Chapter 4

Experimental Results and Analysis

In this chapter the overview of how machine learning can be applied to pneumonia detection with experimental results and some common approaches and techniques used in this field are mentioned and explained. Pneumonia is a lung infection that can be caused by various factors, including bacteria, viruses, or fungi. Machine learning techniques can assist in pneumonia detection by analyzing medical imaging data, such as chest X-rays or computed tomography (CT) scans.

Here is a general framework for using machine learning in pneumonia detection:

Data Collection: It is necessary to compile a dataset of labelled chest X-ray or CT scan pictures. Medical experts should annotate these pictures to show if the patient has pneumonia or not.

Data Preprocessing: To assure consistency and enhance the quality of the photos, preprocessing techniques like scaling, normalisation, and noise removal may be necessary for the gathered data.

Feature Extraction: In order to produce predictions, machine learning models often need input features. Convolutional neural networks (CNNs) or manually built feature extraction approaches can be used to extract features from the photos for the purpose of detecting pneumonia.

Model Training: Training and testing sets are created from the feature-extracted and preprocessed data. You can use a variety of machine learning algorithms, including CNNs, support vector machines (SVMs), and random forests. On the labelled training data, the model is trained, and its parameters are optimised to produce precise predictions.

Model Evaluation: The testing set, which comprises of data that the model hasn't seen during training, is used to evaluate the trained model. Common evaluation criteria for judging the performance of the model include accuracy, precision, recall, and F1-score.

Analysis of Results: The efficiency of the machine learning model in detecting pneumonia is evaluated based on the experimental data. This analysis could include evaluating false positives and false negatives, comparing the model's performance with that of current diagnostic techniques, and looking into any biases or restrictions in the dataset or the model itself.

It's vital to remember that depending on the dataset, algorithm, and evaluation metrics employed, the specific experimental results and analysis for pneumonia diagnosis using machine learning may differ. The quantity and quality of the dataset, the feature selection, and the complexity of the model architecture can all have an impact on how well the model performs.

4.1 Description of Experimental Setup

We will briefly describe the essential elements normally included in such experiments in order to provide a description of the experimental setup for pneumonia identification using machine learning. Remember that depending on the research study or project, the specifics may change.

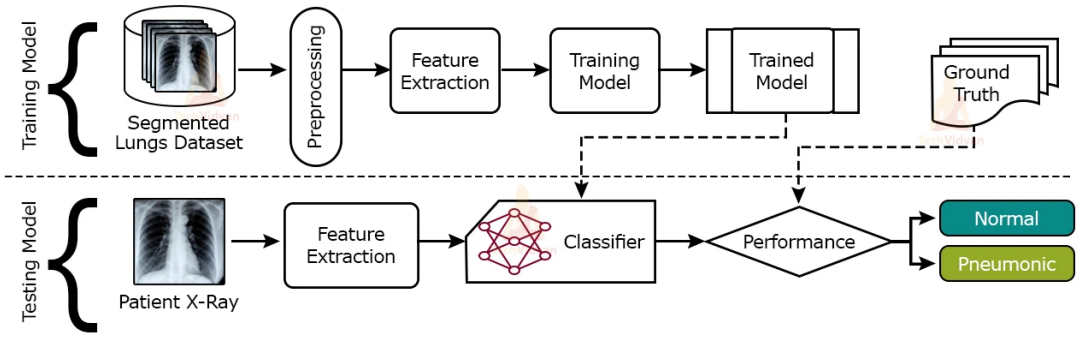

Figure4.1: Training Model & Testing Model [27]

4.1.1 Dataset Selection:

- The first step is to gather a dataset of chest X-ray or CT scan images that include cases of pneumonia and non-pneumonia (normal) cases. The dataset should be diverse and representative of the population being studied. The size of the dataset may vary depending on the available resources and the complexity of the problem.
- The first step to get result is selection of the dataset. Here in our case we will select the dataset of the pneumonia patient and normal X-Ray of the chest.
- Such data set is available in many authentic, approved, validated database management systems, firms and websites like github, md.ai etc.
- We can also customize the dataset by collecting the same from the hospitals and X-Ray laboratories.
- Here in this case if we choose the dataset from the hospital or laboratory then we couldn't compare the results of our novel model of pneumonia detection using machine learning. As compared to that, professional database provider is publically open dataset, so many researchers have fetched the results from the same dataset which is available hence we can compare our results with their result.
- When selecting a dataset for pneumonia detection, there are a few key factors to consider. Here are some guidelines to help you choose an appropriate dataset:
- **Dataset Size**: Look for a dataset that is sufficiently large to ensure that your model can learn robust patterns and generalize well. The more diverse and extensive the dataset, the better. A larger dataset will also help prevent overfitting.
- **Data Quality:** Ensure that the dataset you choose has high-quality images or medical scans of pneumonia cases. The images should be clear and properly labeled, indicating the presence or absence of pneumonia. It's crucial to have accurate ground truth annotations for training and evaluation.
- **Diversity:** Seek a dataset that covers a diverse range of pneumonia cases, including different types (e.g., viral, bacterial) and severities. The dataset should also include images from various demographics, age groups, and different imaging modalities (e.g., X-ray, CT scans).

- **Ethical Considerations:** Pneumonia can be a serious and potentially life-threatening condition. Ensure that the dataset you choose has been obtained and used in compliance with ethical guidelines and patient privacy regulations.
- **Benchmark Datasets:** Consider well-known benchmark datasets in the field of pneumonia detection, such as the ChestX-ray14 dataset, which contains over 100,000 chest X-ray images with associated labels. These benchmark datasets have been widely used and provide a basis for comparison with other studies.
- **Annotated Datasets:** Look for datasets that are already annotated with pneumonia labels or have a means to generate accurate annotations. Manual annotations by expert radiologists or clinicians are highly valuable for training and evaluation purposes.
- Some potential sources for pneumonia detection datasets include publicly available medical image repositories, research institutions, and collaborative platforms like Kaggle, Github, Md.ai or the Medical Imaging Databank. It's important to review the dataset's terms of use, licensing, and any associated publications to understand its limitations and potential biases.
- We need to validate and evaluate your trained model on independent and diverse datasets to ensure its generalizability and performance in real-world scenarios.
- To increase the generalizability of our system, we used pre-trained models from 3 different domains: ImageNet [15], ChestX-ray14 datasets [28], and Custom dataset [29].
- In comparison to the test data, the three pre-trained models we utilised are fairly big and include a lot of parameters. So, in order to prevent over-fitting, we perform some data preprocessing. For X-ray pictures of the chest, we applied image enhancement techniques.

4.1.2 Data Split:

The training set, validation set, and testing set are the three subsets that make up the dataset. The validation set is used to adjust hyperparameters and gauge model performance during training, while the training set is used to train the machine

learning model, and the testing set is used to assess the final performance of the trained model.

Training Set: Using the training set, you train your model for detecting pneumonia. Usually between 18 and 82 percent of the data from your dataset must be represented. With a larger training set, your model may learn more reliable patterns and generalise more effectively. Make sure the practise cases reflect the various degrees and types of pneumonia cases you wish to be able to identify.

Keep in mind that the training set serves as the cornerstone for your model's learning; therefore you must take great care to choose a representative and high-quality training dataset. This will improve the generalisation and accuracy of your model when applied to new data.The following recommendations should be taken into account when choosing a training set for pneumonia detection.

- Sufficient Size: Make sure your training set contains a sufficient amount of data for your model to learn from. Your model will be better able to learn strong patterns and generalise well if the training set is larger and more varied. Overfitting can also be avoided by using a larger training set. The training set typically makes up 60–80% of the entire dataset, as was already noted.
- Representative Data: The training set has to be indicative of the various varieties and degrees of pneumonia you hope to identify. It should have a wide variety of photos or medical scans that cover different ages, demographics, and imaging modalities (such as X-rays and CT scans). This makes sure that as settings and variations change, your model learns to recognise pneumonia.
- Accurate Annotations: Make that the annotations for pneumonia in the training set are precise and trustworthy. Each image or scan should have annotations stating whether or not there is pneumonia present. Expert radiologists' and physicians' manual annotations are extremely important. These annotations act as ground truth labels for your model's training and performance evaluation.
- Balance: Pay close attention to how many samples in your training set tested positive for pneumonia and how many tested negative for it. It can be difficult for the model to train and produce accurate results if there is an imbalance in the data, such as when one class (for example, pneumonia-positive patients) is

considerably more common than the other. To solve the class imbalance, try to represent both classes fairly if possible, or think about using approaches like data augmentation or resampling.

- Ethical Considerations: Make that the information in your training set was acquired and used in accordance with ethical standards and patient privacy laws. By eliminating or anonymizing any personally identifying data from the dataset, you are respecting patient confidentiality and privacy.

