Chapter 3
Methodology

With around 4 million fatalities every year, pneumonia is one of the main causes of death in both infants and the elderly. Depending on the infectious pathogen that harms the tiny air sacs (alveoli) in the lung, it could be a virus, a bacterial infection, or a fungal infection. Patients at risk include those with underlying illnesses including asthma, compromised immune systems, hospitalised infants, and elderly people using ventilators, especially if pneumonia is not caught early. Low accuracy and efficiency of its diagnosis despite current methods necessitate more study for more precise techniques.

This section briefly covers the dataset description, preprocessing, deep learning models, and evaluation metrics. Figure shows the workflow diagram of the methodology followed in this study. Starting with the chest X-ray data, the study follows preprocessing and data augmentation. Preprocessed data are later used to train the proposed model for training and testing. The performance of the trained model is then analyzed using the unseen data, and accuracy and other well-known performance metrics are used.

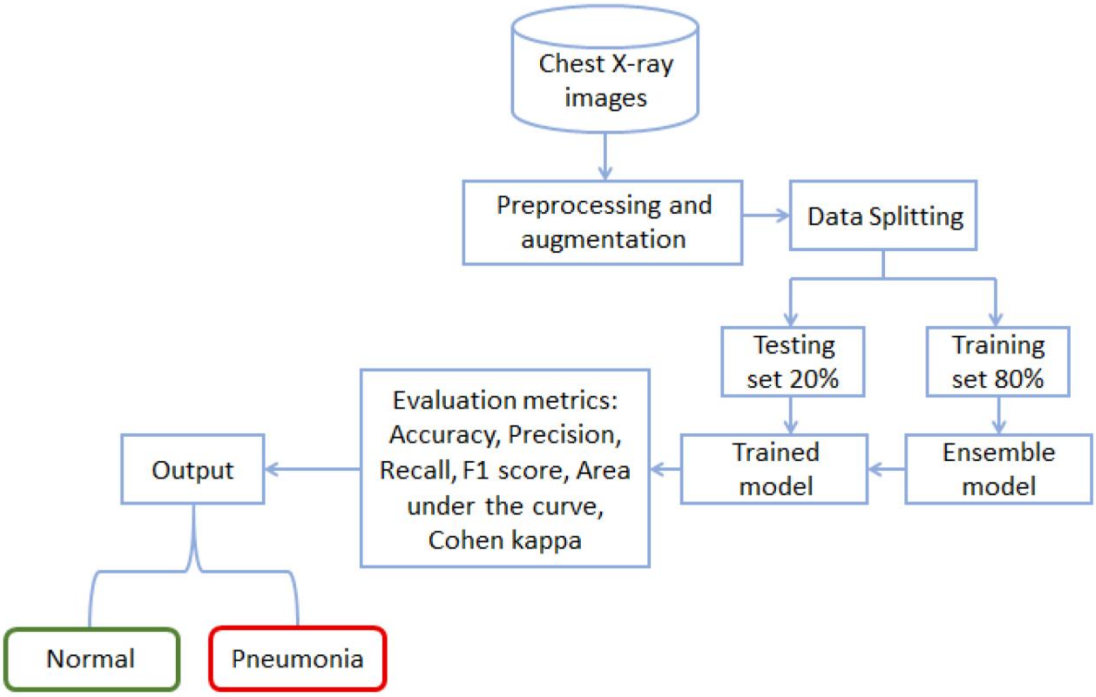


Figure3 .1: The workflow of the proposed methodology.

3.1 Dataset Collection and Pre-processing

To collect and preprocess a dataset for pneumonia detection, we should typically follow these steps:

3.1.1 Data Collection:

Find trustworthy sources, such as hospitals, medical research facilities, or open datasets, that provide medical imaging data that includes pneumonia instances. For a balanced dataset, make sure that the data you gathered includes both pneumonia-positive (cases with pneumonia) and pneumonia-negative (healthy patients) samples. As part of the data collection process, get consent and follow any moral or legal standards.

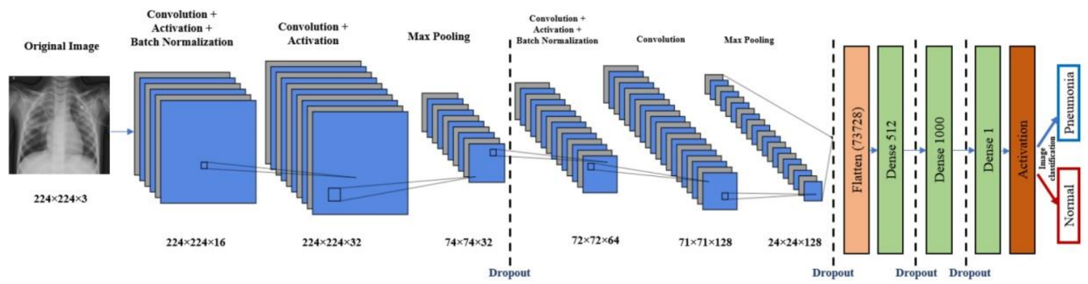


Figure3 .2: Proposed CNN architecture

Identify Reliable Sources:

- Hospitals: Collaborate with hospitals or medical institutions to access their medical imaging archives. Ensure compliance with privacy regulations and obtain necessary permissions.
- Research Databases: Explore publicly available medical image databases such as the NIH Chest X-ray dataset or the RSNA Pneumonia Detection Challenge dataset.
- Partnerships: Collaborate with research organizations or institutions specializing in medical imaging to acquire access to their datasets.

Obtain Pneumonia Cases:

- Identify cases with confirmed pneumonia diagnoses. This may involve consulting radiologists or healthcare professionals experienced in pneumonia diagnosis.
- Obtain relevant metadata associated with each case, such as patient demographics, clinical information, and any additional annotations or labels available.

Include Negative Cases:

- Ensure a balanced dataset by including pneumonia-negative cases, representing individuals without pneumonia. These can be randomly sampled from healthy individuals or from cases where pneumonia is absent.
- This balanced dataset helps the AI model learn to differentiate between pneumonia and non-pneumonia cases effectively.

Data Privacy and Anonymization:

- Take necessary precautions to protect patient privacy and comply with legal and ethical requirements.
- Remove or de-identify any personally identifiable information (PII) from the dataset, adhering to privacy regulations like the Health Insurance Portability and Accountability Act (HIPAA).

Data Diversity:

- Collect a diverse range of imaging modalities, such as chest X-rays, CT scans, or other relevant medical images, to improve the robustness of the AI model.
- Include images from different demographics, age groups, and geographical locations to ensure the dataset's representativeness.

Quality Control:

- Ensure the quality of the collected dataset by involving experienced radiologists or healthcare professionals to review and validate the diagnoses.
- Perform checks for data inconsistencies, artifacts, and image quality issues. Remove any unreliable or erroneous data from the dataset.

It's crucial to remember that gathering a thorough dataset for pneumonia diagnosis using AI may need a lot of money, knowledge, and adherence to rules. The availability of high-quality and clinically validated data can be helped by cooperation with medical experts and institutes.

3.1.2 Data Preprocessing:

Transform the medical imaging data into an appropriate format, such as the widely used DICOM (Digital Imaging and Communications in Medicine) format. To improve comparability, normalise the pixel intensity levels across the photos. The photos should be resized or scaled to a uniform resolution while taking the computational limitations and specifications of your selected model into account. Create training, validation, and test sets from the dataset. According to the size of the dataset and the particular needs of your project, a typical split can be 70% for training, 15% for validation, and 15% for testing. In order to prevent biases during training, shuffle the data.

An essential step in getting a dataset ready for AI-based pneumonia detection is data preprocessing. Consider the following typical preprocessing steps:

Data Format Conversion:

- Convert the medical imaging data into a format suitable for AI algorithms, such as DICOM (Digital Imaging and Communications in Medicine) for medical images like X-rays or CT scans.
- Ensure that the metadata associated with each image, such as patient information and imaging parameters, is properly preserved.

Image Rescaling and Normalization:

- Resize or rescale the images to a consistent resolution. This step is important to ensure uniformity in the dataset and to match the input requirements of your AI model.
- Normalize the pixel intensity values across the images to enhance comparability and reduce the influence of variations in imaging equipment or

protocols. Common normalization techniques include z-score normalization or min-max scaling.

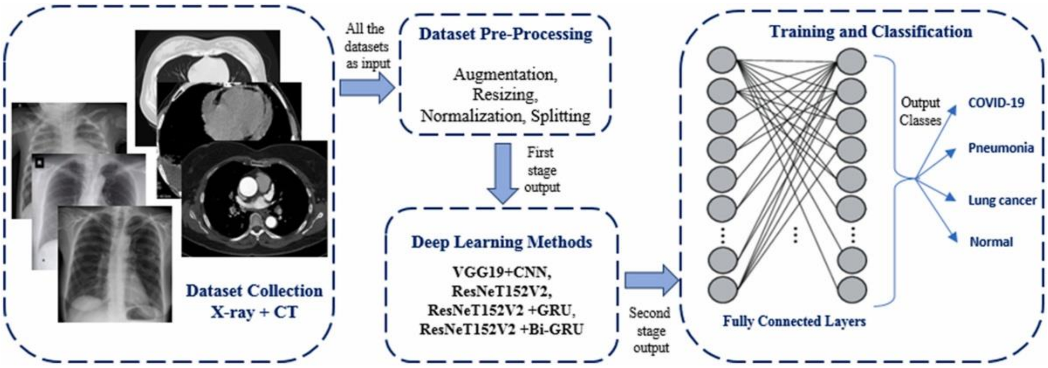


Figure3 .3: Model Block Diagram for Pre-Processing

Data Augmentation:

- Data augmentation techniques can help increase the size and diversity of the dataset, leading to better generalization of the AI model.
- Apply augmentation operations such as rotation, flipping, zooming, shearing, or adding random noise to the images. However, in medical imaging, it's crucial to ensure that the augmentations preserve the diagnostic features and maintain the integrity of the original images.

