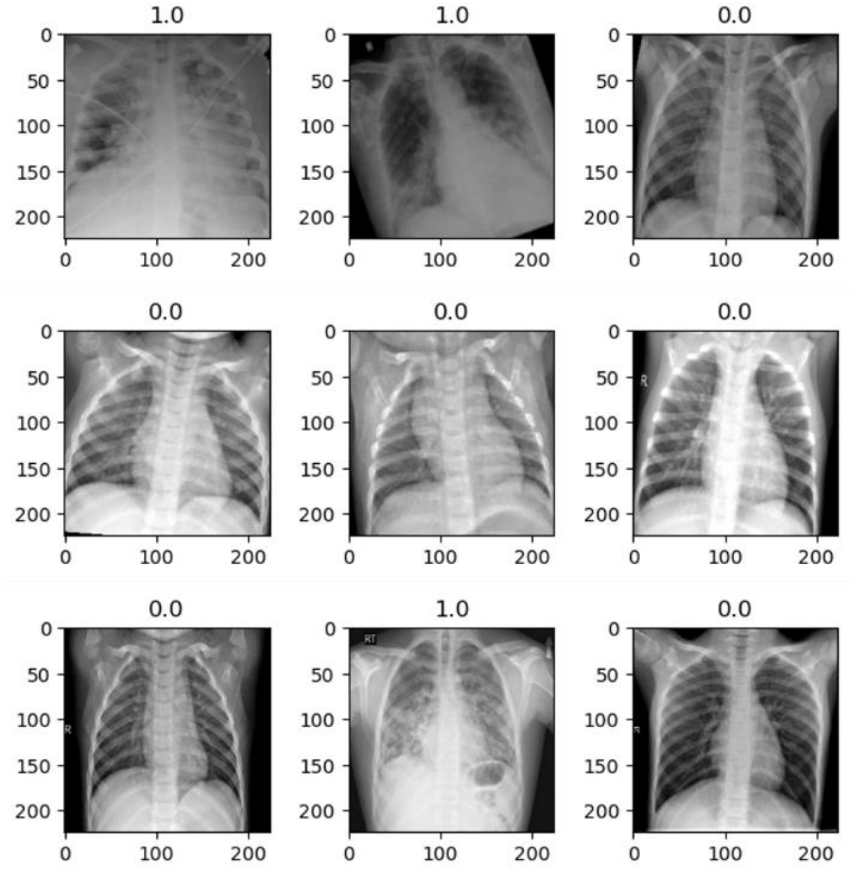

Fig. 13 Sample predicted image using CXR-DBICNet model

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

The Fig. 12 ROC Curve of the training results depicts a ROC (receiver operating characteristic) curve, which is a graphical representation of the performance of a binary classifier as the discrimination threshold is varied. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for different threshold values. We see that in the ROC curve, the top-left corner represents a classifier that makes no false positive errors and has a 1 TPR. The Fig. 13 Sample predicted image using CXR-DBICNet model provides some sample images that were successfully classified by the proposed model.

VII. MODEL COMPARISON

We compared the EfficientNetV2-based models presented by different studies with respect to their accuracy, precision, recall, and F1-score. We evaluated our proposed CXR-DBICNet against a total of five other models. Eff2Net, proposed by [23] is a deep learning model that has achieved an accuracy of 84.70% on the given task. The precision, recall, and F1-score are also reported as 84.92%, 84.70%, and 84.66%, respectively. The neural network proposed by [25] has achieved a significantly higher accuracy of 96.45%. The precision, recall, and F1-score are all reported as 96.00%. DFFCNet, proposed by [20] has achieved a remarkably high accuracy of 99.60%. The precision, recall, and F1-score are also reported as having very high values of 99.79%, 99.70%, and 99.89%, respectively. LightEfficientNetV2, proposed by [22] has achieved an accuracy of 83.42%. The precision, recall, and F1-score are reported as 81.94%, 82.76%, and 82.39%, respectively. Another deep learning model proposed by [29] has achieved a high accuracy of 96.70%. The precision, recall, and F1-score are reported as 95.00%, 66.00%, and 96.70%, respectively. Our proposed CXR-DBICNet model has achieved an accuracy of 96.93%. The precision, recall, and F1-score are reported as 97.50%, 95.00%, and 96.00%, respectively. The Fig. 14 Comparison of existing models with our proposed CXR-DBICNet model represented the detailed comparison of all the models we studied.

