A Study on Enhancing Pneumonia Diagnosis with Deep Learning and Transfer Learning

Parthasarathy V[1], Saravanan S[2]

[1]Research Scholar, Department of Computer and Information Science, Annamalai University, Annamalai Nagar, Chidambaram, Tamilnadu.
Email: sarathympt@gmail.com

[2]Department of Computer and Information Science, Annamalai University, Annamalai Nagar, Chidambaram, Tamilnadu.
Email: aucissaran@gmail.com

Abstract:
Pneumonia, an infection affecting the lungs, can result from bacteria, viruses, and other microorganisms. Particularly difficultfor vulnerable populations like the elderly and individuals with compromised immune systems, pneumonia demands accurate and timely diagnosis to avert complications and enhance patient outcomes. It involves pre-training a model on a comprehensive dataset and fine-tuning it for a specific task using a smaller, task-specific dataset.Transfer learning has been effectively applied to diagnosing pneumonia using chest X-ray images, offering encouraging results. For instance, a study published in the Radiology journal in 2017 harnessed a convolutional neural network (CNN) trained on a substantial chest X-ray dataset, achieving an impressive AUC (area under the curve) score for accurately categorizing images as usual or pneumonia affected.In conclusion, applying deep learning and transfer learning to pneumonia diagnosis using chest X-ray datasets holds great promise in augmenting accuracy and efficiency, ultimately benefiting both patients and healthcare systems.

Keywords: Machine Learning, Deep Learning, CNN, Transfer Learning, Chest X-Ray Images.

DOINUMBER:10.48047/NQ.2022.20.19.NQ99448 NEUROQUANTOLOGY2022;20(19):4860-4869

1. INTRODUCTION AND LITERATURE REVIEW

Pneumonia, a prevalent respiratory infection, arises from diverse pathogens such as bacteria, viruses, and other microorganisms. It poses a significant threat to the health and well-being of vulnerable populations, including the elderly and those with compromised immune systems. Timely and precise diagnosis of pneumonia is pivotal in preventing complications and improving patient outcomes.Historically, the diagnosis of pneumonia relied on clinical symptoms, physical examinations, and radiographic assessments like chest X-rays. However, these methods often introduced subjectivity and failed to consistently deliver accurate results.

The integration of deep learning and machine learning techniques offers a promising solution to enhance the accuracy and efficiency of pneumonia diagnosis. These methodologies involve training models on large sets of labeled images, enabling the

recognition of patterns and features indicative of pneumonia. A widely adopted approach is transfer learning, which entails pre-training a model on a comprehensive dataset and subsequently fine-tuning it using a smaller, task-specific dataset.

Transfer learning has been effectively employed in the detection of pneumonia using chest X-ray images, yielding encouraging outcomes. Notably, a study published in the Radiology journal in 2017 leveraged a convolutional neural network (CNN) trained on an extensive chest X-ray dataset, achieving an impressive AUC score of 0.97 for the accurate classification of images as normal or pneumonia affected.In summary, the application of deep learning and transfer learning in pneumonia diagnosis using chest X-ray datasets holds significant promise in bolstering accuracy and efficiency, ultimately benefiting patients and healthcare systems [1] to [5].

The evaluation of the models involved computing the mean average precision (mAP) at various intersection-over-union (IoU) thresholds [6]. We tested several different encoder architectures, including Xception [7], NASNet-A-Mobile [8], ResNet-34, -50, -101 [9], SE-ResNext-50, -101 [10], DualPathNet-92 [11], Inception-ResNet-v2 [12], and PNASNet-5-Large [13]. To ensure efficient experiments and model iterations, we focused on architectures that strike a good balance between accuracy and complexity/parameters number, resulting in faster training times [14]. VGG nets [15] and MobileNets [16] did not perform as well on the ImageNet dataset [17] in terms of accuracy. On the other hand, SeNet-154 [11] and NasNet-A-Large [9] had the highest number of parameters and required the most floating-point operations. The SE-ResNext architectures demonstrated optimal performance on this dataset, offering a good compromise between accuracy and complexity [14]. For the RSNA Pneumonia Detection Challenge, our model was based on the RetinaNet implementation in PyTorch [18]. However, we made several modifications to the original implementation to enhance its performance and suitability for our specific task.