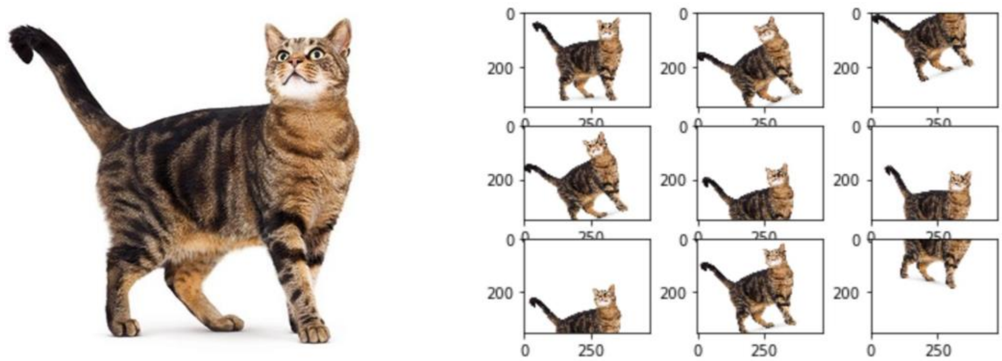


Figure3 .4: Data Augmentation in machine learning [8]

Splitting the Dataset:

- Divide the dataset into three subsets: training, validation, and test sets. The training set is used to train the AI model, the validation set is used for hyperparameter tuning, and the test set is used to evaluate the final model's performance.

- The recommended split is typically around 70-80% for training, 10-15% for validation, and 10-15% for testing. Adjustments can be made based on the size of the dataset and specific requirements.

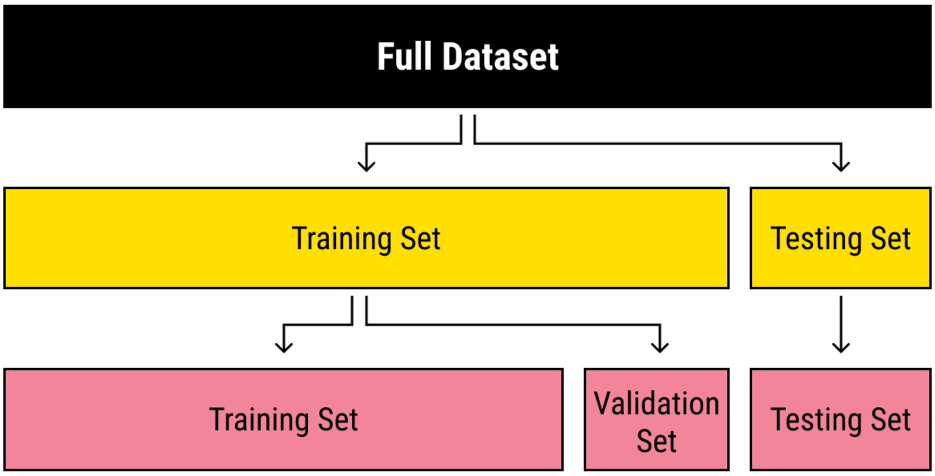


Figure3 .5: Splitting of a dataset into training, testing, and validation datasets [8]

Data Balancing:

- Check if the dataset is imbalanced, i.e., if the number of pneumonia-positive and pneumonia-negative cases is significantly different.
- If the dataset is imbalanced, employ techniques such as oversampling the minority class (pneumonia-positive) or undersampling the majority class (pneumonia-negative) to create a more balanced distribution. Alternatively, you can use techniques like Synthetic Minority Over-sampling Technique (SMOTE) to create synthetic samples of the minority class.

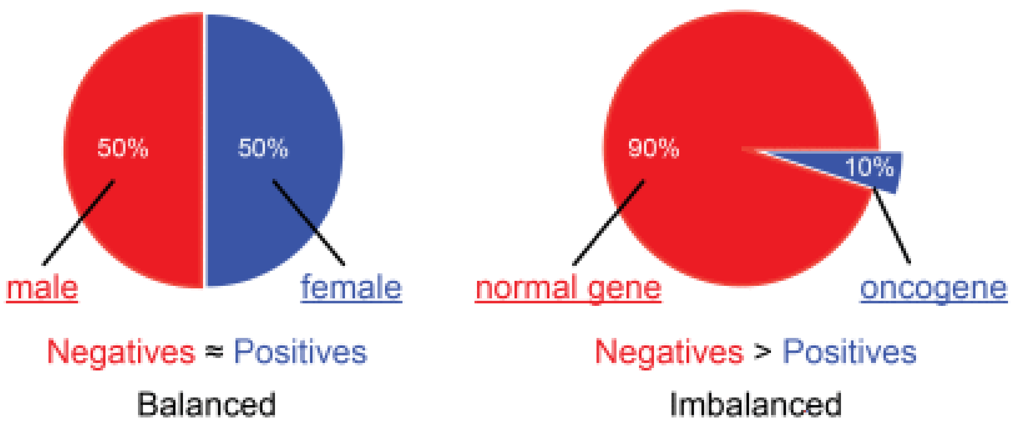


Figure3 .6: Example of balanced and imbalanced data [8]

Preprocessing Labels:

- Assign appropriate labels to each image indicating whether it is pneumonia-positive or pneumonia-negative.
- Ensure that the labeling process is performed accurately, preferably by experienced radiologists or healthcare professionals who can provide reliable ground truth annotations.

Data Shuffle:

- Shuffle the data to eliminate any potential biases during training or evaluation. Randomly shuffling the dataset ensures that the AI model is not influenced by the order of the samples.

It's crucial to keep in mind that the precise preprocessing procedures may change depending on the imaging modalities, the type of data, and the needs of your AI model. To make sure that the preprocessing stages adhere to industry best practises, it is advised to speak with domain specialists, such as radiologists or medical imaging researchers.

3.1.3 Data Augmentation:

The amount of the dataset can assist your model become more reliable and universal. Use tricks like rotation, flipping, magnification, or adding random noise to provide extra training samples. In medical imaging, it is crucial to ensure that the enhanced data maintains the characteristics and integrity of the original medical images in order to prevent delivering false or misleading information.

3.1.4 Labeling:

Give each image the proper labels indicating whether it is positive or negative for pneumonia. To guarantee accuracy, make sure that the labelling procedure is carried out by medical professionals or specialists with knowledge of diagnosing pneumonia.

3.1.5 Balancing the Dataset:

Verify the dataset to see if the proportion of pneumonia-positive and pneumonia-negative instances differs noticeably. If the dataset is unbalanced, you can correct the problem by creating a balanced distribution by either oversampling the minority class (pneumonia-positive) or undersampling the majority class (pneumonia-negative). As an alternative, we can employ strategies like data augmentation targeted directly at the minority class to raise its representation.

3.1.6 Data Split:

Divide the dataset into features (the input photos) and labels (pneumonia-positive or pneumonia-negative) after preprocessing and labelling. Ensure that samples with and without pneumonia are distributed equally among the training, validation, and test sets. These procedures offer a broad framework for gathering and preparing a dataset for the detection of pneumonia. The precise implementation, however, may change based on the data at hand, the imaging modalities (such as X-rays or CT scans), and the demands of your project.

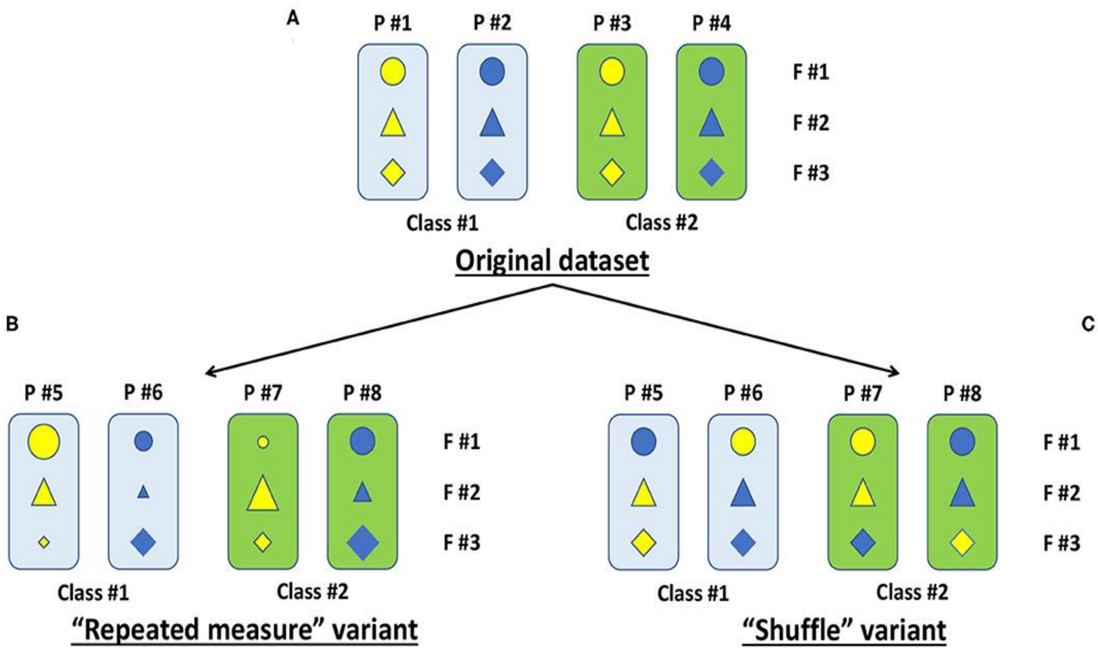


Figure3 .7: Example of data shuffle [8]

3.2 Deep Convolutional Neural Networks

AI pneumonia diagnosis using Deep Convolutional Neural Networks (CNNs) has demonstrated impressive success. As a result of their capacity to automatically learn hierarchical characteristics from raw pixel data, CNNs are especially well-suited for the analysis of medical images. Here is a summary of how CNNs can be applied to the identification of pneumonia:

3.2.1 Architecture Selection:

- We can choose an appropriate CNN architecture that has demonstrated strong performance in image classification tasks. Examples include VGGNet, ResNet, InceptionNet, or DenseNet.
- Pretrained models, such as those trained on large-scale image datasets like ImageNet, can serve as a starting point and provide a good initialization for pneumonia detection.

