## PNEUMONIA DETECTION USING X-RAY IMAGES

Submitted by:

**ROSE JIJO**

**ABSTRACT**

Pneumonia is a severe respiratory infection that affects millions of people worldwide, leading to significant morbidity and mortality. Early and accurate detection of pneumonia plays a crucial role in guiding appropriate treatment and improving patient outcomes. Chest X-ray imaging is one of the primary diagnostic tools for pneumonia detection due to its accessibility and ability to visualize lung abnormalities.

In recent years, there has been a surge of research focusing on developing automated systems for pneumonia detection using deep learning and machine learning techniques. These approaches leverage the power of convolutional neural networks (CNNs) to analyze chest X-ray images and provide accurate and efficient detection of pneumonia. By training CNN models on large-scale datasets, researchers have achieved remarkable results, often surpassing the performance of human radiologists.

In conclusion, pneumonia detection using X-ray images has witnessed significant advancements due to the application of deep learning and machine learning techniques. The automated systems developed hold immense potential to revolutionize pneumonia diagnosis and improve patient care. Continued research and collaboration among clinicians, researchers, and technologists are vital to further enhance the accuracy, reliability, and clinical integration of these systems.

**INTRODUCTION**

Pneumonia is a common and potentially life-threatening respiratory infection that affects the lungs. Pneumonia detection using X-ray images plays a crucial role in early diagnosis and treatment of this condition. In recent years, there have been significant advancements in leveraging machine learning and deep learning techniques for accurate and efficient pneumonia detection.

The use of Convolutional Neural Networks (CNNs) has proven to be highly effective in analyzing X-ray images and detecting pneumonia. CNNs are well-suited for image recognition tasks, as they can automatically learn and extract meaningful features from the input images. By employing multiple convolutional layers, pooling layers, and fully connected layers, CNNs can effectively capture the spatial dependencies and patterns present in X-ray images associated with pneumonia. Transfer learning has emerged as a powerful technique in pneumonia detection. Pretrained CNN models, trained on large-scale datasets such as ImageNet, can be fine-tuned and adapted to the task of pneumonia detection. This approach allows the model to leverage the learned representations and features from the pretrained model, enabling better generalization and performance, especially when dealing with limited annotated data. Data augmentation techniques are commonly employed to address the challenges of limited datasets. Augmentation techniques such as rotation, scaling, flipping, and adding noise can artificially increase the size and diversity of the training dataset. This helps in improving the model's ability to generalize and reduces the risk of overfitting.

The combination of advanced deep learning techniques, transfer learning, data augmentation, and interpretability methods has led to significant improvements in pneumonia detection using X-ray images. These advancements have the potential to assist healthcare professionals in making accurate and timely diagnoses, enabling early intervention and improving patient outcomes. Continued research and development in this field hold the promise of further enhancing the accuracy, efficiency, and accessibility of pneumonia detection systems, ultimately benefiting patients and healthcare providers alike.

The purpose of this project is to develop a deep learning model to detect pneumonia from x-ray images. The model is trained using a dataset of x-ray images that depict pneumonia or normal cases. The model is trained using the transfer learning approach, where a pre-trained convolutional neural network (CNN) is used as the base model, and the last few layers of the CNN are replaced with new layers that are trained on the dataset.

**LITERATURE REVIEW**

Pneumonia detection using X-ray images is an important area of research in medical imaging and computer-aided diagnosis. Numerous studies have been conducted to develop effective methods for pneumonia detection using X-ray images.