Dataset

The "Lung Infection in Chest X-ray Images (Kaggle)" dataset comprises over 5,863 chest X-ray images, including a substantial number depicting pneumonia cases. Originating as part of a Kaggle competition, this dataset has found widespread application in research endeavors. These datasets collectively offer a diverse range of chest X-ray images, serving as valuable resources for the training and evaluation of models designed for pneumonia detection.

The dataset is structured into three folders (train, test, val), each featuring subfolders for image categories (Pneumonia and Normal). In total, it comprises 5,863 X-ray images in JPEG format, categorized into two classes: Pneumonia and Normal.

Anterior-posterior chest X-ray images were drawn from retrospective cohorts of pediatric patients aged one to five years, sourced from Guangzhou Women and Children's Medical Center, Guangzhou. A comprehensive analysis of these chest X-ray images entailed an initial quality control phase, during which low-quality or unreadable X-ray images were systematically eliminated. Subsequently, the diagnostic assessments for these images underwent a thorough grading process, conducted by two expert physicians, before they were deemed suitable for training in the AI system. For an additional layer of quality assurance, a third expert reviewed the evaluation set to validate the grading accuracy.

Fig-1: Normal CXR Images

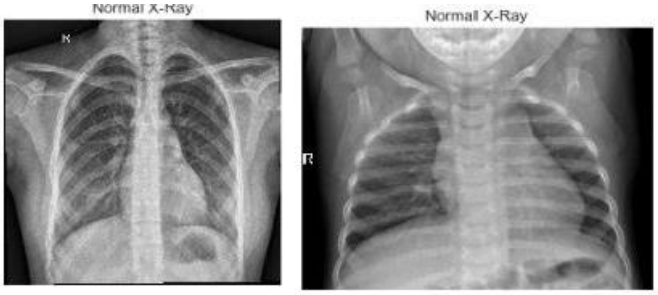
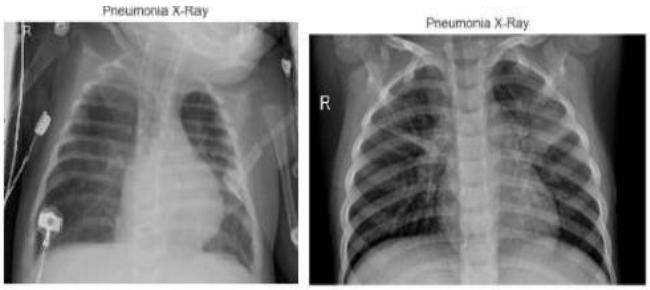


Fig-2: Pneumonia Affected CXR Images



2. METHODOLOGY

The methodology for employing machine learning and deep learning techniques to detect and predict pneumonia varies depending on the specific approach and data sources employed. Presented below is a general outline of the procedural steps involved in this process:

- **Data Collection:** The initial phase involves the acquisition of a dataset comprising chest X-ray images, encompassing both normal and pneumonia-affected images. These datasets can be sourced from clinical settings, hospitals, or online repositories like the Kaggle Chest X-ray dataset.
- **Data Preprocessing:** Subsequently, data preprocessing becomes pivotal. This step involves selecting a subset of images for model training and testing, resizing, cropping, and addressing any data errors or biases.
- **Feature Extraction:** This step involves the extraction of pertinent features from the images, those that are directly linked to pneumonia detection. These features may include patterns and shapes within lung tissue, anomalies in the appearance of the heart and blood vessels, and other indicative characteristics.
- **Model Training:** The subsequent step is the training of a machine learning or deep learning model using the dataset. This encompasses the selection of an appropriate model architecture, such as a Convolutional Neural Network (CNN) or Random Forest Classifier, and the tuning of relevant hyperparameters like learning rates and regularization strength.
- **Model Evaluation:** Once the model is trained, a critical evaluation of its performance on a separate test dataset is essential to assess accuracy and generalizability. Performance metrics, including accuracy, precision, recall, and the area under the curve (AUC), are often calculated.
- **Model Deployment:** Should the model demonstrate strong performance, it can be deployed in a clinical context to aid in pneumonia diagnosis. This deployment could involve integration into a computer-aided diagnosis system or the use of the model to generate a probability score that assists in decision-making.