Validation Set: When adjusting hyperparameters, evaluating a model's performance during training, or choosing a model or set of parameters, decisions are made using the validation set. It must reflect the distribution of data in your training set. 10% to 20% of your dataset should be set aside for the validation set. This collection enables you to keep track of the model's development and spot problems like over- or underfitting.

Your validation set can serve as a reliable benchmark for monitoring and assessing your model's performance during the training phase by following these suggestions. It enables you to identify potential issues like overfitting, underfitting, or inadequate generalisation and allows you to make well-informed decisions on model improvements.Take into account the following recommendations while choosing the validation set for pneumonia detection:

- Representative Subset: The training set's data distribution should be reflected in the validation set. Similar pneumonia cases of various sorts (e.g., viral, bacterial) and severity should be distributed throughout. This helps evaluate the performance of your model on untested data and ensures that the validation set resembles the real-world scenarios it will experience.
- Size: Give the validation set a suitable amount of your dataset. The validation set typically makes up 10–20% of the entire dataset. This gives enough information to assess how well the model performed during training and to tune hyperparameters.
- Diverse Cases: In order to make sure your model is reliable and effectively generalises, include a variety of pneumonia cases in the validation set. To adequately capture the variety of pneumonia presentations, it should include

information on various demographics, age groups, and imaging techniques (such as X-rays and CT scans).

- Monitoring Model Performance: As your model is being trained, keep an eye on its development using the validation set. Analyse parameters including accuracy, precision, recall, and F1-score while evaluating the model's performance on a regular basis. This aids in your choice of model, hyperparameter tuning, or other prospective performance-related adjustments.
- Avoid Data Leakage: Make that the validation set continues to be separate from the training process. When developing a model or fine-tuning hyperparameters, avoid using the validation set because this can result in overfitting or biassed performance estimations. The validation set should only be used to assess the model's effectiveness throughout the training process.
- Cross-Validation: Sometimes, especially when the dataset is small, cross-validation methods can be used. The process of cross-validation entails segmenting the dataset into numerous subsets and iteratively executing training and evaluation on various subset combinations. This makes the estimation of model performance more reliable.

Testing Set: Your trained pneumonia detection model's final performance is assessed using the testing set. It should be totally separate from the training and validation sets to provide an objective evaluation. 10% to 20% of your dataset should be set aside for the testing set. The distribution of pneumonia cases in the testing set should match what your model will see in the actual world.

Remember to refrain from creating models or changing hyperparameters on the testing set. It should only be applied to fully assess your trained model. These suggestions will assist you in obtaining a fair and accurate assessment of how well your algorithm for pneumonia identification performs when used with brand-new, untested data.Take into account the following recommendations while choosing the testing kit for pneumonia detection:

- Independent and Unbiased: The training and validation sets should not have any relationship at all with the testing set. It should be made up of fresh, unheard-of information that your model hasn't encountered during training or

validation. This guarantees a fair assessment of how well your model performs in real-world circumstances.

- Similar Distribution: The distribution of pneumonia cases in the testing set should correspond to the distribution of instances your model would actually experience in use. A diverse range of pneumonia kinds, severity levels, and demographics ought to be present. This makes it easier to evaluate how effectively your model generalises to various situations and variants.
- Size: Set aside a suitable amount of your dataset for the testing set. The testing set makes up typically between 10% and 20% of the entire dataset, just like the validation set. A sufficient sample size guarantees statistical reliability and gives you confidence when evaluating your model's performance.
- Accurate Annotations: Make that the annotated data for pneumonia in the testing set is correct and trustworthy. Each image or scan should have annotations stating whether or not there is pneumonia present. These annotations act as ground truth labels for assessing how well your model performs.
- Confidentiality and Privacy: By deleting or anonymizing any personally identifying information from the testing set, you are respecting the confidentiality and privacy of the patient. Ensure adherence to ethical standards and data protection laws.
- Comprehensive Evaluation: Utilise the testing set to perform a thorough analysis of your model's performance. Measure different assessment criteria, including area under the receiver operating characteristic curve (AUC-ROC), recall, accuracy, precision, and F1 score. These measures shed light on the model's capability to discriminate pneumonia cases from those that aren't pneumonia.

The split ratios indicated above are not set in stone; rather, they can change based on the size and features of your dataset. If your dataset is tiny, you might think about employing cross-validation or data augmentation strategies to expand the size of your training set and enhance model performance.

We take care to maintain the distribution of pneumonia cases across the subsets while dividing your dataset. In order to ensure that each subset contains a proportional

representation of pneumonia cases with diverse features, random sampling is frequently used for this purpose.

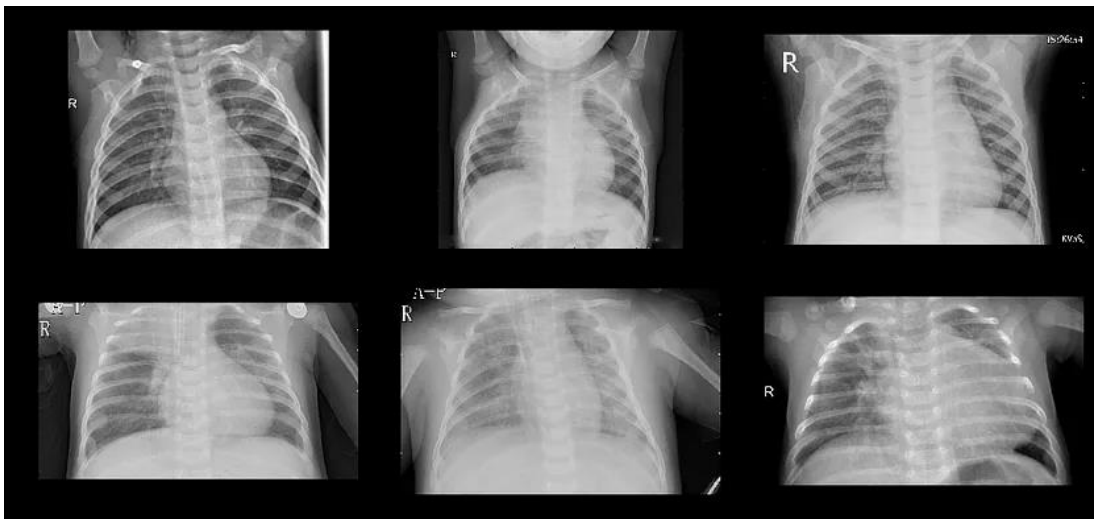

Figure4.2: Sample for testing, validation & testing dataset [27]

Finally, keep in mind to only use the testing set for final evaluation. Avoid using the testing set when developing the model or modifying the hyperparameters as this can result in overly optimistic performance estimations.

4.1.3 Data Preprocessing:

The photos are preprocessed to make sure they are in a format that the machine learning algorithms can use. In order to do this, it may be necessary to apply noise reduction techniques, normalise pixel intensities, and resize the photos to a constant resolution. Preparing the dataset for pneumonia identification through data preprocessing is crucial. Here are a few typical preprocessing methods for finding pneumonia:

- Data cleaning: Remove any irrelevant or noisy data points that could impair the performance of the model by doing data cleaning. In this step, missing numbers are handled, duplicates are eliminated, and any flaws or inconsistencies in the data are corrected.
- Image Standardisation: It's crucial to standardise the images when working with medical imaging data, such as chest X-rays or CT scans. In order to enhance the quality and consistency of the photographs, this entails scaling the

images to a constant resolution, normalising pixel values, and using any required image enhancement algorithms.

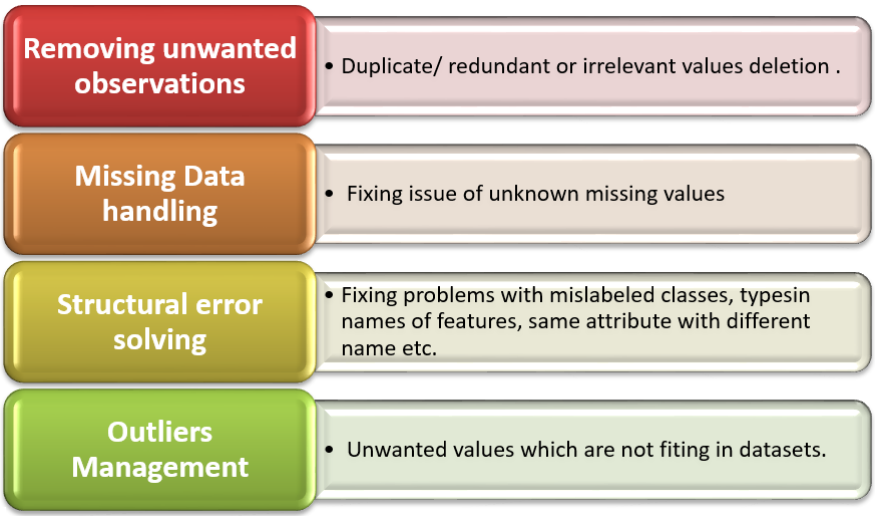

Figure4.3: Data Cleaning Steps

- Data Augmentation: The size and variety of the dataset can be artificially increased by using augmentation techniques. This can entail altering the images through the application of transformations such rotation, scaling, flipping, and noise. The model's capacity to generalise to new data is increased and overfitting is decreased with the aid of data augmentation.