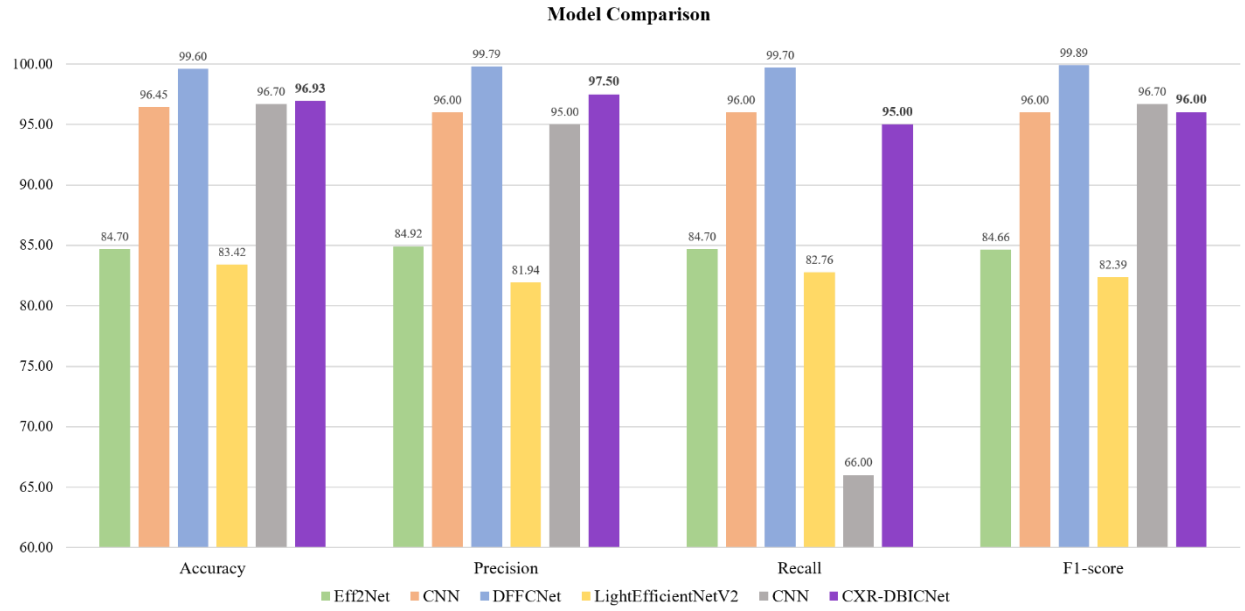

Fig. 14 Comparison of existing models with our proposed CXR-DBICNet model

Compared to the other models, CXR-DBICNet has a higher accuracy score than Eff2Net and LightEfficientNetV2. In terms of precision, CXR-DBICNet has higher value of 97.50%, indicating that it is very good at identifying true positives. DFFCNet has the highest precision score of 99.79%. The other models have precision scores ranging from 81.94% to 96.00%. Regarding recall, CXR-DBICNet has a value of 95.00%, which is lower than the recall score of DFFCNet at 99.70%. However, CXR-DBICNet has a higher recall score than Eff2Net, LightEfficientNetV2.