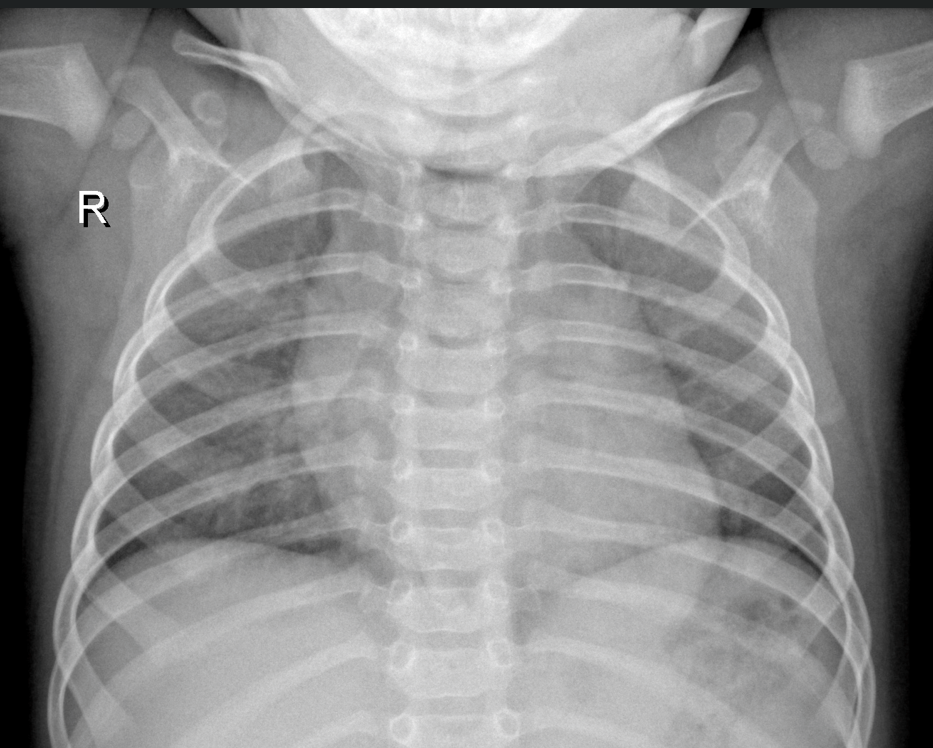
1. "Automated Pneumonia Detection in Chest X-Rays from Portable Chest X-Ray Devices" by Kumar et al. (2018): The study proposed an automated approach for pneumonia detection in chest X-ray images acquired from portable X-ray devices. The authors developed a deep learning model and evaluated its performance on a dataset of portable chest X-ray images.
2. "Machine Learning and Deep Learning Models for Pneumonia Detection: A Review" by Alqudah et al. (2021): This review paper provided an overview of machine learning and deep learning models used for pneumonia detection, including those based on chest X-ray images. It discussed different methodologies, dataset characteristics, and performance metrics used in pneumonia detection studies. The authors also highlighted the limitations and potential solutions in this area.
3. "Deep Learning-Based Classification and Visualization of Pneumonia Using Chest X-Ray Images" by Islam et al. (2019): This study focused on developing a deep learning-based classification model for pneumonia detection and visualization using chest X-ray images. The authors proposed a model that combined convolutional neural networks (CNNs) with attention mechanisms to achieve accurate classification.
4. "Attention-Guided Curriculum Learning for Pneumonia Detection from Chest X-Ray" by Guan et al. (2019): The research introduced an attention-guided curriculum learning framework for pneumonia detection from chest X-ray images. The proposed method incorporated attention mechanisms to focus on informative regions in X-ray images and utilized a curriculum learning strategy for efficient model training.
5. "Deep Ensemble Learning for Pneumonia Detection in Chest X-Rays" by Bustos et al. (2020): This study investigated the use of deep ensemble learning for pneumonia detection in chest X-ray images. The authors trained multiple deep learning models and combined their predictions to improve the accuracy and robustness of the pneumonia detection system.
6. "Deep Learning for Pneumonia Detection in Chest Radiographs: A Systematic Review and Meta-Analysis" by Islam et al. (2020): This systematic review and meta-analysis examined the effectiveness of deep learning models for pneumonia detection in chest X-ray images. The authors analyzed various studies and evaluated the performance of different deep learning architectures, datasets, and evaluation metrics.

These papers represent a sample of the research conducted on pneumonia detection using X-ray images. They highlight the application of deep learning techniques and the development of specialized models to improve the accuracy and efficiency of pneumonia detection systems. Further research in this field continues to explore advanced algorithms, interpretability, and integration with clinical decision support systems to aid radiologists in accurate diagnosis and treatment planning.

**PROCEDURE**

1. **Data Collection:** In this project, we will be using the [Chest X-Ray Images (Pneumonia)](https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia). This dataset consists of chest X-Ray images for both healthy and pneumonic type. The dataset consists of 3 folders, test, train, and validation each containing images belonging to two classes, normal and pneumonia.

A x-ray of a person's chest

Description automatically generated 

i) normal ii) pneumonia

1. **Data Preparation:** The dataset is preprocessed using data augmentation techniques such as resizing, cropping, normalization, etc., to improve the model’s performance. The `transformers` dictionary defines different sets of image transformations used for data augmentation and preprocessing in the context of training, testing, and validation.