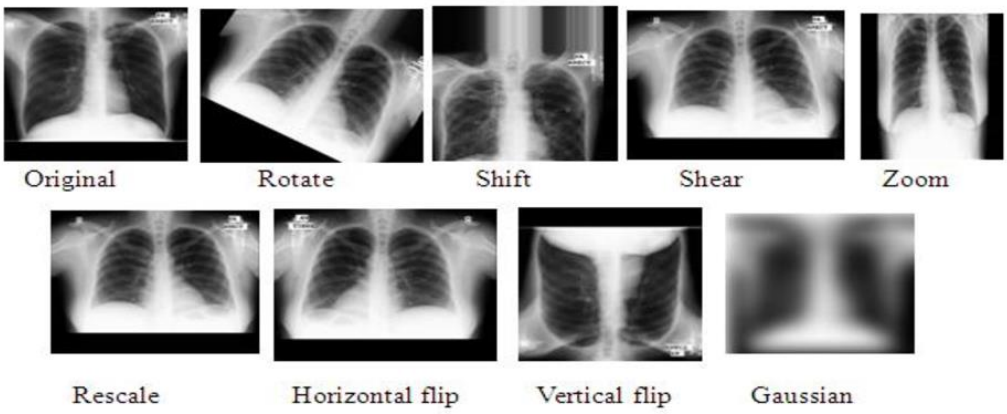

Figure4.4: Data Augmentation examples [8]

- Balancing the Dataset: If there is a class imbalance in the dataset, correct it by balancing the proportion of examples with and without pneumonia. This can be done using methods like undersampling (such as arbitrarily eliminating majority class samples) or oversampling (such as duplicating minority class

samples). To prevent bias in favour of the majority class, work to develop a balanced dataset.

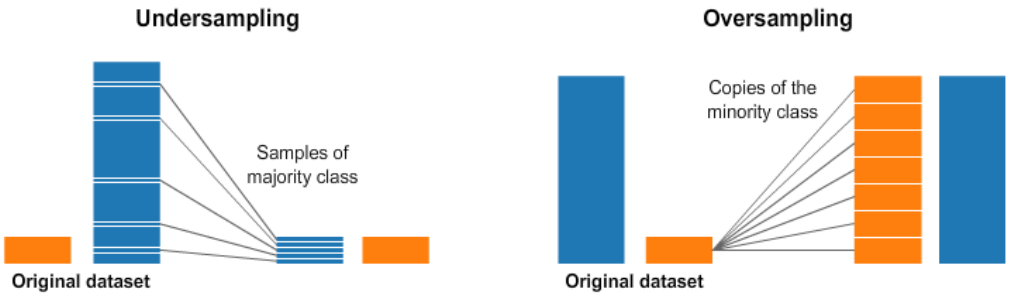

Figure4.5: Understanding Undersampling& Oversampling [8]

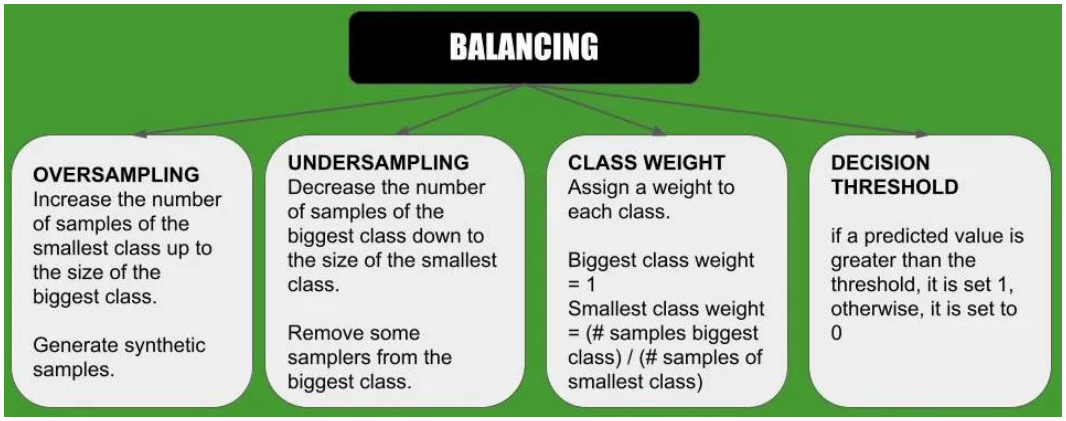

Figure4.6: Balancing classification for imbalance dataset [8]

- Feature Scaling: Normalise or standardise the input features to make sure they are on a same scale. This is known as feature scaling. This can enhance the convergence and performance of machine learning algorithms and help prevent some features from taking over the learning process.
- Train-Test Split: Split the preprocessed dataset into training, validation, and testing subsets as previously stated (train-test split). This division guarantees impartial performance evaluation by providing independent subsets for model training, evaluation, and testing.
- Label Encoding:Encode categorical labels into numerical representations appropriate for model training if the dataset contains such labels. Depending on the characteristics of the labels and the machine learning algorithm selected, common strategies include one-hot encoding or label encoding.

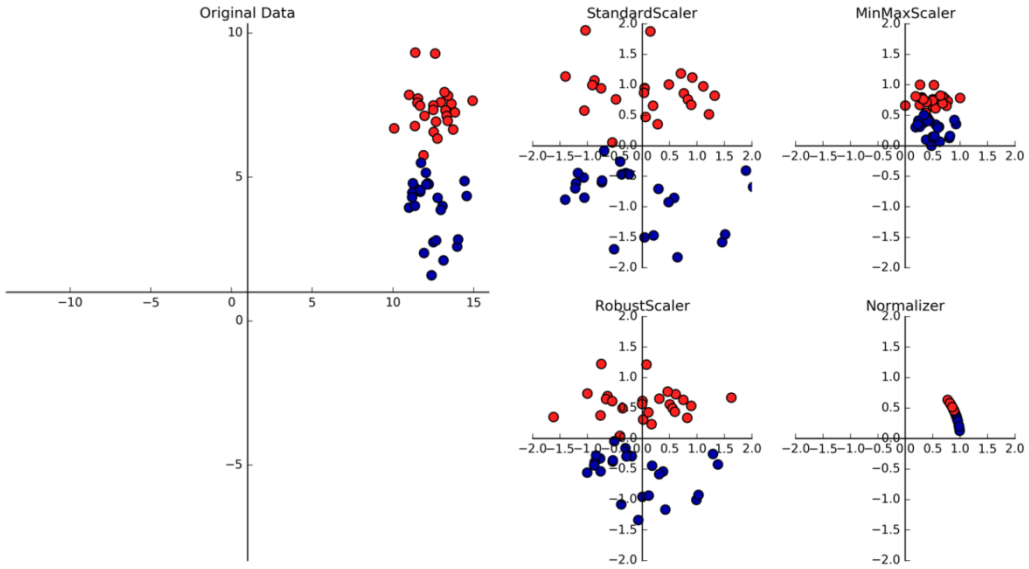

Figure4.7: Feature Normalization Example [8]

- Data Normalization: Normalise the supplied data so that its mean and variance are both zero. This normalisation step stabilises the training process and avoids the model's learning from being dominated by features with bigger scales.