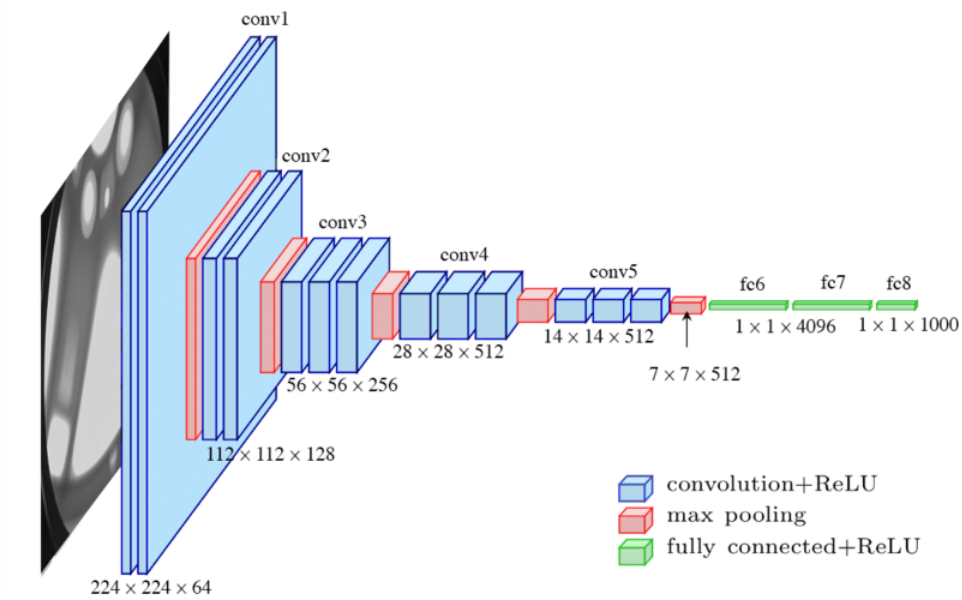


Figure3 .8: VGG-Net Architecture Explained [8]

When selecting a Convolutional Neural Network (CNN) architecture for pneumonia detection, several well-established models have been successful in image classification tasks and can serve as a starting point. Here are some popular CNN architectures:

VGGNet (Visual Geometry Group Network):

- VGGNet is known for its simplicity and is composed of a series of stacked convolutional layers with small 3x3 filters.

- It comes in different variations, such as VGG16 and VGG19, which differ in depth and number of layers.
- VGGNet has been widely used in medical image analysis tasks, including pneumonia detection.

ResNet (Residual Neural Network):

- ResNet introduced the concept of residual connections, enabling the training of very deep neural networks.
- It utilizes residual blocks where the output of a layer is added to the input, allowing the network to learn residual mappings.
- ResNet models, such as ResNet50 or ResNet101, have shown excellent performance in various image classification tasks. The arrow shows the identity mapping

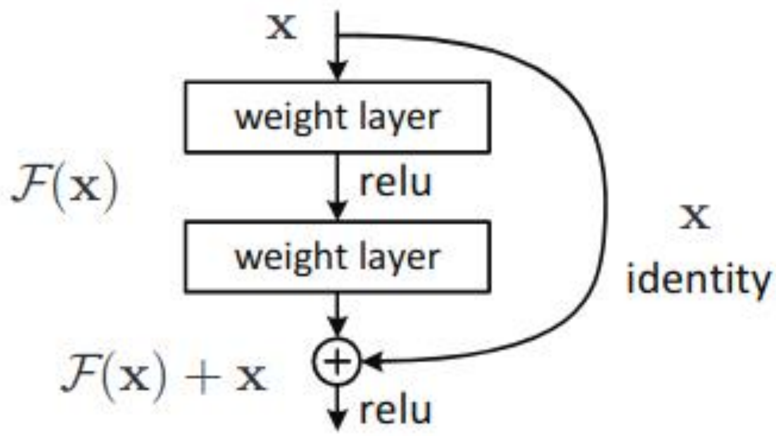


Figure3 .9: Residual learning building block that is used in ResNets [8]

InceptionNet (GoogLeNet):

- InceptionNet introduced the concept of Inception modules, which employ multiple parallel convolutional operations at different scales.
- It incorporates 1x1, 3x3, and 5x5 convolutions within the same module to capture different levels of abstraction.
- InceptionNet models, like InceptionV3 or InceptionResNetV2, have achieved strong results in image classification and detection tasks.

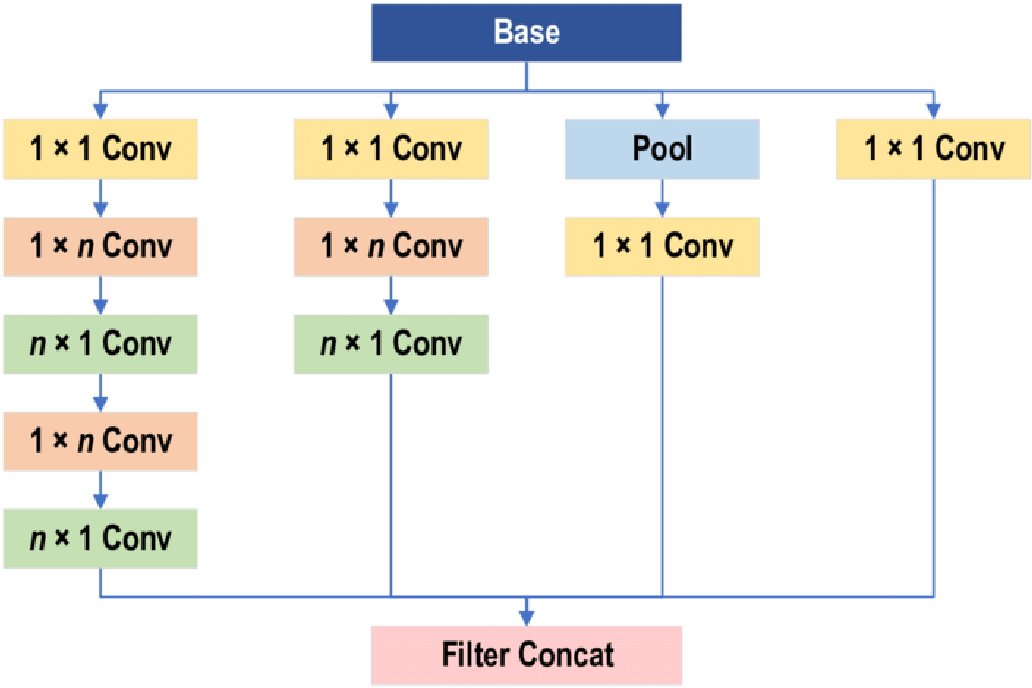


Figure3 .10: Inception module of InceptionNet [4]

DenseNet (Densely Connected Convolutional Network):

- DenseNet introduces dense connections, where each layer receives direct input from all preceding layers in a feed-forward fashion.
- It promotes feature reuse, enhances gradient flow, and reduces the number of parameters.
- DenseNet models, such as DenseNet121 or DenseNet169, have demonstrated strong performance in medical image analysis tasks.

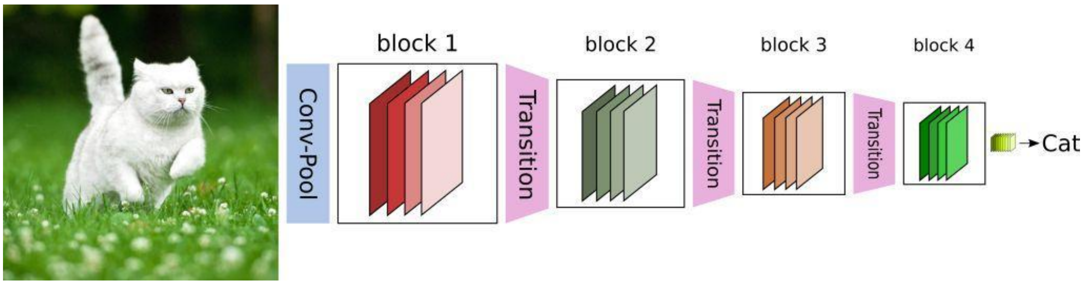


Figure3 .11: Densely Connected Convolutional Networks in Tensorflow [6]

EfficientNet:

- EfficientNet introduces a scalable architecture that balances model depth, width, and resolution using a compound scaling method.

- It achieves state-of-the-art performance with efficient resource utilization.
- EfficientNet models, such as EfficientNetB0 to EfficientNetB7, are designed to be scalable for different computational constraints.