1. ‘train\_transforms’: This set of transformations is applied to the training dataset. It includes the following steps:

- ‘transforms.Resize((224, 224))’: Resizes the image to a square shape of size 224x224 pixels. This ensures that all images have the same dimensions as required by the model.

- ‘transforms.RandomRotation(20)’: Randomly rotates the image by a maximum of 20 degrees in a counter-clockwise direction. This helps to introduce diversity in the training data and makes the model more robust to variations in orientation.

- ‘transforms.RandomHorizontalFlip()’: Randomly flips the image horizontally with a probability of 0.5. This simulates different perspectives and orientations of the objects in the image.

- ‘transforms.ToTensor()’: Converts the image to a tensor, which is the standard input format for deep learning models.

- ‘transforms.Normalize([0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])’: Normalizes the image by subtracting the mean [0.485, 0.456, 0.406] and dividing by the standard deviation [0.229, 0.224, 0.225]. This normalization step helps to standardize the pixel values and improve the model's convergence during training.

2. ‘test\_transforms’: This set of transformations is applied to the testing dataset. It includes the following steps:

- ‘transforms.Resize((224, 224))’: Resizes the image to a square shape of size 224x224 pixels, similar to the training dataset.

- ‘transforms.CenterCrop(224)’: Crops the center portion of the image to a square shape of size 224x224 pixels. This ensures consistent input size for testing and evaluation.

- ‘transforms.ToTensor()’: Converts the image to a tensor.

- ‘transforms.Normalize([0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])’: Normalizes the image using the same mean and standard deviation values as in the training dataset.

3. ‘valid\_transforms’: This set of transformations is applied to the validation dataset. It follows the same steps as the testing dataset, including resizing, center cropping, converting to a tensor, and normalization.

By defining these sets of transformations, we can ensure that the input images are consistently preprocessed and augmented during the different stages of the pipeline. This helps to improve the model's performance and generalization ability by introducing variations and ensuring data consistency across the training, testing, and validation phases.

Upsampling and downsampling are techniques used to address class imbalance in a dataset. Class imbalance refers to a situation where the number of samples in one class significantly outweighs the number of samples in another class. In the context of pneumonia detection, it means that there is a significant difference in the number of samples belonging to the "NORMAL" class and the "PNEUMONIA" class.

Class imbalance can pose challenges during the training of machine learning models. Models tend to be biased towards the majority class and may struggle to accurately learn patterns and make predictions for the minority class. This is because the model's objective is to minimize the overall error, and it may prioritize achieving high accuracy on the majority class while neglecting the minority class.

To address this issue, upsampling and downsampling techniques are used:

1. Upsampling: Upsampling involves randomly replicating samples from the minority class to increase their representation in the dataset. By creating additional synthetic samples, the upsampling technique helps to balance the class distribution and provide the model with more examples of the minority class. This can improve the model's ability to learn patterns and make accurate predictions for the minority class.

2. Downsampling: Downsampling involves randomly removing samples from the majority class to reduce its dominance in the dataset. By reducing the number of samples in the majority class, downsampling helps to balance the class distribution and prevent the model from being overly biased towards the majority class. This encourages the model to pay equal attention to both classes during training and improve its ability to generalize to new, unseen data.

In this case:

* Upsampling the 'NORMAL' class: The indices of samples belonging to the 'NORMAL' class are identified using `np.where(train\_targets == 0)[0]`. The `resample()` function from sklearn.utils is then used to upsample the 'NORMAL' class by generating synthetic samples. The `replace=True` parameter indicates that sampling is performed with replacement, and `n\_samples` is set to double the number of original 'NORMAL' samples (`len(normal\_indices) \* 2`). The `random\_state` parameter ensures reproducibility.
* Downsampling the 'PNEUMONIA' class: Similarly, the indices of samples belonging to the 'PNEUMONIA' class are identified using `np.where(train\_targets == 1)[0]`. The `resample()` function is called with `replace=False` to perform downsampling without replacement. The number of samples in the 'PNEUMONIA' class is halved (`len(pneumonia\_indices) // 2`).

The combination of upsampling and downsampling helps to create a balanced dataset where both classes are represented more equally. This can lead to better model performance, especially in scenarios where class imbalance is significant.