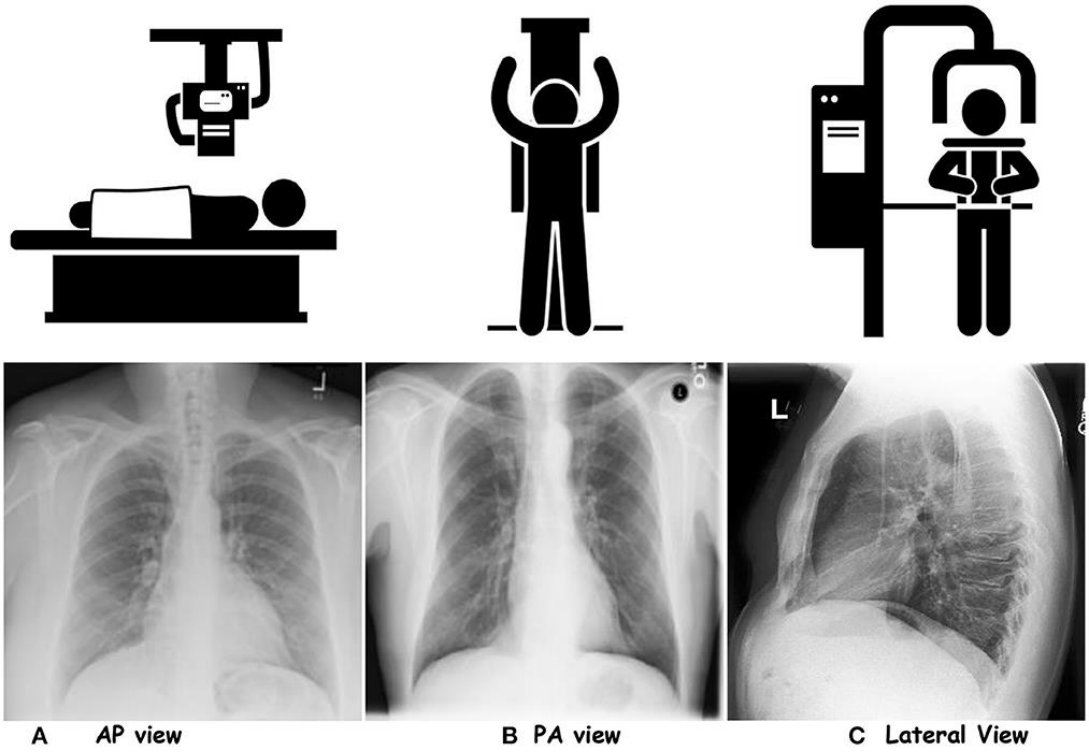

Figure4.8: Lable Encoding Example - Showcasing the chest-X rays for three projections. (A) AP view, (B) PA view, and (C) Lateral View [8]

- Handling Missing Data: If the dataset contains missing values, you may want to think about using imputation techniques to complete the data. There are many techniques that can be used, including mean or median imputation, regression-based imputation, and specialized imputation algorithms.

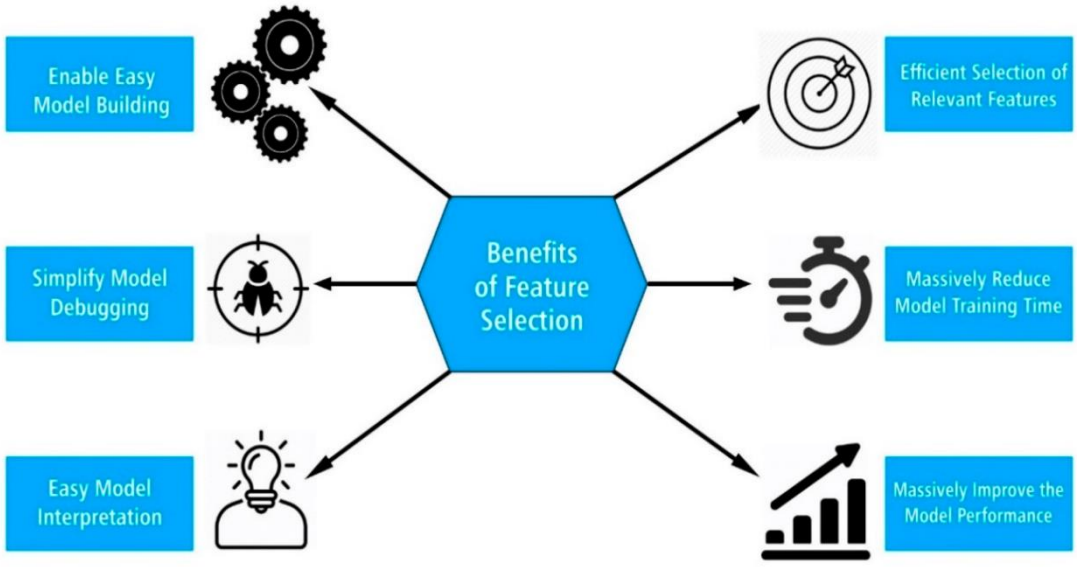

Figure4.9: Summary of the benefits of feature selection [10]

- Feature Selection:To find the most useful and pertinent features for pneumonia diagnosis, use feature selection or dimensionality reduction strategies. This aids in lowering noise, enhancing model effectiveness, and avoiding overfitting.

The specific preprocessing steps may vary depending on the nature of the dataset, the imaging modality used, and the machine learning algorithms employed. It's important to tailor the preprocessing steps to the specific requirements and characteristics of the data at hand.

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.

4.1.4 Feature Extraction:

Depending on the approach, features may be extracted manually or automatically from the preprocessed images. In the case of deep learning models, convolutional neural networks (CNNs) are commonly used to automatically extract relevant features from the images. Alternatively, handcrafted features can be extracted using techniques like edge detection, texture analysis, or shape descriptors.

For the purpose of detecting pneumonia through feature extraction, medical pictures like chest X-rays or CT scans are often mined for useful and discriminative features. Here is a description of the procedure:

- Image Preprocessing: Preprocess the medical photos to improve their quality and get rid of noise and artefacts before feature extraction. Techniques like scaling, normalisation, noise reduction, and contrast enhancement might be used for this.
- Region of Interest (ROI) Selection: Find the area in the image that contains the lungs or is impacted by pneumonia. This is the relevant region of interest. This stage assists in concentrating the feature extraction process on the areas that are most informative.
- Feature Extraction Techniques: To extract pertinent elements from the ROI, use machine learning-based feature extraction approaches. Among the methods that are frequently utilised are:
    - Intensity-based Features: Extract statistical information from the histogram, such as the mean and standard deviation, to show the distribution of pixel intensities inside the ROI.

  - Texture Features: To identify the texture patterns in the image, use texture analysis techniques as Haralick features, Gabor filters, or local binary patterns. These qualities, including coarseness, contrast, and homogeneity, can be captured by these features.
  - Shape and Contour Features: To gather the shape and contour details of abnormalities in the lung region, extract shape-based parameters such area, perimeter, circularity, or compactness.
  - Local Binary Patterns (LBP): LBP is a texture descriptor that describes how a core pixel interacts with its surrounding pixels. It is frequently utilised in the identification of pneumonia and can capture local texture patterns.
  - Deep Learning Features: To extract high-level features from the ROI, you can also use pre-trained convolutional neural networks (CNNs), such VGG, ResNet, or Inception. These networks have the capacity to acquire intricate and esoteric properties that are beneficial for detecting pneumonia.
- Feature Selection and Dimensionality Reduction: To choose the most pertinent and discriminative features from a high-dimensional extracted feature set, use feature selection techniques like mutual information, chi-square, or recursive feature removal. Additionally, it is possible to lower the dimensionality of the feature space while still maintaining the most crucial data by using dimensionality reduction techniques like Principal Component Analysis (PCA) or t-SNE.
- Classification: The collected features are then sent into a machine learning technique for the detection of pneumonia, such as logistic regression, support vector machines (SVM), random forests, or deep learning models. The labelled data is used to train the model, and the learned classifier can be used to determine whether or not pneumonia is present in a new image.

The specific attributes of the imaging data, the available computational resources, and the application's performance requirements all influence the choice of feature extraction algorithms. It's critical to test out several approaches and judge how well they capture important data for accurate pneumonia detection.

4.2 **Model Selection**

Different machine learning algorithms can be employed for pneumonia detection, such as CNNs, support vector machines (SVMs), random forests, or ensemble methods. The selection of the model depends on the complexity of the problem, available computational resources, and prior knowledge about the dataset.

The Deep Neural Network (DNN) problem is improved by the CNN model. Only one-dimensional data is used in the DNN model. Image data, however, only has two dimensions. When switching to one-dimensional data from two-dimensional data, DNN causes a significant loss issue. The CNN model has been put forth, and it may be trained using image data by applying filters to the image that have a specific size. We used pre-trained models that were trained on data from ImageNet, Chest X-ray14 databases, and custom datasets, which represent three distinct domains. ResNet152, Dense121, and ResNet18 are the three models.

4.2.1 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 Figure4.10. 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. One-dimensional data is flattened into a fully connected layer (FC) by this procedure.