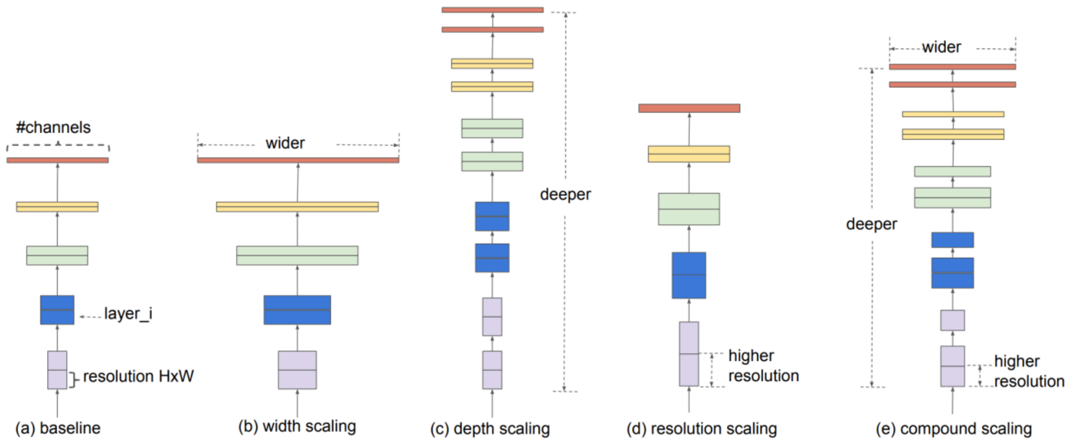


Figure3 .12: Model Scaling (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ration. [6]

These are merely a few instances of CNN architectures that are frequently applied to image classification problems, such as pneumonia diagnosis. Regarding model complexity, performance, and resource needs, each design has advantages and disadvantages of its own. It is advised to investigate and contrast their performance on your particular dataset and task in order to choose the best design. Additionally, keep in mind that when choosing an architecture, you should take into account your dataset's size and kind, resource availability, and computational constraints.

3.2.2 Transfer Learning:

- Utilize transfer learning to leverage the knowledge learned from a pretrained CNN model on a large dataset.
- Retrain only the last few layers (or specific blocks) of the CNN while keeping the earlier layers frozen to avoid overfitting and expedite training.
- Fine-tuning the CNN with pneumonia-specific data helps adapt the model to the pneumonia detection task.

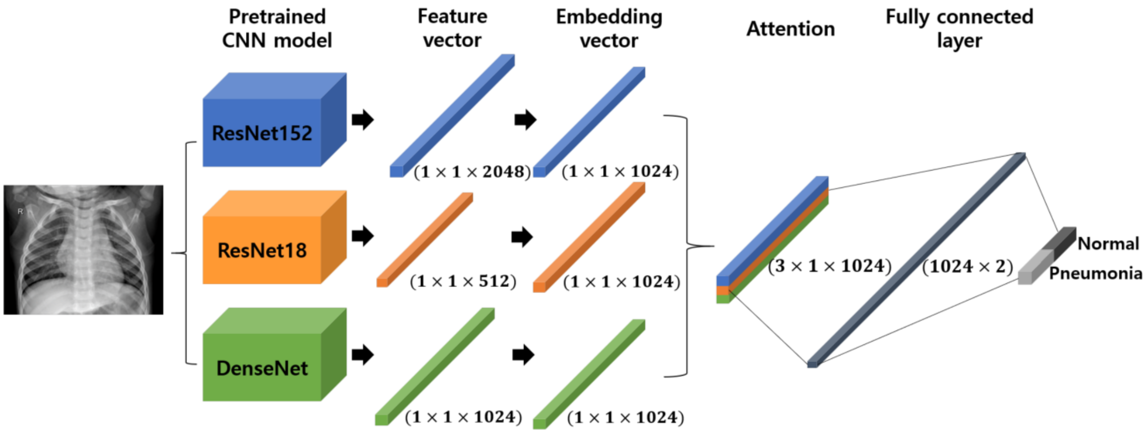


Figure3 .13: The framework based on transfer learning with attention mechanism with pneumonia [39]

3.2.3 Input Preprocessing:

- Normalize the input images to ensure consistent pixel intensity values.
- Resize or rescale the images to match the input size required by the chosen CNN architecture.
- Consider converting the images to a three-channel format (e.g., RGB) if the pretrained CNN expects input in that format.
- We utilized pre-trained models from 3 different domains, ImageNet [15], ChestX-ray14 datasets [28] and Custom dataset [29] to gain the generalization ability of our framework. The three pre-trained models we used are quite large and have many parameters compared to the test data.
- So, we apply some data preprocessing to avoid over-fitting. We used image augmentation methods for chest X-ray images. Data preprocessing is a series of processes that makes input data suitable for specific analysis. For example, Crop means cropping the pixels at the edges when the center of the image is important. Rotating rotates images at a random angle.
- Flip means flipping images left and right or up and down to make more data. It also includes normalization, which converts the data range to a value between 0 and 1 in order to reduce the impact on the relative size of data. We used crop, rotate, and flip and normalization in data preprocessing.
- First, we modified the image size to (224×224), rotated randomly selected images, flipped those images horizontally and applied normalization to make

each dimension in data have values within the same range. This improves the ability to generalize the model.

- In case low-quality images are given as input, our approach can reach the compliance performance through data pre-processing. However, the performance can improve when higher quality images are given as input.
- CNN is a model that improves the problem of Deep Neural Network(DNN). DNN model uses only one-dimensional data. However, image data is two-dimensional. Using DNN for two-dimensional data creates a large loss problem in the process of changing to one-dimensional data.
- CNN is a proposed model that can be trained on image data by applying filters of a certain size to the image. We utilized pre-trained models that were trained on three different domains, ImageNet, Chest X-ray14 datasets and Custom datasets. The three models are ResNet152, Dense121, ResNet18.

ResNet : In order to combat the performance loss brought on by vanishing gradient, ResNet [10] was devised. Performance deteriorates and then abruptly decreases as model depth rises. Using a skip-connection that adds inputs at the very end to ensure that the gradient is at least 1, ResNet learns from the residual. By including a shortcut link, the issue is changed to reflect how far the residual deviates from the starting value. Only the residual (the difference between the output of the previously learned layers and the output of the additional layer) needs to be learned because the output values of the current layer and the previous layer are combined to receive as input. This network resolves the gradient disappearing issue using these techniques. To meet our objective, we merely modify the last fully linked layer (classifier) and the number of layers in these ResNet topologies. ResNet's general structure is depicted in Figure 3.13. When the channel in the main path through the input differs from the channel in the shortcut path through which the residuals flow, the conv block is a block that aligns the channels in the shortcut path. When the shortcut path channel and the main path channel coincide, the identity block does nothing more complicated than basic addition. Data that is one dimensional is flattened into a fully connected layer (FC). ResNet18 and ResNet152 were the two ResNet variants we used. 18 layers and 152 layers, respectively, make up each model. The problem of performance deterioration brought on the vanishing gradient is resolved by ResNet. The performance of

ResNetimproves with layer depth. There are structural variations between ResNet18's use of 3 by 3 filters and ResNet152's use of 1 by 1 filter.

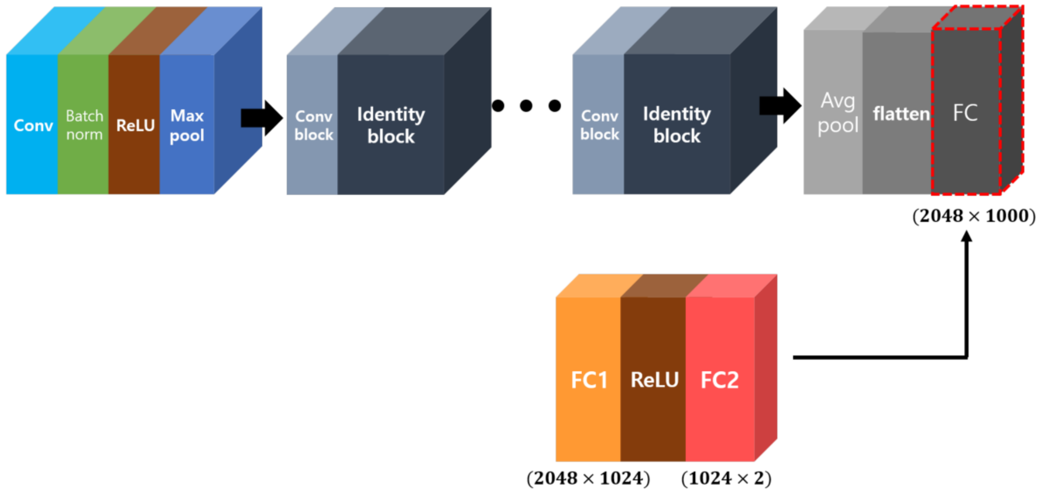


Figure3 .14: PretrainedResNet with changed classifier on ImageNet[39]

DenseNet : DenseNet [11] is a method of stacking all the components that have been passed through during learning, whereas ResNet [10] is a method of moving forward by adding the prior information. Each layer with the same feature map size is directly connected to the other levels of the network to maximise information flow between them. All layers of the DenseNet structure (illustrated in Figure 3) are connected in a feed-forward fashion. Each DenseBlock's convolution layer concatenates the outputs of all earlier convolution levels. The convolutional operation receives this feature map that has been concatenated. Continuously concatenating the feature maps of the preceding and subsequent layers rather than simply adding them together is how the connection is made. To maintain the feed-forward feature, each layer receives a new input from all preceding layers and sends the most recent feature map to the following layer. Learning is facilitated by DenseNet since each layer directly approaches the gradient derived from the loss function and input and compensates for the loss of initial information as the layer depth increases. Additionally, compared to other models, filters are distributed more thickly on each layer, which effectively takes information and lowers the number of parameters. However, the amount of processing power needed increases with layer depth. We use the pre-trained DenseNet weights, but we change the size of DenseNet121's last fully connected layer to accommodate the amount of our classification classes.