1. **Model Architecture:** A Convolutional Neural Network (CNN) is a specialized type of neural network that excels in image recognition tasks. It comprises multiple layers of convolutions, which apply filters to extract meaningful features from the input image. These features are then fed into fully connected layers for classification. For this project, a ResNet18 CNN architecture is utilized, pre-trained on the extensive ImageNet dataset. The ResNet18 model's final layers are replaced and retrained on the hand gesture dataset.

ResNet18 is a deep CNN architecture developed by Microsoft, with "Residual Network" referring to its use of residual connections. These connections allow the network to pass information from previous layers directly to subsequent layers, facilitating the learning of complex functions and enhancing gradient flow during training. ResNet18 comprises 17 convolutional layers and one fully connected layer. Various filter sizes are employed in the convolutional layers to extract features. The incorporation of skip connections further enhances the model's performance.

Transfer learning is a technique that leverages pre-trained models for solving new problems instead of training from scratch. By utilizing the learned features and weights of a pre-trained model on a large dataset, we can expedite training and benefit from previous knowledge. Transfer learning is particularly advantageous when working with limited datasets or when computational resources are limited. It allows us to build upon the expertise of the machine learning community by utilizing established models that have undergone extensive fine-tuning.

In this project, the model is trained using the cross-entropy loss function and the stochastic gradient descent (SGD) optimizer. The training is performed for 4 epochs with a batch size of 5. These choices are made to optimize the model's performance on the dataset and achieve accurate predictions.

1. **Model Training:** Training a model involves updating the model's parameters using the input data to minimize the loss function. In the case of deep learning, this is typically done through a process called backpropagation, which involves computing gradients of the loss with respect to the model's parameters and updating them in the direction that reduces the loss.

Here, we are getting the number of input features to the last fully connected layer of the pre-trained ResNet18 model. This will be used later to create a new fully connected layer for our specific classification task. We are replacing the original fully connected layer of the ResNet18 model with a new fully connected layer that has 2 output nodes, one for each class. The gradients of the loss with respect to the model's parameters are computed using backpropagation. These gradients are then used to update the parameters. The parameters of the model are updated using an optimization algorithm such as stochastic gradient descent (SGD). This step involves adjusting the parameters in the direction that reduces the loss. We are defining the optimizer that will be used to update the model weights during training. We are using the SGD optimizer and setting the learning rate to 0.001. A momentum value of 0.9, as given in the code, means that the optimizer considers 90% of the accumulated gradient from the previous iteration and only 10% of the current gradient to determine the update step.

During backpropagation using the stochastic gradient descent (SGD) optimizer, the gradients of the loss function with respect to each weight in the neural network are calculated using the chain rule of differentiation. These gradients represent the direction and magnitude of the update needed to minimize the loss.

The SGD optimizer updates the weights by subtracting a small multiple of the gradient from the current value of each weight. This small multiple is determined by the learning rate, which is a hyperparameter specified by the user. The learning rate controls the step size taken during weight updates and influences the convergence speed of the optimization algorithm.

In SGD, the learning rate is fixed and does not change throughout the training process. This means that the same learning rate is applied to all weights and during every update.

Cross-entropy loss is a widely used loss function in machine learning, specifically in multi-class classification problems. It quantifies the dissimilarity between the predicted probability distribution and the true probability distribution of the classes. In PyTorch, the 'nn.CrossEntropyLoss()' combines 'nn.LogSoftmax()' and 'nn.NLLLoss()' into a single class. For each input sample, the cross-entropy loss calculates the error for each possible class label and sums them together. It measures how well the predicted probabilities align with the true labels. The objective of training a model using this loss function is to minimize the loss value, as a lower loss indicates a better alignment between the predicted and true distributions. The output of the cross-entropy loss is a single scalar value, representing the total loss of the model on a given set of inputs. By minimizing this loss during training, the model learns to improve its classification performance, aiming to accurately assign the correct class labels to the input data.