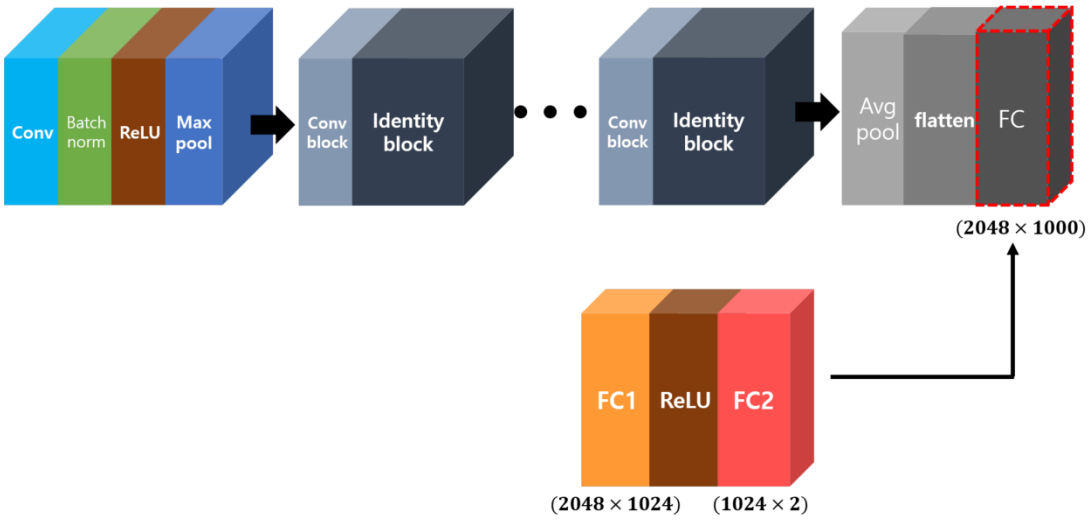

Figure4.10: PretrainedResNet with changed classifier on ImageNet [10]

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 ResNet improves with layer depth. ResNet18 employs 3x3 filters while ResNet152 employs 1x1, demonstrating structural differences.

4.2.2 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 (shown in Figure 4.11) are connected in a feed-forward fashion. Each DenseBlock's convolution layer combines 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.

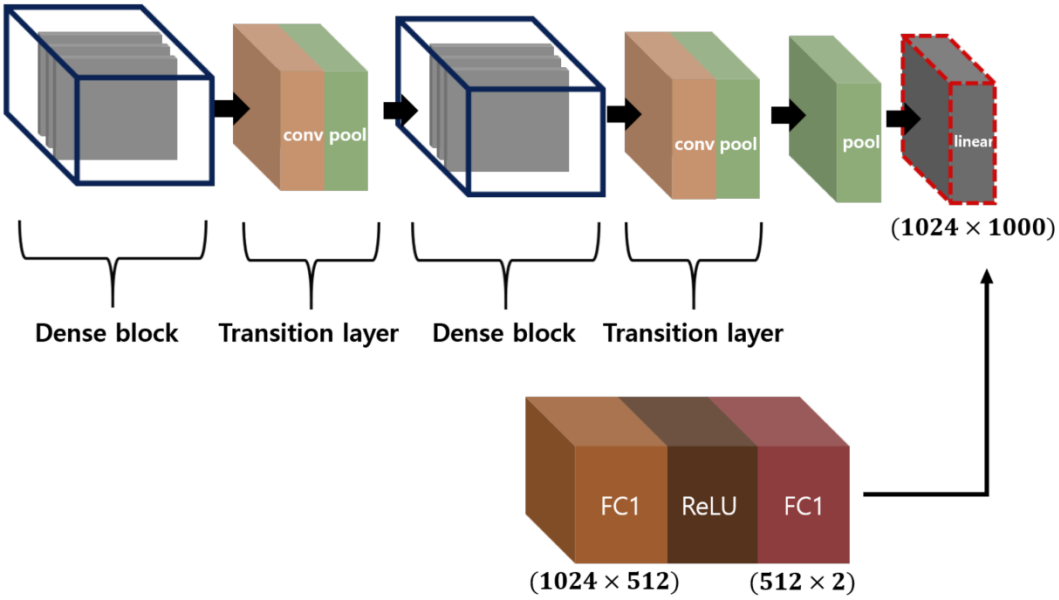

Figure4.11: DenseNet - PretrainedDenseNet with changed classifier on Chest X-ray14 of NIH [11]

Learning is made simpler by DenseNet, which compensates for the loss of initial information as the layer depth increases and brings each layer closer to the gradient generated from the loss function and input. 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.

4.3 **Transfer Learning**

Transfer learning is the process of using models that have already been trained on huge datasets to perform tasks in related industries. Our aim is to improve pneumonia diagnosis by using chest X-ray pictures. So, with a similar goal, we want to deploy

CNN-based models. Each of the pre-trained classification models we utilised was trained using 1,281,167 ImageNet data points with 1000 classes, 112,120 NIH chest X-ray data points with 14 classes, and 43,956 custom data points with 11 classes.

Despite the paucity of data needed to fully train the model using a pre-trained model, the new approach can diagnose pneumonia well. While we continue to use the weights that the layers of the pre-trained models have, we adopt a way to adjust the final fully linked layer that best fits our job.

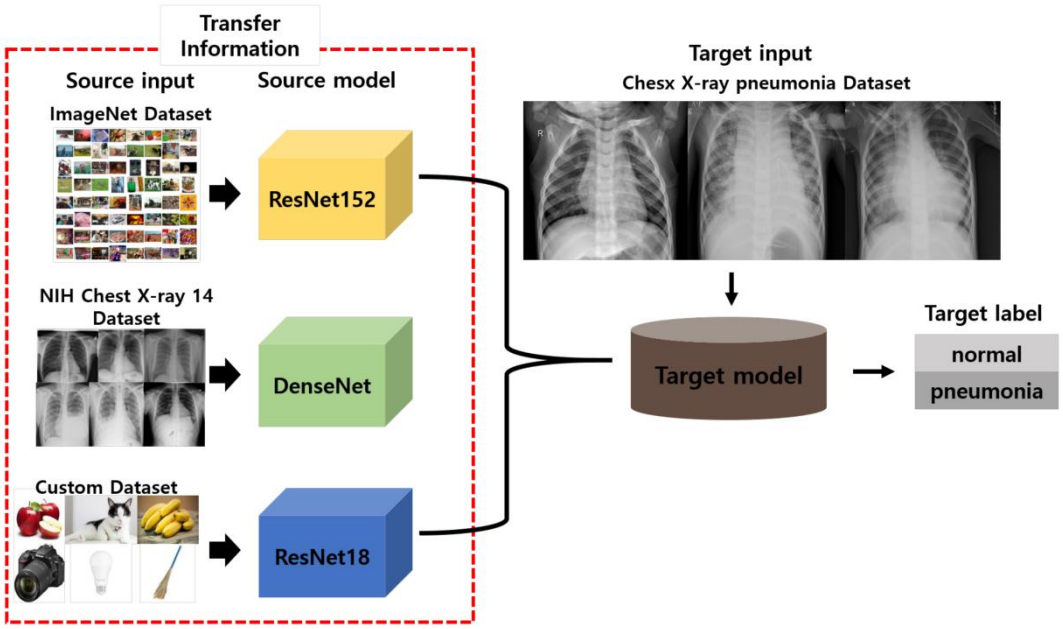

Figure4.12: Overall process of transfer learning [10]

One CNN model that has been trained on ImageNet and divided into 1000 classes is ResNet152 [10]. As shown in Figure 4.12, we modified the pretrained ResNet152's end layer to serve as the new classifier, allowing it to be divided into two classes: photos indicative of pneumonia and images indicative of normalcy. However, if the data domains utilised by users and those used for pre-training are sufficiently different, there is a risk of training degradation when utilising pre-trained models for transfer learning.

4.4 **Attention Mechanism**

A sequence-to-sequence translation model based on an encoder-decoder was where the attention mechanism was initially introduced [30]. Short summaries of context-based information from the input sequence are permitted. Self-attention

allows for direct long-distance interdependency by focusing attention on one context rather than many. The attention mechanism is frequently employed in computer vision to support the CNN model.

It concentrates on a certain trait that is crucial for classification. The attentiveness mechanism's primary goal is to force the model to pay attention to crucial portions of the input data [30].The attention mechanism applies a weight filter with the same size as the original image and searches for correlations with distant pixels. Finding areas that differ from typical X-rays when identifying pneumonia, i.e., areas with a lot of information for classifying pneumonia, is helpful.

Throughout the scan, our attention will be on the region surrounding the lungs, and pneumonia may be detected when this region seems fuzzy. It is crucial to know precisely which area of the image to examine when diagnosing a particular condition, and attention is efficient since it automatically locates and concentrates on key areas of the image. In the area of computer vision, we suggest using feature vectors as a target for the attention mechanism.

4.4.1 Attention Mechanism

The identification of associations with pixels in remote areas is a challenge for previous convolution operations. Only specific local fields contain each of the features. Self-Attention [31] was suggested as a solution to these issues. Self-Attention computation is comparable to convolution procedures, however it is simple to recognise relationships with distant regions..