In summary, the methodology for pneumonia detection employing machine learning and deep learning encompasses a series of steps that require careful consideration and optimization to achieve optimal performance.

3. Models

Various machine learning and deep learning models have been applied to pneumonia detection. The following methods have been utilized for this purpose:

3.1 Convolutional Neural Networks (CNNs)

CNNs, tailored for image classification tasks, comprise multiple layers of interconnected nodes trained to identify patterns and features in images. CNNs have demonstrated effectiveness in pneumonia detection across several studies.

3.2 DenseNet

DenseNet, a type of CNN, has been employed for image classification, including pneumonia detection. Its distinctive feature is dense connectivity, wherein each layer connects to all preceding layers, facilitating efficient learning and reducing overfitting.

3.3 VGG-16

VGG-16, known for its use of small, 3x3 convolutional filters and a large number of layers, excels in capturing fine-grained details in images. It has demonstrated strong performance in pneumonia detection from chest X-ray images.

3.4 ResNet

ResNet has been utilized for diverse image classification tasks, including pneumonia detection. Its unique feature, residual connections, enhances learning efficiency and mitigates overfitting. Its deep architecture allows it to identify complex patterns and features in data.

3.5 InceptionNet

InceptionNet, recognized for its use of inception modules, efficiently learns multiple scales and sizes of features in data. Its relatively shallow architecture, compared to other CNNs, renders it efficient and easier to train. It has shown promise in pneumonia detection using chest X-ray images.

Careful model selection and performance evaluation are imperative to determine the most suitable approach for specific datasets and tasks.

Evaluation Metrics

The TP Rate, or True Positive Rate, is a frequently employed performance metric in binary classification scenarios, common in fields like machine learning and statistics. It's also recognized as Sensitivity, Recall, or Hit Rate. These metric gauges the proportion of actual positive instances that a model or classifier correctly identifies as positive. Its calculation involves the formula:

Similarly, the FP Rate, or False Positive Rate, is another vital performance measure in binary classification tasks within fields such as machine learning and statistics. It's also referred to as the False Alarm Rate. This metric assesses the proportion of actual negative instances that a model or classifier inaccurately categorizes as positive. The calculation employs the formula:

On the other hand, Precision is a performance metric employed in binary classification scenarios, providing an assessment of a model's ability to make accurate positive predictions. It's particularly valuable when the cost of false positives is substantial or when there's a need for high reliability in positive predictions.

Recall, also known as Sensitivity or True Positive Rate, is a performance metric designed to quantify a model's capability to accurately identify all positive instances within a dataset.

Lastly, the F-Measure, often referred to as the F1 Score, is a performance metric tailored for binary classification tasks, aiming to strike a balance between precision and recall. It comes in handy when dealing with imbalanced class distributions and when you want to evaluate a model's performance while considering both false positives and false negatives.

TP Rate = (True Positives) / (True Positives + False Negatives) ... (1)
FP Rate = (False Positives) / (False Positives + True Negatives) ... (2)
Precision = (True Positives) / (True Positives + False Positives) ... (3)
Recall = (True Positives) / (True Positives + False Negatives) ... (4)

Where True Positives (TP) are the instances that are actually positive and are correctly predicted as positive by the model. False Negatives (FN) are the instances that are actually positive but are incorrectly predicted as negative by the model. False Positives (FP) are the instances that are actually negative but are incorrectly predicted as positive by the model. True Negatives (TN) are the instances that are actually negative and are correctly predicted as negative by the model.

F1 Score = 2 * (Precision * Recall) / (Precision + Recall) ... (5)

Where Precision is the ratio of true positives to all instances predicted as positive. Recall is the ratio of true positives to all actual positive instances.

MCC, short for Matthews Correlation Coefficient, serves as a widely adopted performance metric, especially valuable for assessing the effectiveness of binary classification models, especially in scenarios with imbalanced datasets. It offers a holistic assessment by considering true positives,

true negatives, false positives, and false negatives. The computation of the Matthews Correlation Coefficient follows this formula:

MCC = (TP * TN - FP * FN) / √((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))... (6)

The MCC value ranges from -1 to +1. A higher MCC indicates a better overall performance of the model. An MCC of +1 represents a perfect prediction, an MCC of 0 suggests that the model's predictions are no better than random, and an MCC of -1 indicates a completely inverse relationship between the prediction and the actual values.