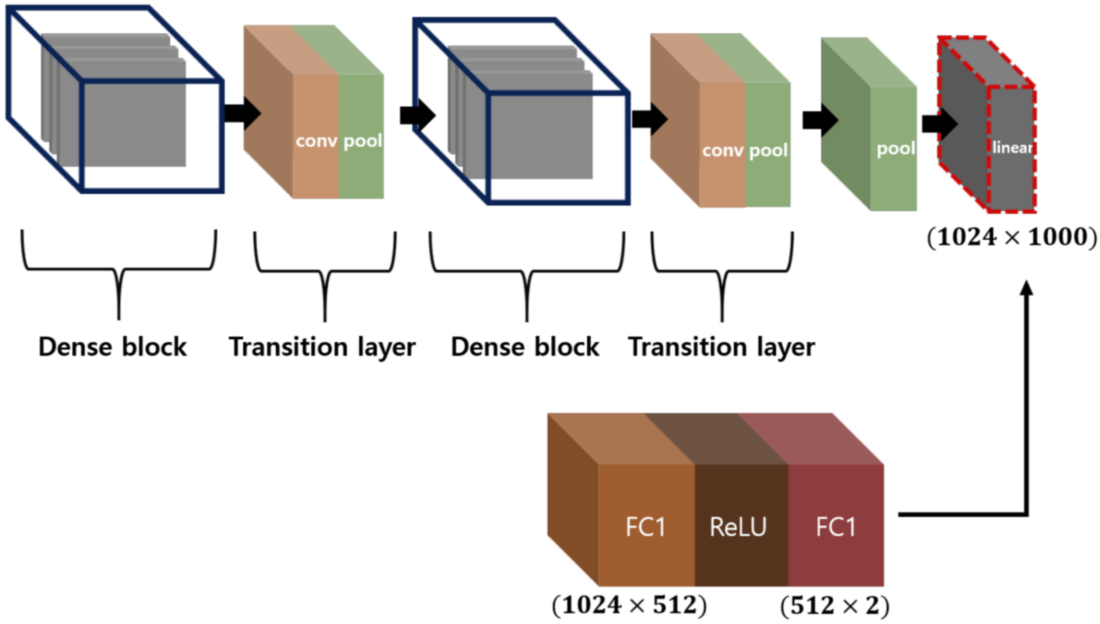


Figure3 .15: PretrainedDenseNet with changed classifier on Chest X-ray14 [39]

3.2.4 Model Training:

- Split the dataset into training, validation, and test sets.
- Feed the preprocessed images into the CNN and train the network using backpropagation and gradient descent optimization algorithms.
- Monitor the training process using appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score) on the validation set to guide hyperparameter tuning and prevent overfitting.
- Regularization techniques such as dropout or L2 regularization can be applied to mitigate overfitting.

3.2.5 Evaluation and Testing:

- Assess the performance of the trained CNN model on the test set using standard evaluation metrics.
- Analyze additional metrics like sensitivity (recall), specificity, and area under the receiver operating characteristic curve (AUC-ROC) to measure the model's discriminatory power.
- Visualize and interpret the model's predictions and learned features to gain insights into its decision-making process.

3.2.6 Iterative Refinement:

- Iteratively refine the CNN model by adjusting hyperparameters, exploring different architectures, or incorporating additional techniques like ensembling or data augmentation to improve performance.

It's important to note that deploying an AI model for pneumonia detection in a clinical setting requires thorough validation, rigorous evaluation, and compliance with relevant regulatory and ethical considerations. Collaboration with medical experts, radiologists, or healthcare professionals is crucial to ensure the safety and reliability of the AI system.

3.3 Multimodal Fusion Techniques

Information that has been combined and integrated using several modalities or sources is referred to as using multimodal fusion techniques. Any type of data source, including sensor data, audio, video, text, pictures, and other structured or unstructured data types, can be referred to as a modality. Multimodal fusion aims to improve decision-making, analysis, and knowledge beyond what can be accomplished by individual modalities alone by utilising the complimentary qualities of various modalities. There are several types of multimodal fusion techniques commonly used:

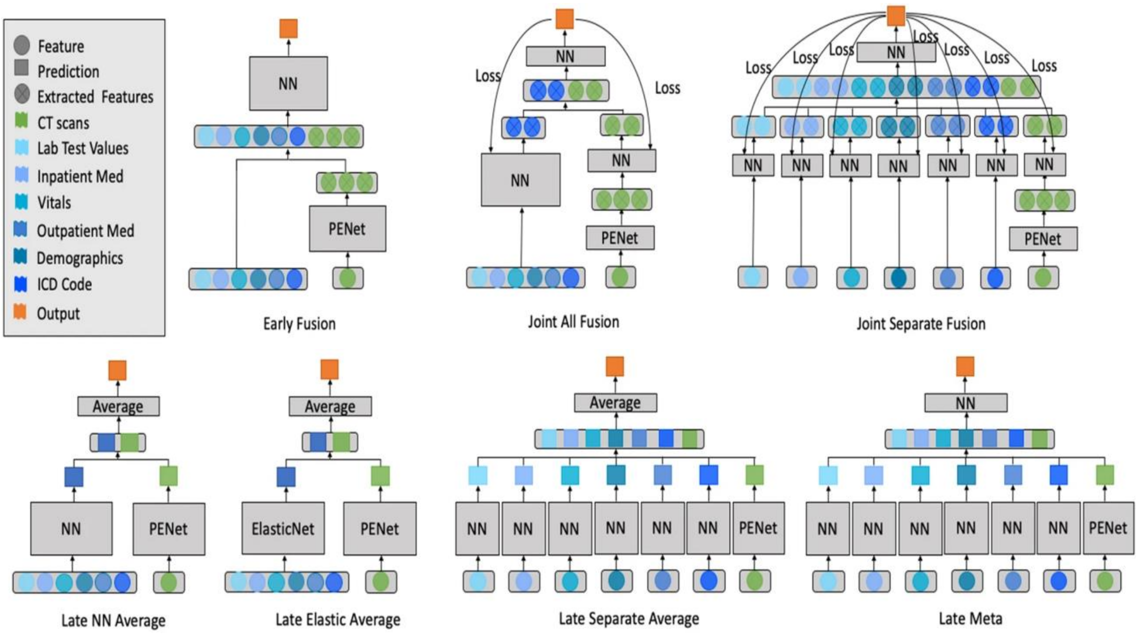


Figure3 .16: Multimodal fusion with deep neural networks [9]

3.3.1 Early Fusion:

Prior to any additional processing, early fusion combines data at the raw data level from various modalities. As an illustration, in image and text fusion, a single feature vector is created by concatenating or combining the visual features taken from images and the textual features extracted from text documents.

3.3.2 Late Fusion:

In late fusion, each modality is processed independently, and the findings are combined afterwards. Using methods like voting, averaging, or weighted combining, the outputs of the several modality-specific models are combined. This method gives you the freedom to model each modality separately, but it might not fully take advantage of how interdependent the modalities are.

3.3.3 Hybrid Fusion:

Early and late fusion are intended to be balanced by hybrid fusion techniques. They combine data at the feature level as well as the decision level. Based on the unique properties of the data and the task at hand, hybrid fusion approaches can be created.

3.3.4 Fusion at Different Levels:

Different levels of abstraction can be used while doing multimodal fusion. Raw sensor-level data or low-level characteristics are combined in low-level fusion. Higher-level characteristics or representations that were retrieved from each modality are combined in mid-level fusion. Topic models or knowledge graphs, for example, which are derived from the modalities, are integrated through high-level fusion.

Various machine learning methods and models are frequently used in multimodal fusion techniques to extract characteristics, capture intermodal linkages, and generate predictions or choices. Along with more conventional machine learning methods like support vector machines (SVMs) or decision trees, these algorithms can include deep learning structures like convolutional neural networks (CNNs), recurrent neural networks (RNNs), or transformer models.

Numerous industries, such as computer vision, natural language processing, healthcare, multimedia analysis, human-computer interaction, and more use multimodal fusion techniques. These techniques enable richer and more thorough analysis by combining data from many modalities, which improves performance and yields new insights across a range of disciplines.

3.4 Transfer Learning for Pneumonia Detection

The process of using information gained from one activity or area to do better on a related task or domain is known as transfer learning. Transfer learning can be used in the context of pneumonia detection to make use of pre-trained models and information obtained from substantial datasets in adjacent fields, such as general medical imaging or chest X-ray analysis, to improve the precision and effectiveness of pneumonia identification. Here is an explanation of transfer learning for pneumonia detection:

3.4.1 Pre-trained Models:

Deep learning models that have been pre-trained have been trained on sizable datasets for a particular goal, like object detection or picture categorization. These models have mastered the art of finding general and useful features in the data. These pre-trained models are used as a starting point or feature extractor for pneumonia detection in transfer learning.

3.4.2 Feature Extraction:

Pre-trained models' initial layers are responsible for capturing low-level features that are shared by a variety of tasks and domains. To extract meaningful representations from chest X-ray pictures for the purpose of detecting pneumonia, these layers can be employed as a feature extractor. One can take advantage of the learnt representations and reduce computation and training time by leveraging pre-trained models.

3.4.3 Fine-tuning:

Pre-trained models can be used to extract characteristics, which can then be fine-tuned to fit the model to the particular goal of pneumonia diagnosis. A smaller dataset that is specific to pneumonia is used to update the weights of some or all of the layers in the pre-trained model during fine-tuning. This enables the model to retain its pre-training-phase broad knowledge while learning task-specific information. Steps for the fine tuning may be

- Pretrain a neural network model, i.e., the source model, on a source dataset (e.g., the ImageNet dataset).
- Create a new neural network model, i.e., the target model. This copies all model designs and their parameters on the source model except the output layer. We assume that these model parameters contain the knowledge learned from the source dataset and this knowledge will also be applicable to the target dataset. We also assume that the output layer of the source model is closely related to the labels of the source dataset; thus it is not used in the target model.
- Add an output layer to the target model, whose number of outputs is the number of categories in the target dataset. Then randomly initialize the model parameters of this layer.
- Train the target model on the target dataset, such as a chair dataset. The output layer will be trained from scratch, while the parameters of all the other layers are fine-tuned based on the parameters of the source model.