A graphic illustration of self-attention is shown in Figure 4.13. An input is separated into Query, Key, and Value with the same shape as (BsizeLQueryDmodel). In that sequence, these refer to batch size, query length, and model dimension. Here, we employ Multi-Head-Attention, which executes Self-Attention from several angles. Prior to passing the score to the soft-max layer, we first obtain the mapped score from the convolution operation of the query and key (S=$softm(Q \otimes KT)$).

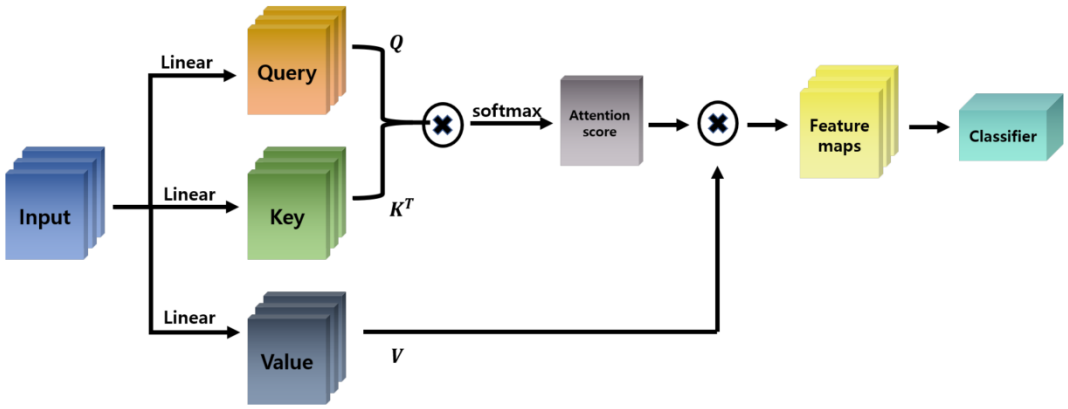

Figure4.13: Overall process of Self-Attention [31]

The output of the attention operation is the convolution operation of the Score and Value (O=$S \otimes V$). Classification is performed in a fully connected layer using the final output of Self-Attention.

4.4.2 SENet

SENet [20] makes use of the channel interdependency to raise the calibre of the network-generated feature representation. Recalibrating the features obtained from the convolution process with the importance per channel is the aim of SENet.

The CNN channel weights are selectively adjusted by SENet using channel-based attention. By adding SE blocks after the convolution procedure, SENet boosts the performance of the CNN model. The two components of this concept are squeezing and excitement. The entire information is compressed during the squeeze operation in order to embed global information.

Excitation operation scales the importance of each feature map. In this step, the squeezed important information is recalibrated. (shown in Figure 4.14) These two parts are tied together and called SE blocks. The first step is a simple convolution operation that converts the dimension of ($H' \times W' \times C'$) to ($H \times W \times C$). It converts two-dimensions of ($H \times W$) feature maps of C channels into (1×1)-sized C feature maps. Using a Global Average Pooling (GAP), each two-dimensional feature map is averaged to one value. It compresses the global information from each channel. This operation was referred in Equation (1), where (i,j) represents the output of the

convolutional operation $X(H' \times W' \times C')$ with c filters. In (i,j), i and j are the mean indices of H and W, respectively

$$z_c = \frac{1}{H\ X\ W} \sum_{i=0}^{H} \sum_{j=0}^{W} (u_c(\mathrm{i}, \mathrm{j})) \qquad \ldots (1)$$

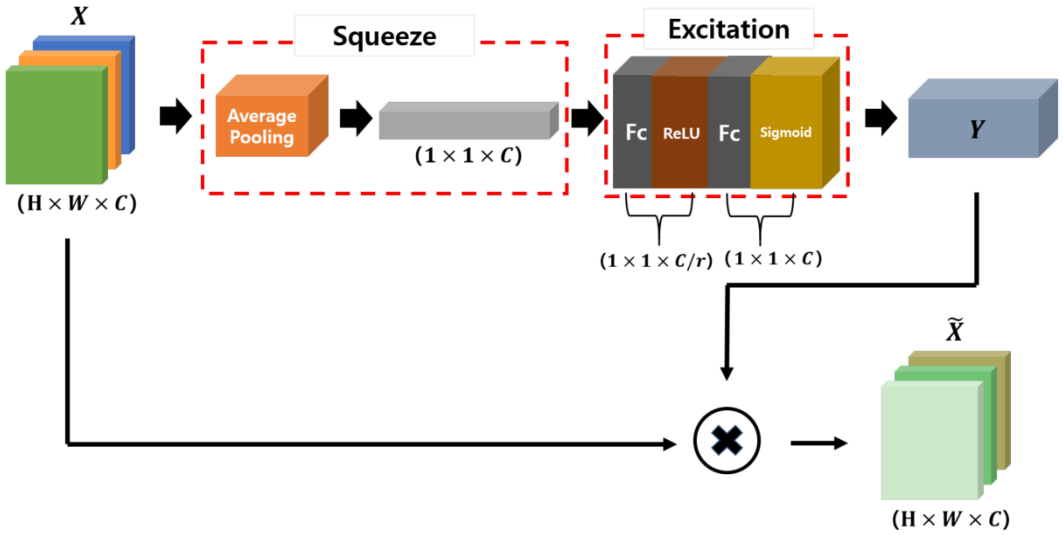

Figure4.14: Overall process of SENet [20]

The excitation operation (shown in Equation (2)) adjusts the fully linked layer and the non-linear function to compute channel-wise dependency. In order to reduce the quantity of computing, dimension reduction is carried out in the middle. Apply a non-linear function at the end to the fully connected layer's outputs.

$$s=Sigmo(W2ReLU(W1)) \qquad \ldots(2)$$

W_1 and W_2, respectively, refer to the fully connected layers. After all operations are performed, multiply each of the C feature maps before GAP and print them out (shown in Equation (3)).

$$\tilde{x}=s_c \bullet u_c \qquad \ldots(3)$$

As a result, a feature map of where all of these values are scaled by the importance of channel with values between 0 and 1.

4.4.3 Efficient Channel Attention

The use of the self-attention mechanism has significantly enhanced the performance of earlier CNN models. The model's intricacy is too great, though. Two FC layers are used by SENet to calculate weights. To lower the amount of computing, dimension reduction is carried out through two fully connected layers.

Non-linear activation functions are employed to emphasise significant characteristics. In ECA [21], instead of employing two FC layers, the channel weight is created by performing channel attention without dimensionality reduction using 1D convolution of kernel size k (see Figure 4.15) A filter using 1D convolution that only takes into account the immediate local area reduces the complexity of the model.

ECA increases the harmony between complexity and performance. Cross-channel interaction via the channel dimension *C* function determines the kernel size k. Given that $C=ø(k)$ is the function between *k* and *C* and that *C* is often set to be a multiple of 2, it is possible to set *C* as a non-linear function, such as $C= ø(k)=2^{\Upsilon x k-b}$. Equation (4) illustrates how this function can be represented in terms of *k*. In order to apply ECA modules to CNN models, existing SE blocks are swapped out.

$$k = \varphi(C) = \left|\frac{log_2 C}{\gamma} + \frac{b}{\gamma}\right| \qquad \ldots (4)$$

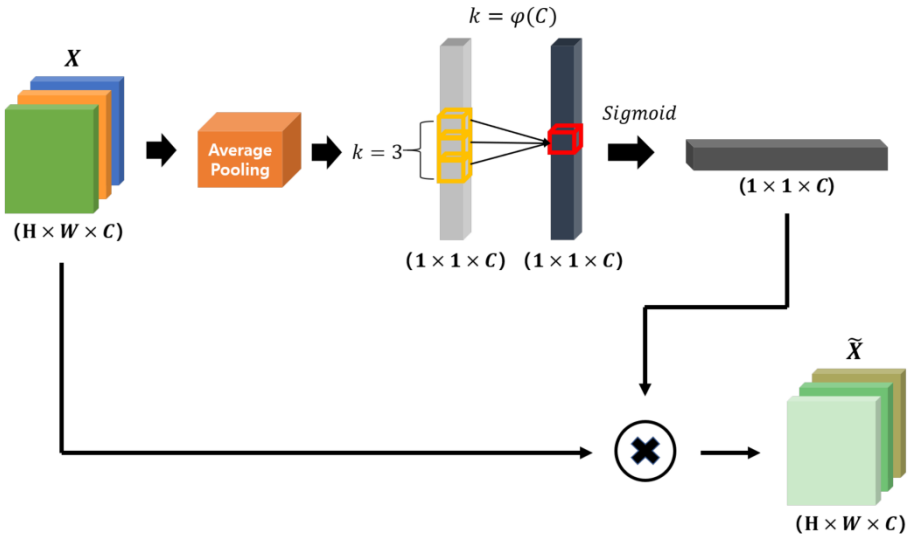

Figure4.15: Overall process of ECA [21]

4.5 **Datasets& Algorithm**

For the training, we employed three different domains of datasets. We employed the structure of pre-trained models that were each trained on ImageNet [15], Chest X-ray14 [28], Custom datasets [29]. The ImageNet dataset is well known for being easily accessible and large. It contains 1,281,167 data classified as 1000 classes.