The ROC (Receiver Operating Characteristic) is a visual representation that assesses a binary classification model's performance across different threshold settings. Its primary purpose is to analyze and contrast the trade-offs between a model's True Positive Rate (sensitivity) and its False Positive Rate (1 - specificity). The ROC curve itself materializes by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at varying threshold levels, revealing how adeptly a model distinguishes between positive and negative classes across diverse decision thresholds.The True Positive Rate, also referred to as Sensitivity or Recall, quantifies the ratio of correctly identified positive cases concerning all actual positives:

TPR = TP / (TP + FN). ... (7)

False Positive Rate (FPR), also known as Fall-out, is the ratio of false positives to all actual negatives:

FPR = FP / (FP + TN) ... (8)

An exemplary ROC curve is typically depicted as TPR (y-axis) versus FPR (x-axis) and generally originates from the origin (0, 0), advancing upward. The diagonal line represents a random classifier with no predictive power, while an effective classifier's ROC curve endeavors to maximize its distance from this diagonal, preferably reaching the upper-left corner of the chart.

The Area Under the ROC Curve (AUC-ROC) serves as a prominent summary statistic for the ROC curve. Models with AUC-ROC values close to 1 exhibit superior discrimination ability, while those near 0.5 are no more effective than random chance. This metric facilitates model comparisons, where a higher AUC-ROC signifies an enhanced overall performance.

The PRC (Precision-Recall Curve) is a graphical portrayal of a binary classification model's effectiveness, with a specific emphasis on the balance between precision and recall under varying threshold conditions. This visualization proves especially valuable in scenarios with imbalanced datasets, where one class significantly outweighs the other. To construct a PRC, consider the following steps:

1. Vary the classification threshold of your model to obtain different sets of predictions.
2. For each threshold setting, calculate True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN).
3. Calculate precision and recall for each threshold setting:
   Precision = TP / (TP + FP)
   Recall (Sensitivity) = TP / (TP + FN)
4. Plot precision on the y-axis and recall on the x-axis for each threshold setting. Connect the data points to form the PRC.

An ideal classifier's Precision-Recall Curve (PRC) would commence at the lower-left point (0, 0) and ascend to the upper-right point (1, 1). In real-world applications, the PRC is frequently employed alongside the AUC-PRC (Area Under the Precision-Recall Curve) to derive a unified performance metric encapsulating the model's overall effectiveness across various thresholds. A higher AUC-PRC signifies superior performance. Equations1 to 8 are used to find the model accuracy, which is used to find the model performance and error.

Table 1:CNNEvaluationMetrics

Metric	Precision	recall	F1-score	support
Pneumonia	0.9400	0.9500	0.9200	370
Normal	0.9200	0.9000	0.9100	204
Micro average	0.9400	0.9400	0.9400	644
Macro average	0.9400	0.9300	0.9300	644