Looking at the F1-score, which is a harmonic mean of precision and recall, CXR-DBICNet has a score of 96.00%, which is lower than the F1-score of DFFCNet at 99.89% but higher than Eff2Net and LightEfficientNetV2. The CXR-DBICNet is a strong model with high accuracy and precision scores. However, its recall and F1-score values are not as high as those of some of the other models, such as DFFCNet, which has achieved the highest recall and F1-score values. CXR-DBICNet could be a good option when accuracy and precision are essential factors to consider for the given binary classification.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 10, Issue 7, July 2023

DOI: 10.17148/IARJSET.2023.107120

VIII. CONCLUSION

In this research study, we proposed a novel deep neural network named the chest X-ray deep binary image classification network, abbreviated as CXR-DBICNet, for identifying COVID viral illness in chest X-ray images. To make the model more effective for binary classification, additional layers are added on top of the base layer that is provided by EfficientNetV2 B0. A curated dataset of CXR images from Mendeley's data, which is often used for research and exploratory experiments, is utilized to test this proposed model. The proposed model was trained using CXR normal and COVID images before being tested with a new set of images. Testing yielded a binary accuracy of 96.93%, a precision of 97.50%, a recall of 95.00%, and an F1-score of 96.00%. A total of 4552 CXR images were used for this purpose. To validate the performance of CXR, the results were compared with five state-of-the-art models. The results show that the proposed CXR-DBICNet works better than the earlier studies that were done to find COVID from CXR images. As an outcome, our proposed approach significantly benefits society and hospitals by allowing professionals to make more rapid and precise diagnoses of COVID.

In future work, the model's performance can be improved by combining large datasets with deep learning classification models embedded with CXR-DBICNet as the base layer. This research was focused on binary classification; as a future enhancement, we could increase the classes and evaluate the model for multi-class classification.

REFERENCES

[1] C. Mulrenan, K. Rhode and B. M. Fischer, "A Literature Review on the Use of Artificial Intelligence for the Diagnosis of COVID-19 on CT and Chest X-ray," *Diagnostics*, vol. 12, no. 4, p. 869, 2022.

[2] S. Rezayi, M. Ghazisaeedi, S. R. N. Kalhori and S. Saeedi, "Artificial Intelligence Approaches on X-ray-oriented Images Process for Early Detection of COVID-19," *Journal of Medical Signals & Sensors*, vol. 12, no. 3, pp. 233-253, 2022.

[3] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi and D. Shen, "Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation, and Diagnosis for COVID-19," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 4-15, 2020.

[4] B. Yegnanarayana, Artificial Neural Networks, PHI Learning Private Limited, India, 2019.

[5] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[6] F. Chollet, Deep Learning with Python, Manning Publishing Co., 2018.

[7] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[8] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[9] H. C. Reis and V. Turk, "COVID-DSNet: A novel deep convolutional neural network for detection of coronavirus (SARS-CoV-2) cases from CT and Chest X-Ray images," *Artificial Intelligence in Medicine*, vol. 134, 2022.

[10] V. Ravi, V. Acharya and M. Alazab, "A multichannel EfficientNet deep learning-based stacking ensemble approach for lung disease detection using chest X-ray images," *Cluster Computing*, 2022.

[11] E. Luz, P. Silva, R. Silva, L. Silva, J. Guimarães, G. Miozzo, G. Moreira and D. Menotti, "Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images," *Research on Biomedical Engineering*, vol. 38, pp. 149-162, 2022.

[12] G. Marques, D. Agarwal and I. d. l. T. Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Applied Soft Computing*, vol. 96, 2020.

[13] N. Muralidharan, S. Gupta, M. R. Prusty and R. K. Tripathy, "Detection of COVID19 from X-ray images using multiscale Deep Convolutional Neural Network," *Applied Soft Computing*, vol. 119, p. 108610, 2022.

[14] G. Srivastava, A. Chauhan, M. Jangid and S. Chaurasia, "CoviXNet: A novel and efficient deep learning model for detection of COVID-19 using chest X-Ray images," *Biomedical Signal Processing and Control*, vol. 78, p. 103848, 2022.