The Chest X-ray 14 dataset is the largest publicly available chest X-ray dataset released by the National Institute Of Health(NIH). It contains 112,120 X-ray images of 30,805 patients and classified into 14 chest pathology labels, including pneumonia. The Chest X-ray 14 dataset has a similar domain as our test dataset. The Custom dataset is classified into 11 classes and contains a total of 43,956 images. This dataset was used to demonstrate the generalization ability of our framework. With our framework, even pretrained models in unrelated domains can also be used to implement the target task.

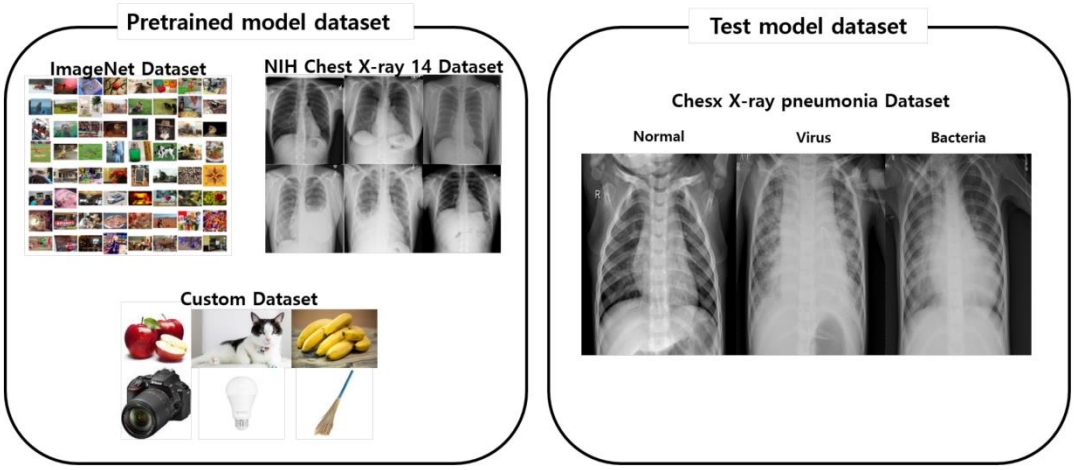

Figure4.16: A sample of dataset [15]

For the testing, we used publicly available chest X-ray images collected from pediatric patients between 1 to 5 years old from Guangzhou Women and Children's medical center [32], which was not used for the pre-trained model. This dataset was collected by researchers that have achieved approvals from the Institutional Review Board (IRB) and Ethics Committee on data collection and experimentation. The dataset consists of 1583 normal images and 4273 pneumonia images. Of the 5856 images, 5216 images are used for framework training, and 16 images are used as data to evaluate models during training.

Category	Train	Validation	Test
Normal	1341	8	234
Pneumonia	3875	8	390
Total	5216	16	624

Table4.1:The splits of the pneumonia datasets from Guangzhou Women and Children's Medical Center [32]

Class	Training Set	Testing Set	Total
Normal	3167	708	3875
Pneumonia	3188	687	3875
Total	6355	1395	7750

Table4.2:Number of samples for train and test split for VIYU Model.

A comparative and analytical summary for the different Deep-CNN method results and accuracy is given in Table 4.2

Ref.	Dataset	Train	Test	Classifiers/Models	Year	Accuracy
[51]	Chest X-ray images (augmented)	9000	419	Different pre-trained CNN model	2020	0.98
[52]	Chest X-ray images	5232	624	Ensemble model	2020	97.4
[53]	Chest X-ray images	5216	640	CNN model	2021	90.78
[54]	Chest X-ray images (rearranged)	3722	2134	Proposed CNN model	2019	0.9373
[56]	COVID-19 X-ray images	90%	10%	Ensemble learning deep	2020	99
[55]	Chest X-ray images	80%	20%	Ensemble model	2021	86.3
[57]	Chest X-ray images (rearranged)	4686	1170	Ensemble deep learning model	2021	97.81
[59]	COVID-19 and CXR images joined data	6086		Ensemble deep learning model	2021	95.05
[60]	CT chest COVID-19 images dataset	-	-	Ensemble deep learning	2021	93.57
[58]	Heart disease dataset	800 records		Ensemble deep learning	2020	91
VIYU	Chest X-ray images	6355	1395	Deep Learning – VIYU Model	2023	98.08

Table4.3:Summary of Deep-CNN related work

The size of the validation set was too small, so it was used for reference only, and cross-validation was not performed separately. Cross-validation is used to avoid overfitting. However, this process takes a lot of time. So, we chose not to use the cross-validation procedure. Instead of cross-validation, we used the validation set to determine the optimal number of epochs to avoid overfitting. Finally, 624 images were used to conduct tests on the framework (shown in Table 4.1). Pneumonia is divided into bacterial and viral pneumonia. However, we focus on the binary classification task of classifying normal and pneumonia by defining two pneumonia categories as one pneumonia category due to lack of data.

Images of the normal class are much smaller than images of the pneumonia class. We used F-score to evaluate our approach properly given the class-imbalanced dataset. Although we used a class-imbalanced dataset, there was no major problem with model performance. A sample image of the dataset is shown in Figure 4.16.

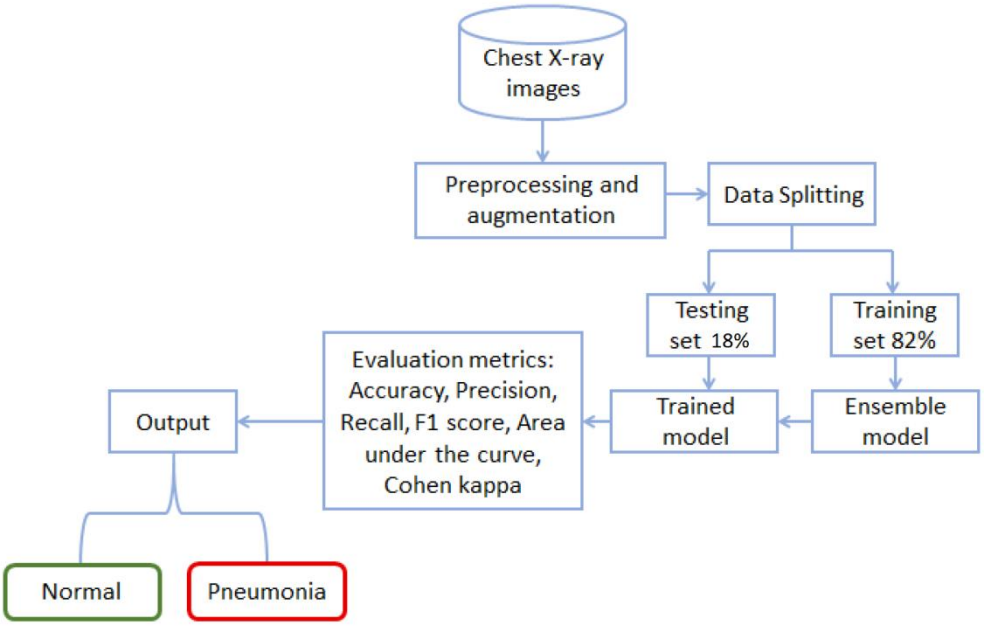

Figure4.17: The workflow of the proposed methodologys

4.6 **Result, Evaluation &Comparision**

Once training is complete, the performance of the trained model is evaluated using the testing set. The model's predictions on the testing set are compared with the

ground truth labels to calculate evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the model's ability to correctly classify pneumonia cases

4.6.1 Result & Evaluation

Our method's main objective is to increase the precision with which pneumonia is identified from chest X-ray pictures. To test the generalizability of our system, we created three different pre-trained model types depending on different domains. In order to validate the effectiveness of the categorization of pneumonia and normal, feature vectors were extracted using the backbone structures of pre-trained ResNet152, DenseNet121, and ResNet18.

The feature vector of the model has three dimensions: 2048, 1024, and 512. We combine each feature vector with the same 1024 embedding dimensions to test our system. Self-Attention, SE, and ECA were the three types of attention techniques we used. The Guangzhou Women and Children's Medical Centre provided 5216 photographs for the suggested method to train on, and the remaining 624 images were utilised to test the model's effectiveness. Each of the pre-trained models had its last layer redesigned with a new classifier that could distinguish between photos of pneumonia and typical images. We used the Adam optimizer [33] and the NLL (Negative Log-Likelihood) loss function for the experiment. After every 10 epochs, the learning rate was reduced by =0.1 from its initial value of 0.0001.