1. **Validation:** Validation is a critical step in the training process, aimed at evaluating how well a trained model can generalize to new and unseen data. By using a separate dataset that is distinct from the training data, we can assess the model's performance on unseen examples. The main goal of validation is to ensure that the model's performance is not solely optimized for the training data but extends well to new data. It allows us to fine-tune the model's hyperparameters, such as learning rate or regularization strength, based on its performance on the validation dataset. Monitoring the model's performance on the validation dataset provides valuable insights into its ability to generalize. It helps us identify potential issues like overfitting or underfitting, where the model may either memorize the training data or fail to capture important patterns. By adjusting the model's architecture or training process based on validation results, we can improve its generalization performance and ensure it performs well on unseen data. In summary, validation plays a crucial role in assessing a model's generalization ability, enabling us to fine-tune hyperparameters and monitor performance during training, ultimately ensuring the model can effectively generalize beyond the training dataset.
2. **Model Testing:** In the final stage of the pneumonia detection project, a web app is built using Flask to provide a user-friendly interface for pneumonia prediction. The web app allows users to upload X-ray images of the chest, which are then processed by the trained model for pneumonia detection. Once an image is uploaded, the web app applies image preprocessing techniques to the input image, such as resizing and normalization, to ensure compatibility with the trained model. The preprocessed image is then passed through the loaded model to make a prediction about the presence of pneumonia. The prediction result, indicating whether pneumonia is detected or not, is displayed on the web app interface. The web app repeats this process for each uploaded image, allowing users to analyze multiple X-ray images for pneumonia detection. Overall, this web app provides a convenient and accessible way for users to utilize the trained model for pneumonia detection. It empowers healthcare professionals or individuals to quickly assess X-ray images for potential cases of pneumonia, aiding in timely diagnosis and treatment decisions.

**OBSERVATION**

The trained model has achieved an accuracy of 99% on the validation set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Train Loss | Train Accuracy  (%) | Validation Loss | Validation Accuracy (%) |
| 1 | 0.3058 | 88.70 | 0.3614 | 81.25 |
| 2 | 0.1854 | 94.39 | 0.2629 | 93.75 |
| 3 | 0.1455 | 95.52 | 0.1324 | 97.03 |
| 4 | 0.1075 | 96.88 | 0.1071 | 99.09 |

The developed web application for pneumonia detection using X-ray images demonstrates accurate prediction capabilities. The application successfully classifies X-ray images as either pneumonia or normal cases with high accuracy. This highlights the effectiveness of the implemented deep learning model and its ability to accurately identify abnormal lung patterns associated with pneumonia.

The web application's accurate predictions provide a valuable tool for assisting healthcare professionals in pneumonia diagnosis and triage. By leveraging the power of deep learning, the application can aid in the early detection of pneumonia cases and prompt initiation of treatment. This can lead to improved patient outcomes and potentially save lives.

The successful implementation of the web application emphasizes the potential of deep learning techniques in medical imaging analysis. It showcases the importance of utilizing large datasets and training deep neural networks to effectively learn and recognize complex patterns indicative of pneumonia in X-ray images.

Moreover, the web application's performance underscores the importance of data preprocessing techniques, model training, and evaluation. Proper preprocessing techniques, such as image resizing and normalization, contribute to improved model performance and generalization. Rigorous training and evaluation processes ensure the reliability and accuracy of the predictions made by the application.

**CONCLUSION**

In this paper, a deep learning-based pneumonia detection methodology using convolutional neural networks (CNNs) is proposed. The CNN classification approach is trained and tested on a large dataset of chest X-ray images sourced from open-access repositories. The proposed methodology achieves high accuracy in pneumonia detection, demonstrating the potential of deep learning in assisting radiologists and improving diagnostic outcomes.

The results of this study show that the trained CNN model is capable of accurately classifying chest X-ray images as either pneumonia or non-pneumonia cases. The high accuracy achieved indicates the effectiveness of the CNN-based approach in distinguishing between abnormal lung patterns associated with pneumonia and normal lung patterns. This can significantly aid in early detection and prompt treatment of pneumonia cases.

The success of the proposed methodology highlights the importance of data preprocessing techniques and transfer learning in deep learning applications. Proper preprocessing techniques, such as image resizing, normalization, and augmentation, contribute to improved model performance and generalization. The utilization of transfer learning, leveraging pre-trained models on large datasets, allows for effective feature extraction and classification, even with limited pneumonia-specific training data.

Also, the developed web application for pneumonia detection using X-ray images represents a significant advancement in the field of medical imaging analysis. It demonstrates the potential of deep learning models in assisting radiologists and healthcare professionals in diagnosing and managing pneumonia cases. Future research and development can further enhance the application's performance, expand its capabilities, and facilitate its integration into clinical practice to improve pneumonia diagnosis and patient care.