Weighted average	0.9400	0.9400	0.9400	644

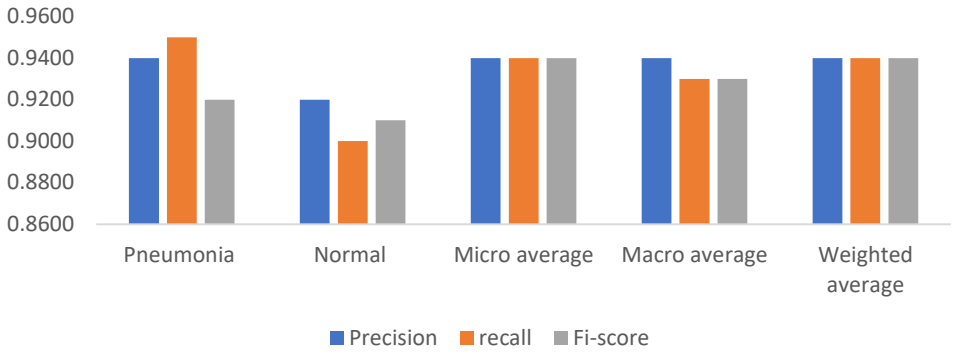


Fig. 3 (a).CNNEvaluationMetrics

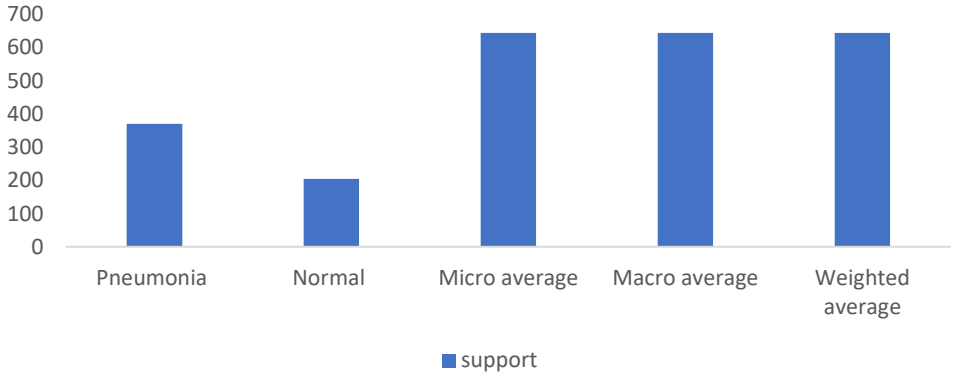


Fig. 3 (b).CNNEvaluationMetrics

Table 2:CNN_2EvaluationMetrics

Metric	Normal	Pneumonia	Micro Average	Macro average	Weighted average
Precision	0.5675	0.9195	0.7949	0.8439	0.8881
Recall	0.9169	0.6777	0.7949	0.8474	0.7949
F1-score	0.7995	0.7993	0.7944	0.7949	0.7959
Support	244	400	644	644	644

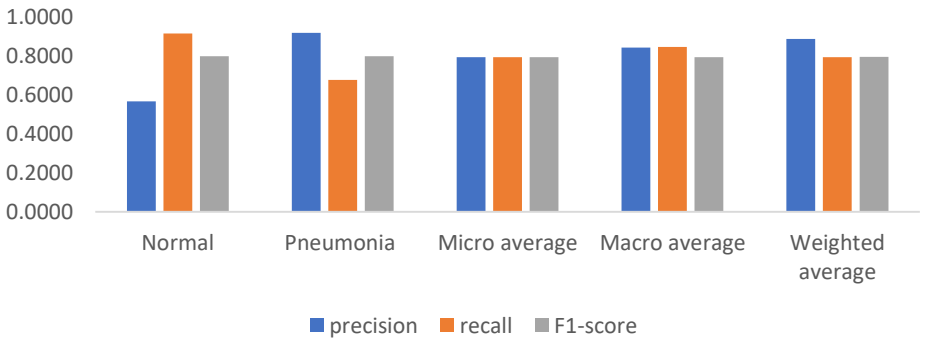


Fig. 4 (a).CNN_2EvaluationMetrics

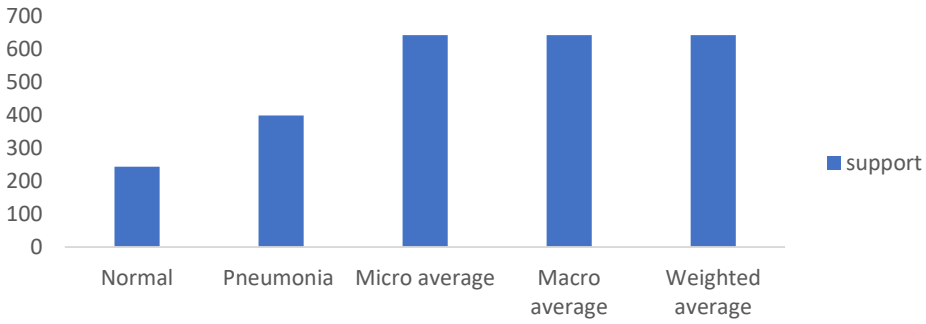


Fig. 4 (b).CNN_2EvaluationMetrics

4. RESULTS AND DISCUSSIONS

Within this section, our objective is to delve into the classification performance by scrutinizing key metrics, including accuracy and loss. Numerous prior studies have conducted comprehensive analyses on the application of machine learning for pneumonia detection. In the broader context, these investigations have unveiled encouraging outcomes, with machine learning models consistently exhibiting robust accuracy in discerning pneumonia from medical images.In our endeavor, we endeavor to visually represent the Training and Validation Accuracy by plotting them on a graph, with accuracy along the y-axis and epochs along the x-axis. The resulting performance insights for various models are outlined as follows:

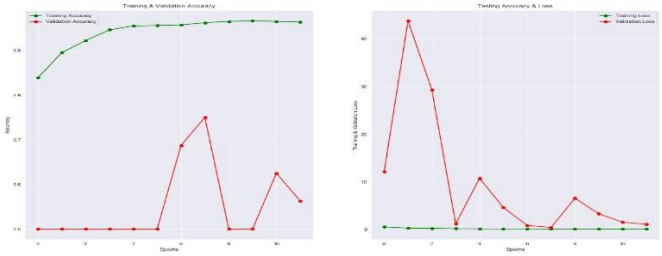


Fig.5. CNNmodel

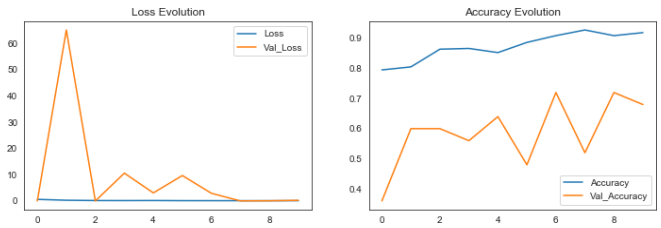


Fig. 6. CNN_2model

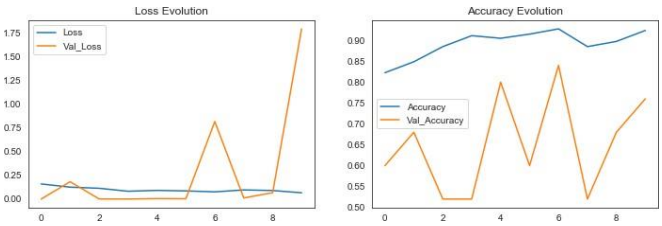


Fig. 7. DenseNetmodel

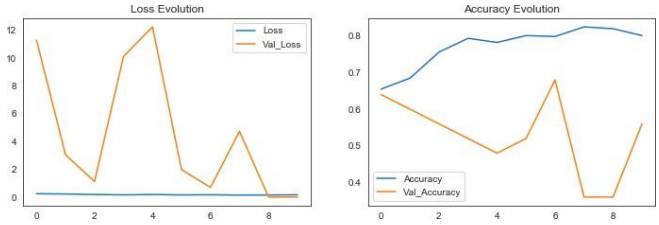


Fig. 8. VGG-16model

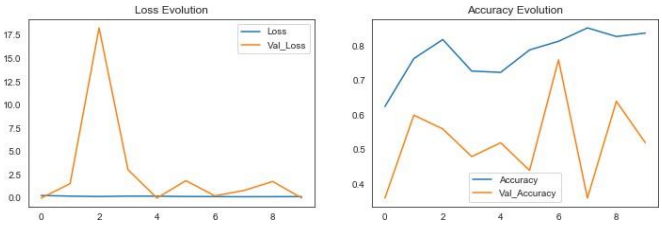


Fig. 9. ResNetmodel

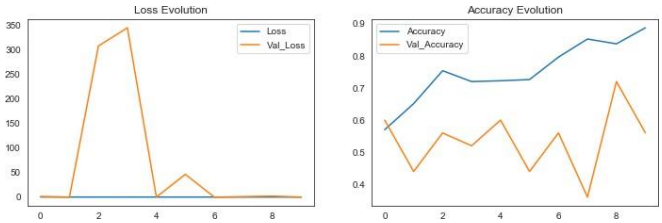


Fig. 10. InceptionNetmodel

5. Comparative Analysis of Varied Models

This section is dedicated to comparing distinct preprocessed models, gauging their performance through metrics like accuracy and loss. It's important to note that the optimal choice for pneumonia detection hinges on the unique attributes of the dataset and the desired performance criteria. Exploring several different models might be necessary to pinpoint the most effective one.Our examination involves a comparison of these diverse models in terms of both testing and training accuracy. The accuracy scores are outlined as follows:

Table 3:Accuracyofallmodelstested(in%)