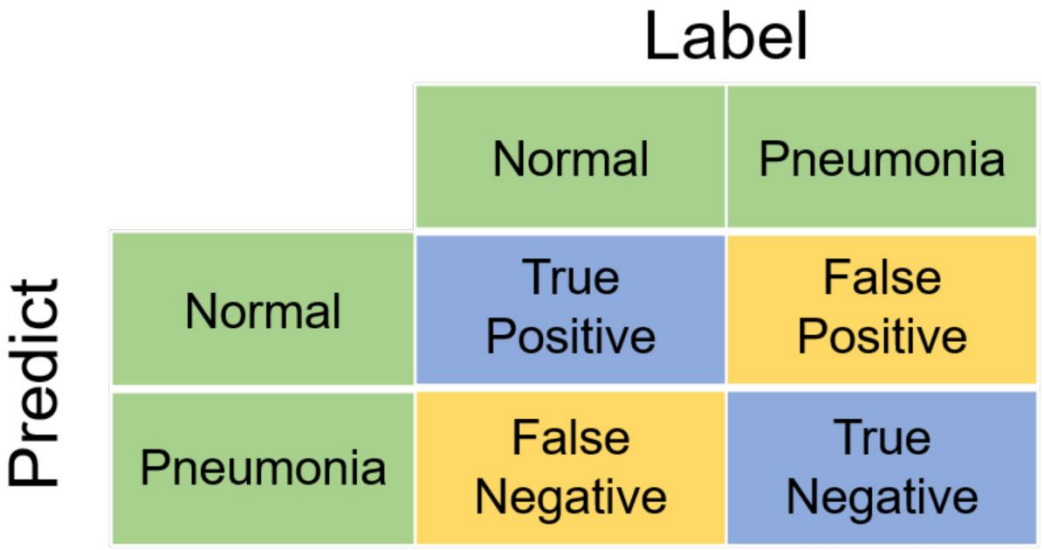

Figure4.18: Matrix of relationships between actual and model-predicted answers

The epoch shows how many training sessions were used to prepare the datasets. The neural network divides its computations into two phases: forward pass, which calculates each layer's weight from input to output, and backward pass, which repeats the calculation process and alters the weight. One epoch will have been done after the two processes are finished. 30 training epochs were used to train all models. In Table 4.2, the outcomes for each model are displayed. As evaluation metrics, we employed Recall, Precision, Area Under the Curve (AUC), and Test Accuracy.

By using Figure 4.18 and Equation (5) to calculate the Recall and Precision, the following can be discovered. The binary classification we carried out has a total of four possible outcomes. False Positive, True Negative, True Positive, and False Negative are the categories. Whether the model properly or erroneously predicted something can be expressed as True or False. The model's prognosis of normal is Positive, whereas its prediction of pneumonia is Negative.

$$\text{Recall} = \frac{TP}{TP + FN} \ldots (5)$$

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

Fall-out and Recall indices are used by AUC. Fall-out is calculated as follows (as given in Equation (6)), and the area of the resulting graph, which has Fall-out on the x-axis and Recall on the y-axis, is known as the AUC. This graph is known as the Receiver Operating Characteristic (ROC) curve. When the data labels are not balanced, the model's performance is shown by the F-score, which is the harmonic mean of Precision and Recall (seen in Equation (7)). Test accuracy (represented in Equation (8)) denotes classification accuracy using test datasets.

$$\text{Fallout} = \frac{FP}{TN + FP} \ldots (6)$$

$$\text{F Score} = 2 \text{ x } \frac{Precesion\ X\ Recall}{Precesion + Recall} \ldots (7)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \dots (8)$$

Model	Data	Epoch	Recall (%)	Precision (%)	AUC (%)	F-Score	Accuracy (%)
VGG-16	-	25	97.38	98.41	97.92	0.979	97.93
Inception-V3	-	25	95.54	97.46	96.57	0.964	96.58
ResNet50	-	25	97.51	98.15	97.87	0.978	97.87
ResNet152	ImageNet dataset	25	98.97	93.46	93.72	0.961	95.03
DenseNet121	NIH dataset	25	98.72	94.13	94.23	0.964	95.35
ResNet18	Custom dataset	25	98.97	93.24	93.50	0.960	94.87
Self-attention	-	25	98.72	96.46	93.72	0.976	95.03
ECA	-	25	98.21	95.99	95.68	0.971	96.31
SE-Attention	-	25	98.46	96.24	96.03	0.973	96.63
VIYU	-	25	98.91	98.10	98.22	0.974	98.08

Table4.4: Comparison of different CNN, transfer learning models and models with different attention mechanisms on the test dataset in terms of performance metrics.

The proposed models were evaluated by metrics such as recall, precision, AUC, F-score and accuracy. VIYU the stae-of-the-art model achieved accuracy, F-score, AUC, precision and recall of 98.08%, 0.974, 98.22%, 98.10%, and 98.91%, respectively. VIYU the state-of-the-art model achieved the best performance in terms of accuracy, followed by SE and ECA.

4.6.2 Comparison

The experimental results are analyzed to understand the performance of the machine learning model. This analysis may include comparing the model's performance with existing diagnostic methods or other machine learning models, identifying patterns or limitations in the predictions, and assessing false positives and false negatives. Additionally, visualization techniques can be used to interpret the model's decision-making process and highlight regions of importance in the images.

In this part, we evaluated how well our models performed against current CNN-based models that support the pneumonia diagnosis. In order to compare the models, we used a dataset of chest X-rays from the Guangzhou Women and Children's Medical Centre and the same domain. Table 4.3 provides a summary of the findings.

A pre-trained InceptionV3 model that was trained on ImageNet was employed by Kermany et al. [32]. Instead of using random initialization, the model's weights from the pre-trained InceptionV3 were used. By unfreezing and updating the pretrained weights on chest X-ray pictures as fine-tuning, they learned the model. They achieved recall and classification accuracy of 92.8% and 93.2%, respectively.

The DenseNet121 [11] design was employed by Cohen et al. [34] and demonstrated to work effectively on chest X-ray images with a CheXnet DenseNet121 model [28]. With default parameter settings of b1 = 0.9 and b2 = 0.999, a learning rate of 0.001, and a learning rate degradation of 0.1 once the validation accuracy converged, they employed Adam optimisation. They obtained a 98.4% AUC.

Model	Recall (%)	Precision (%)	AUC (%)	F-Score	Accuracy (%)
Kermany et al. [32]	93.2	-	96.8	-	92.8
Cohen et al. [34]	-	-	98.4	-	-
Rajaraman et al. [35]	96.2	**97.7**	**99.3**	0.970	96.2
Sahlol et al. [24]	87.22	-	-	-	94.18
So-Mi Cha et al.[39]	**98.46**	96.24	96.03	**0.973**	**96.63**
Ayan and Über [37]	82	-	-	-	87
Sharma H. et al. [38]	-	-	-	-	90.68
Saraiva et al. [36]	94.85	95.72	-	0.953	95.07
M. Baraiya (VIYU)	**98.91**	98.10	98.22	**0.974**	**98.08**

Table4.5: Comparative results for other models on same test dataset. Bold numbers indicate best performance.

Rajaraman et al. [35] used customized models based on a pre-trained CNN model as the feature selector. They selected specific regions of interest (ROI) on chest X-ray images to perform the classification of pneumonia and normal images. They evaluated models with two types of dataset, original data as baseline and cropped ROI data. The best model was customized VGG16 and we outperformed their model in terms of accuracy, F-score, and recall. Cohen et al. and Rajaraman et al. attempted to detect pneumonia with the customized CNN model.

Sahlol et al. [24] used a MobileNet as the feature extractor and the AEO algorithm as the feature selector. The AEO algorithm finds only the relevant features from a lot of features that are extracted from a MobileNet. So-Mi Cha et al. [39] used a SE-Attention model algorithm & they achieved best results in recall, precesion, F-score and accuracy, of 98.46%, 96.24%, 0.973 & 96.63 respectively.

Saraiva et al. [36] used two neural networks, the multilayer perceptron and neural network to detect and classify the pneumonia. They achieved best results in recall, precision, F-score and accuracy, of 94.85%, 95.72%, 0.953% and 95.07%, respectively. Ayan and Über [37] also used deep-learning-based methods Xception and VGG16 for pneumonia classification. The VGG16 model outperformed Xception with an accuracy of 87%.

Sharma et al. [38] proposed two CNN architectures that were designed from scratch with or without a dropout layer. They used deep CNN architectures as the feature extractor. The accuracy of the results was 90.68%. Despite other papers not providing as detailed results as ours for evaluation, we outperformed other methods in terms of classification accuracy, recall and F-score.