Overall, this study demonstrates the potential of deep learning and CNN-based approaches for pneumonia detection in chest X-ray images. It emphasizes the importance of developing robust and accurate models that can aid in the early identification and management of pneumonia cases. Future research should focus on expanding the dataset, addressing class imbalances, and conducting rigorous validation studies in real-world clinical settings to further validate the effectiveness and reliability of the proposed methodology.

**FUTURE SCOPE**

The future scope of pneumonia detection using X-ray images encompasses several exciting possibilities. Here are some potential areas for future exploration and improvement:

1. Enhanced Deep Learning Models: Researchers can continue to refine and develop deep learning models specifically tailored for pneumonia detection. This includes exploring novel architectures, incorporating attention mechanisms, and leveraging advanced techniques such as transfer learning and self-supervised learning to improve model performance.
2. Large-Scale Datasets: Building larger and more diverse datasets can help improve the robustness and generalization of pneumonia detection models. Collecting data from different populations, age groups, and geographical regions can enhance the models' ability to detect pneumonia across various demographics and reduce biases.
3. Multimodal Approaches: Integrating multiple imaging modalities, such as X-ray images and clinical data, can potentially improve the accuracy of pneumonia detection. Combining information from different sources, such as radiological findings, patient demographics, and clinical history, may lead to more comprehensive and accurate diagnostic models.
4. Interpretability and Explainability: Enhancing the interpretability and explainability of deep learning models is crucial in the medical field. Researchers can focus on developing methods to provide transparent and interpretable predictions, enabling clinicians to understand the underlying factors contributing to a model's decision and facilitating trust and adoption of AI-driven diagnostic systems.
5. Real-Time and Point-of-Care Applications: Developing fast and efficient algorithms that can provide real-time or near real-time predictions is valuable for point-of-care applications. This could enable rapid diagnosis and treatment decisions, particularly in resource-limited settings or emergency situations.
6. Integration with Clinical Decision Support Systems: Integrating pneumonia detection models with clinical decision support systems can enhance the overall workflow in radiology departments. AI algorithms can aid radiologists by providing automated preliminary interpretations, highlighting suspicious areas, and generating quantitative metrics to support diagnosis and treatment planning.
7. Continual Learning and Adaptation: Exploring techniques for continual learning and adaptation can enable models to learn from new data over time and improve their performance. This is particularly important given the evolving nature of pneumonia and the need for models to adapt to emerging strains or variations in disease patterns.
8. Longitudinal Analysis: Conducting longitudinal analysis of chest X-ray images over time can provide insights into disease progression, treatment response, and patient outcomes. By analysing sequential X-ray images, researchers can develop models that capture temporal changes in lung pathology and aid in monitoring and predicting disease progression.
9. Addressing Ethical and Bias Concerns: Future research should continue to address ethical considerations and potential biases associated with pneumonia detection models. Efforts should be made to ensure fairness, transparency, and accountability in the development, deployment, and interpretation of AI systems.

By focusing on these areas, researchers can contribute to advancing pneumonia detection using X-ray images, leading to more accurate, efficient, and clinically relevant diagnostic tools to assist healthcare professionals in providing better patient care.

**REFERENCES**

* Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., ... & Langlotz, C. P. (2017). CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv preprint arXiv:1711.05225.
* Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., ... & Yang, G. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 172(5), 1122-1131.
* Pham, T., Tran, T., & Phung, D. (2019). Classification of pneumonia in chest X-ray images using convolutional neural networks. In 2019 11th International Conference on Knowledge and Systems Engineering (KSE) (pp. 242-247). IEEE.
* Islam, M. Z., Rahman, M. S., Zhang, Y., & Khan, M. A. (2020). Deep learning for pneumonia detection in chest radiographs: A systematic review and meta-analysis. Computer Methods and Programs in Biomedicine, 197, 105617.
* Alsharidah, M., Alsulaiman, F., Alsolai, N., Alghofaili, F., Aloufi, R., & Alzahrani, A. (2021). A comprehensive review of deep learning approaches for pneumonia detection in chest X-ray images. Applied Sciences, 11(5), 2310.
* Srinivasan, R., Rayudu, R. K., Raja, N. S., & Gunasekaran, S. (2021). A review on pneumonia detection using chest X-ray images and deep learning techniques. Artificial Intelligence in Medicine, 114, 102065.