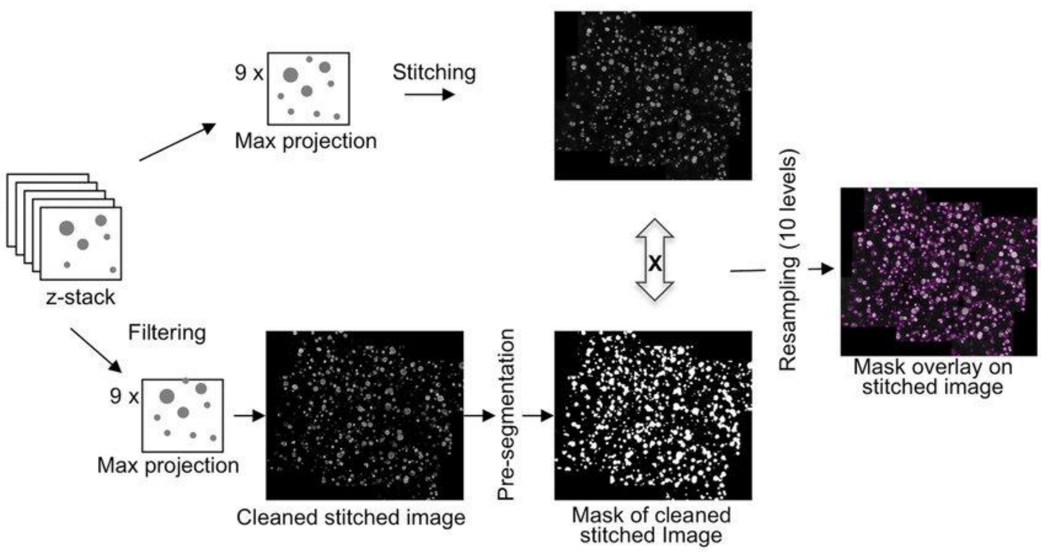


Figure3 .17: Image processing and feature extraction [9]

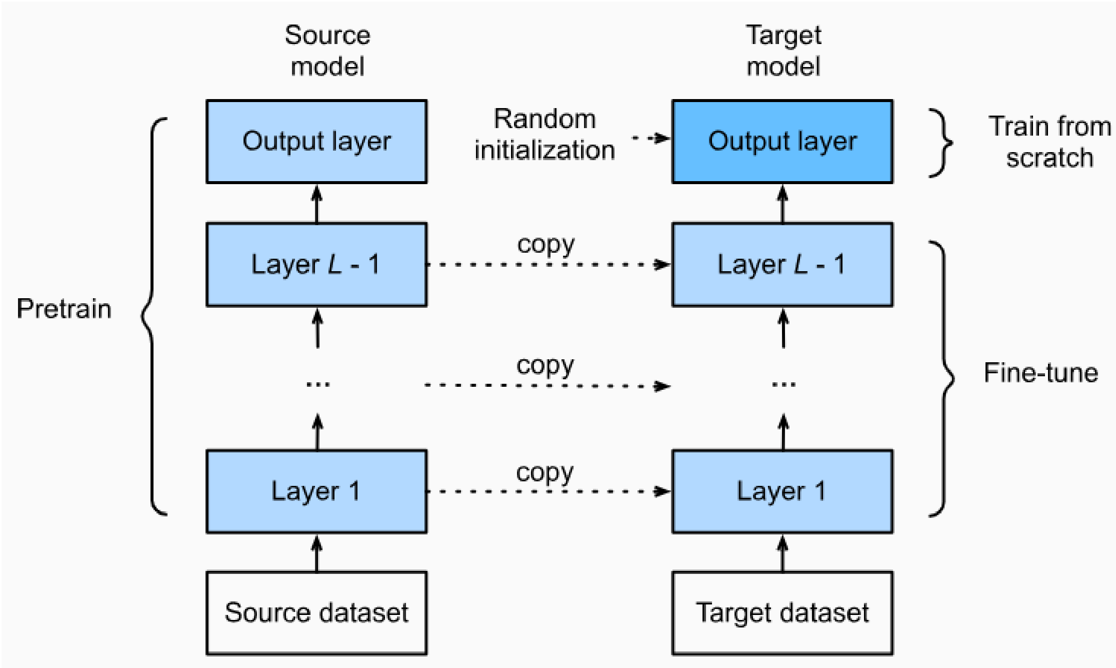


Figure3 .18: Image Fine tuning [9]

3.4.4 Transfer Learning Benefits:

Transfer learning for the early identification of pneumonia has various benefits. First off, it lessens the demand for big annotated datasets for starting fresh with deep learning models. The use of prior information and representations acquired from similar tasks is also made possible, which can enhance generalisation and accuracy. Last but not least, transfer learning can shorten training time by starting the model with weights that have already been taught.

3.4.5 Domain Adaptation:

Dealing with domain shifts, such as variations in imaging methods or data distribution across various hospitals or sources, may be necessary for pneumonia detection. In order to close the domain gap and enhance model performance on the intended task, domain adaptation techniques can be used in transfer learning to align the source domain (pre-training data) with the target domain (pneumonia detection data).

Researchers and practitioners in the field of pneumonia detection can take advantage of the abundance of knowledge and expertise stored in pre-trained models by using transfer learning techniques, creating more precise and effective detection systems.

3.5 Model Training and Evaluation

Model training and evaluation for pneumonia detection involves the process of training machine learning or deep learning model using labeled data and assessing its performance on unseen data.

3.5.1 Model Selection:

- DealingChoose an appropriate machine learning or deep learning model architecture suitable for pneumonia detection.
- Consider models such as convolutional neural networks (CNNs) or hybrid models for multimodal fusion, depending on the specific requirements and available data

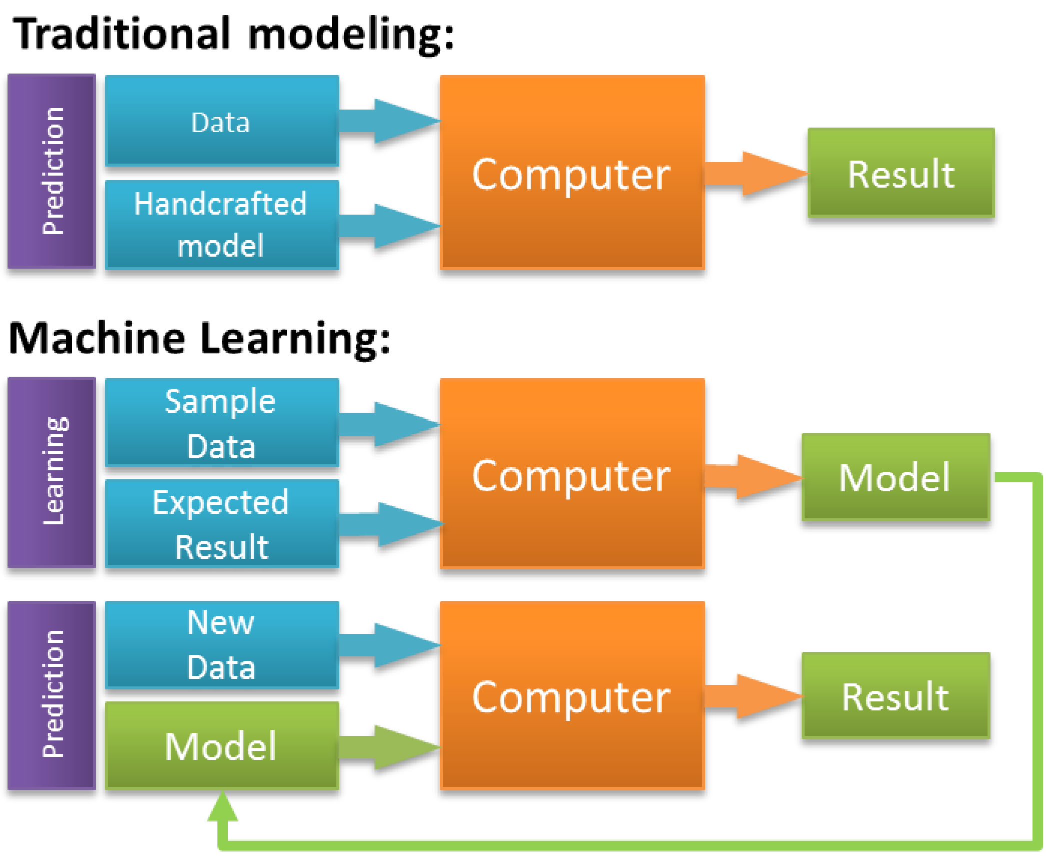


Figure3 .19: Traditional v/s Machine Learning Model [9]

3.5.2 Model Training:

- Initialize the model with random weights or pre-trained weights from a related task using transfer learning.
- Train the model using the training set.

- Define an appropriate loss function, such as binary cross-entropy, and an optimization algorithm, like stochastic gradient descent (SGD) or Adam, to update the model's weights.
- Adjust hyperparameters (e.g., learning rate, batch size) to optimize training performance.
- Monitor the model's performance on the validation set during training and perform early stopping if necessary.