Models	**Accuracy**
CNN	95.9900
CNN_2	72.9200
DenseNet	91.1900
VGG16	70.2000
ResNet	77.4100
InceptionNet	80.7700

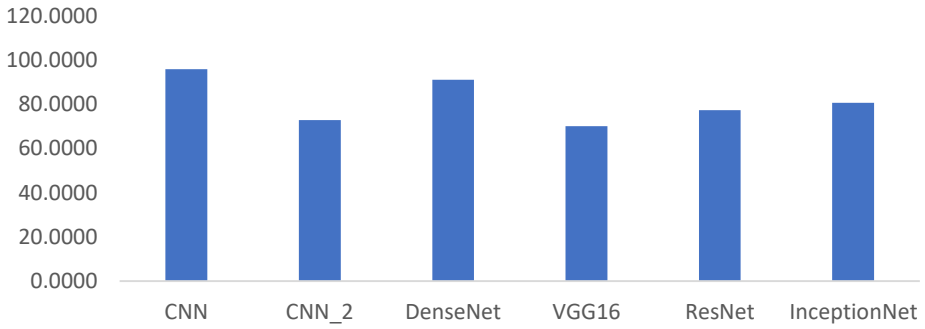


Fig-11:Accuracyofallmodelstested(in%)

6. CONCLUSIONS AND FURTHER RESEARCH

This research introduced a different model and was designed for identifying pneumonia in chest X-ray images. These models have been meticulously crafted from the ground up, relying primarily on transfer learning and CNN architectures. Nonetheless, it is crucial to acknowledge certain limitations when deploying CNNs for pneumonia detection.

Based on Table 1 and Figure 3, CNNs for pneumonia detection return the best performance for precision, recall, and F1-score. In this research takes into consideration five different metrics, namely Pneumonia, Normal, Micro, Macro, and Weighted average. All five metrics return solid positive performance.

CNN_2EvaluationMetrics return less performance compared to the CNN approach. The related results are shown in Table 2 and Figure 4. The accuracyofallmodelstested is based on various metrics: the CNN approach returns nearly 96%, and the remaining model returns moderate performance, 72% to 91%. In this case, VGG16 returns 70% accuracy. Related results are shown in Table 3 and Figure 11.

The necessity for a substantial volume of annotated data to effectively train the model a process that can be resource-intensive and time-consuming. As we look to the future, there is a pressing need for further research to gain deeper insights into the strengths and constraints of CNNs in the context of pneumonia detection. Such efforts should also pinpoint the most productive approaches to different dataset types.

7. REFERENCES

[1]. KaggleDatasetaccessedon10October2022:https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

[2]. AlomMZ,HasanM,IslamMT,etal.AutomaticpneumoniadetectionfromchestX-rayimagesusingadeep convolutional neural network. In 2018 InternationalConference on Informatics, Electronics and Vision (ICIEV)(pp.1-6).IEEE,2018

[3]. Han B,KimY,KimH,Lee S.Adeeplearning-basedapproachfordetectingpneumoniafromchestX-rays.ComputersinBiologyand Medicine,98:58-64,2018.

[4]. TulabandhulaS,MehtaK,ElgendyIY,etal. Deeplearning-basedautomateddetectionofpneumoniafromchest radiographs. Journal of Medical Systems, 43(2): 31,2019.

[5]. Li Q, Zhang L, Chen M, et al. Automated detection ofpneumonia in chestX-ray images using a deep learningmodel. Radiology,291(3):673-681,2019.

[6]. www.kaggle.com/c/rsna-pneumoniadetection-challenge/overview/evaluation, 2018.

[7]. Franc¸oisChollet. Xception: Deep learning with depth wise separable convolutions, 2016.

[8]. BarretZoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition, 2017.

[9]. Kaiming He, Xiangyu Zhang, ShaoqingRen, and Jian Sun. Deep residual learning for image

recognition, 2015.
[10]. Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[11]. Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 4467–4475. Curran Associates, Inc., 2017.
[12]. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.
[13]. Barry Kelly. The chest radiograph. Ulster Med J, 2012.
[14]. Simone Bianco, RemiCadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. IEEE Access, 6:64270–64277, 2018.
[15]. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.
[16]. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.
[17]. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.
[18]. Chenxi Liu, BarretZoph, Maxim Neumann, Jonathon Shlens,Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search, 2017.