[15] V. Ravi, H. Narasimhan, C. Chakraborty and T. D. Pham, "Deep learning-based meta-classifier approach for COVID-19 classification using CT scan and chest X-ray images," *Multimedia Systems*, vol. 28, pp. 1401-1415, 2022.

DOI: 10.17148/IARJSET.2023.107120

[16] L. T. Duong, P. T. Nguyen, L. Iovino and M. Flammini, "Automatic detection of Covid-19 from chest X-ray and lung computed tomography images using deep neural networks and transfer learning," *Applied Soft Computing*, vol. 132, p. 109851, 2023.

[17] 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, pp. 469-478, 2023.

[18] M. Chetoui, M. A. Akhloufi, B. Yousefi and E. M. Bouattane, "Explainable COVID-19 Detection on Chest X-rays Using an End-to-End Deep Convolutional Neural Network Architecture," *Big Data and Cognitive Computing*, vol. 5, no. 4, p. 73, 2021.

[19] M. O. Oba and S. Viriri, "Ensemble of EfficientNets for the Diagnosis of Tuberculosis," *Computational Intelligence and Neuroscience*, vol. 2021, p. 9790894, 2021.

[20] J. Liu, W. Sun, X. Zhao, J. Zhao and Z. Jiang, "Deep feature fusion classification network (DFFCNet): Towards accurate diagnosis of COVID-19 using chest X-rays images," *Biomedical Signal Processing and Control*, vol. 76, p. 103677, 2022.

[21] S. Kim, B. Rim, S. Choi, A. Lee, S. Min and M. Hong, "Deep Learning in Multi-Class Lung Diseases' Classification on Chest X-ray Images," *Diagnostics*, vol. 12, no. 4, p. 915, 2022.

[22] M. L. Huang and Y. C. Liao, "A lightweight CNN-based network on COVID-19 detection using X-ray and CT images," *Computers in Biology and Medicine*, vol. 146, p. 105604, 2022.

[23] R. Karthik, T. S. Vaichole, S. K. Kulkarni, O. Yadav and F. Khan, "Eff2Net: An efficient channel attention-based convolutional neural network for skin disease classification," *Biomedical Signal Processing and Control*, vol. 73, p. 103406, 2022.

[24] G. Huo, D. Lin and M. Yuan, "Iris segmentation method based on improved UNet++," *Multimedia Tools and Applications*, vol. 81, p. 41249–41269, 2022.

[25] C. K. Sunil, C. D. Jaidhar and N. Patil, "Cardamom Plant Disease Detection Approach Using EfficientNetV2," *IEEE Access*, vol. 10, pp. 789-804, 2022.

[26] I. d. Zarzà, J. d. Curtò and C. T. Calafate, "Detection of glaucoma using three-stage training with EfficientNet," *Intelligent Systems with Applications*, vol. 16, p. 200140, 2022.

[27] H. Wang, Y. Liu, N. Xu, Y. Sun, S. Fu, Y. Wu, C. Liu, L. Cui, Z. Liu, Z. Chang, S. Li, K. Deng and J. Song, "Development and validation of a deep learning model for survival prognosis of transcatheter arterial chemoembolization in patients with intermediate-stage hepatocellular carcinoma," *European Journal of Radiology*, vol. 156, p. 110527, 2022.

[28] U. Sait, G. L. KV, S. P. Prajapati, R. Bhaumik, T. Kumar, S. Shivakumar and K. Bhalla, "Curated Dataset for COVID-19 Posterior-Anterior Chest Radiography Images (X-Rays).," Mendeley Data, 2022.

[29] D. Liu, W. Wang, X. Wu and J. Yang, "EfficientNetv2 Model for Breast Cancer Histopathological Image Classification," in *2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI)*, Zhuhai, China, 2022.

BIOGRAPHY

Mr. **Suresh Mohan** received his M.Sc. (Computer Technology) degree from Anna University, Chennai, India. He is currently working as a software engineer in the IT industry. His research interests include data science and machine learning techniques.