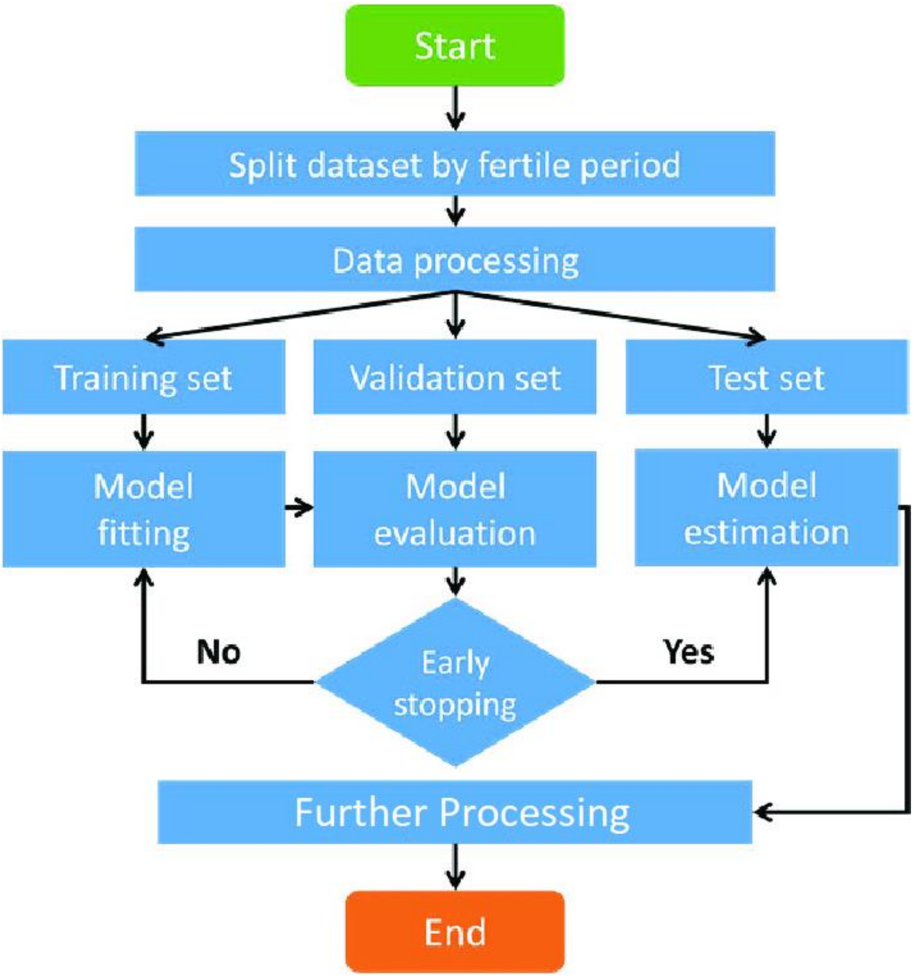


Figure3 .20: Model training process [9]

3.5.3 Model Evaluation:

- Initialize Evaluate the trained model on the testing set, which contains unseen data.
- Calculate various evaluation metrics, including accuracy, precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve.

- Generate a confusion matrix to analyze the model's performance in classifying pneumonia and non-pneumonia cases.
- Consider additional metrics, such as sensitivity and specificity, based on the specific requirements of pneumonia detection.

3.6 Performance Metrics

For simplicity, we will mostly discuss things in terms of a binary classification problem where let's say we'll have to find if an image is of a patient is having pneumonia (positive) or is found healthy (negative). Some common terms to be clear with are: True positives (TP): Predicted positive and are actually positive. False positives (FP): Predicted positive and are actually negative. True negatives (TN): Predicted negative and are actually negative. False negatives (FN): Predicted negative and are actually positive

3.6.1 Confusion matrix:

- It's just a representation of the above parameters in a matrix format. Better visualization is always good

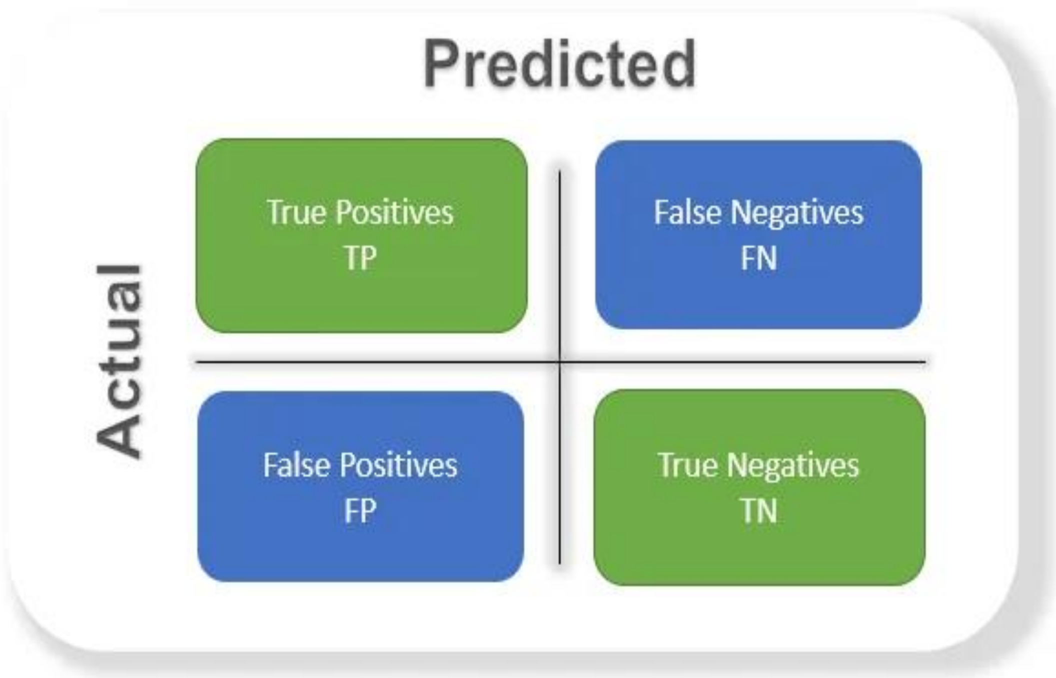


Figure3 .21: Matrix of relationships between actual and model-predicted answers [39]

3.6.2 Accuracy:

- The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalanced
- Take for example a pneumonia detection model. The chances of actually having pneumonia are very low. Let's say out of 100, 90 of the patients don't have pneumonia and the remaining 10 actually have it. We don't want to miss on a patient who is having pneumonia but goes undetected (false negative). Detecting everyone as not having cancer gives an accuracy of 90% straight. The model did nothing here but just gave pneumonia free for all the 100 predictions

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

3.6.3 Precision:

- Percentage of positive instances out of the total predicted positive instances. Here denominator is the model prediction done as positive from the whole given dataset. Take it as to find out 'how much the model is right when it says it is right'.

$$Precision = \frac{TP}{TP + FP}$$

3.6.4 Recall/Sensitivity/True Positive Rate:

- Percentage of positive instances out of the total actual positive instances. Therefore denominator (TP + FN) here is the actual number of positive instances present in the dataset. Take it as to find out 'how much extra right ones, the model missed when it showed the right ones'.

$$Recall = TP\ rate = \frac{TP}{TP + FN}$$

3.6.5 Specificity:

- Percentage of negative instances out of the total actual negative instances. Therefore denominator (TN + FP) here is the actual number of negative instances present in the dataset. It is similar to recall but the shift is on the negative instances. Like finding out how many healthy patients were not having pneumonia and were told they don’t have pneumonia. Kind of a measure to see how separate the classes are.

$$\frac{TN}{TN + FP}$$

3.6.6 F1 score:

- It is the harmonic mean of precision and recall. This takes the contribution of both, so higher the F1 score, the better. See that due to the product in the numerator if one goes low, the final F1 score goes down significantly. So a model does well in F1 score if the positive predicted are actually positives (precision) and doesn't miss out on positives and predicts them negative (recall).

$$F1\ Score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precesion * recall}{precesion + recall}$$

- One drawback is that both precision and recall are given equal importance due to which according to our application we may need one higher than the other and F1 score may not be the exact metric for it. Therefore either weighted-F1 score or seeing the PR or ROC curve can help.

3.6.7 PR curve:

- The PR Curve is another way to interpret the effect of the boundary threshold effect on our classifier. It measures the usefulness and the completeness of the classification.
- It is the curve between precision and recall for various threshold values. In the figure below we have 6 predictors showing their respective precision-recall curve for various threshold values.
- The top right part of the graph is the ideal space where we get high precision and recall. Based on our application we can choose the predictor and the threshold value. PR AUC is just the area under the curve. The higher its numerical value the better.
- As it happens for the ROC Curve, higher curves correspond to better classifiers and the tradeoff between the two parameters will have to be chosen depending on the characteristics of the classifier.

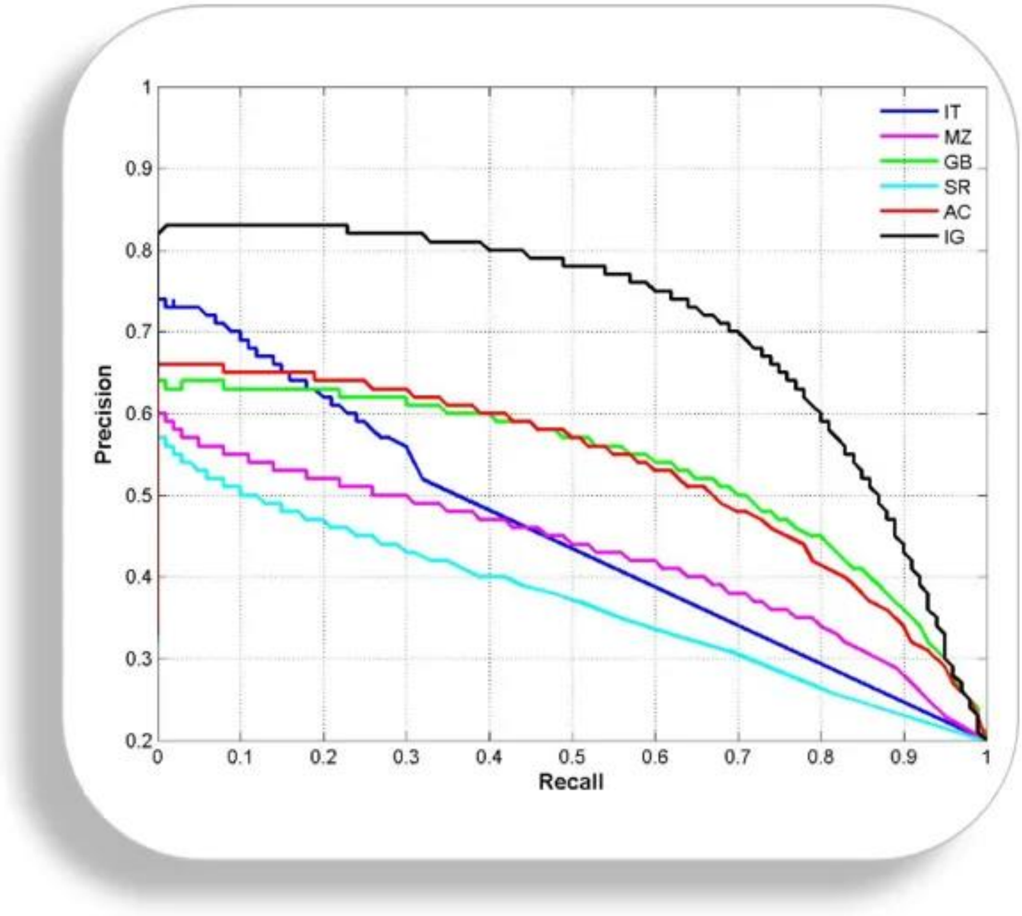


Figure3 .22: PR Curve sample [25]

3.6.8 ROC curve:

- ROC stands for receiver operating characteristic and the graph is plotted against TPR and FPR for various threshold values. As TPR increases FPR also increases.
- As you can see in the first figure, we have four categories and we want the threshold value that leads us closer to the top left corner.
- Comparing different predictors (here 3) on a given dataset also becomes easy as you can see in figure 2, one can choose the threshold according to the application at hand.
- ROC AUC is just the area under the curve, the higher its numerical value the better.

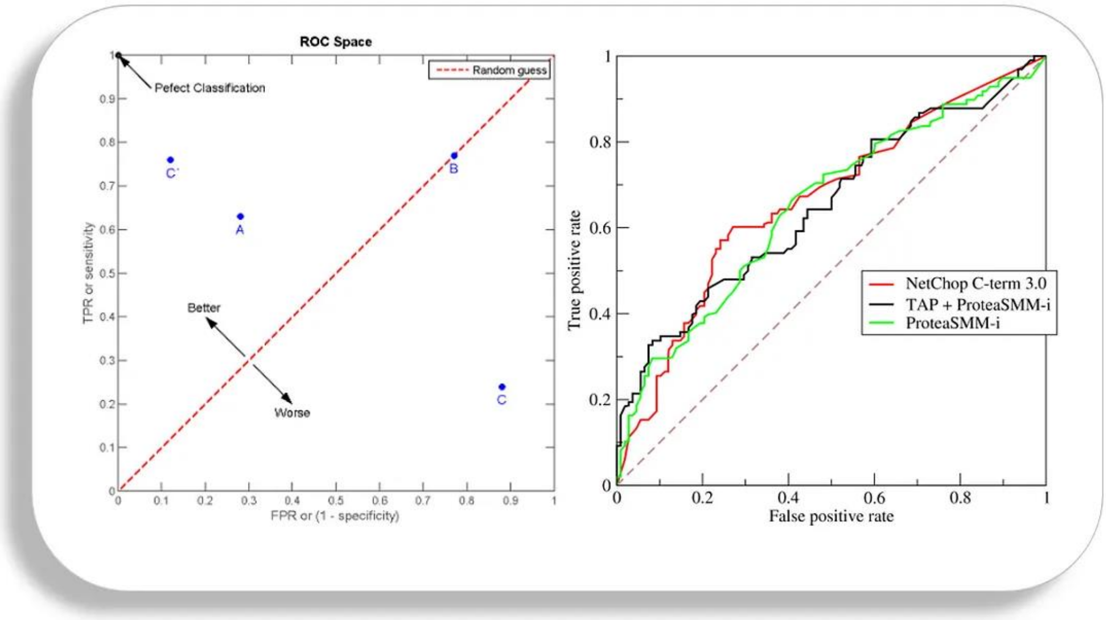
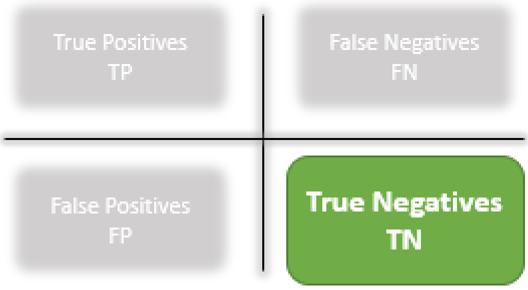


Figure3 .23: ROC Curve sample [26]

Both the metrics are widely used to judge a models performance. Which one to use PR or ROC?The answer lies in TRUE NEGATIVES.



They are helpful in unbalanced classes because TN is not present in the precision-recall equation. When there is a predominance of the negative class due to a class imbalance. The significant proportion of TRUE NEGATIVES in the dominant negative class, which provides stronger resistance to the imbalance, is not sufficiently taken into account by the metric. When the detection of the positive class is crucial, this is critical.

When it comes to diagnosing pneumonia patients, there is a significant class disparity because so few people with the disease actually have it. We want to be precise with the diagnosis of the person who has been detected so that we don't miss out on someone who has cancer and goes unnoticed.

When both classes are significant to us, the ROC equation is helpful because it takes TN or the negative class into account. Similar to the detection of dogs and cats. When a CNN model outputs whether an image is of a cat or a dog, the significance of true negatives ensures that both classes are given importance.

3.7 Ethical Considerations

AI-based pneumonia diagnosis raises a number of ethical issues that need to be carefully considered to ensure responsible and fair application. Here are several significant ethical issues for AI-based pneumonia detection.

3.7.1 Privacy and Data Security:

To preserve patient privacy and adhere to data protection laws, patient data utilised for pneumonia diagnosis, such as medical pictures and health information, must be handled with the utmost care. There should be safeguards in place to ensure secure storage and transmission of sensitive information, limit access to authorised people, and de-identify or anonymize data.

3.7.2 Bias and Fairness:

AI models used to identify pneumonia should be trained and assessed to reduce bias and assure equity across various populations. Biases based on colour, ethnicity, gender, age, or other sensitive characteristics must be taken into account and addressed because they may affect how accurate and impartial the detection system is. To lessen prejudice and prevent discriminatory effects, diverse and representative datasets should be employed.

3.7.3 Transparency and Explainability:

AI models for pneumonia detection should be straightforward and comprehensible in order to foster understanding and foster a sense of trust. It is crucial to make sure that patients and healthcare professionals can easily understand and defend the AI system's decision-making process. It is recommended to use explainable AI techniques and approaches to aid in mistake analysis and reveal the model's logic.

3.7.4 Clinical Validation and Collaboration:

AI-based pneumonia detection models should go through thorough clinical validation to guarantee its reliability, safety, and efficacy. To verify the model's performance against accepted clinical standards, identify any constraints, and assure the model's incorporation into clinical processes, cooperation between AI researchers, healthcare practitioners, and domain specialists is essential.

3.7.5 Human Oversight and Interpretation:

AI technologies should enhance human expertise, not replace it. The AI-generated results should be available to healthcare practitioners, but they should still use their clinical judgement when making final diagnosis and treatment choices. Instead of replacing healthcare workers, AI should be viewed as a tool to support and augment them, boosting their capabilities.

3.7.6 Continual Monitoring and Improvement:

Performance indicators for pneumonia detection algorithms, such as measurements of accuracy, bias, and robustness, should be regularly tracked. Based on fresh information, cutting-edge research, and input from healthcare professionals, regular updates and enhancements should be made. To ensure the model's success and resolve any potential ethical issues, it is crucial to continue a continual cycle of monitoring, evaluation, and improvement.

3.7.7 Regulatory Compliance:

When managing patient data for the detection of pneumonia, compliance with pertinent laws and standards, such as HIPAA (Health Insurance Portability and Accountability Act) or GDPR (General Data Protection Regulation), is essential. Respecting legal and regulatory regulations helps safeguard data security, protect patient privacy, and uphold moral standards when implementing AI.

Interdisciplinary cooperation between AI researchers, healthcare practitioners, policymakers, ethicists, and patients is necessary to address these ethical issues. To maximise its benefits while minimising potential hazards and societal harm, it is crucial to build policies, frameworks, and governance structures that support the ethical and responsible use of AI in pneumonia detection.

3.8 Summary

The approach and methods used to create an efficient detection system are described in the methodology part of a thesis on pneumonia detection using machine learning. It gives a thorough account of the techniques for gathering data, preparing it, extracting features, choosing a model, and evaluating it.

This chapter gives enough information to enable experiment replication and guarantee objectivity throughout the study process. The reader can grasp the methods used to construct an efficient detection system because it is providing a clear and methodical approach to pneumonia detection using machine learning.