Electronic Letters on Computer Vision and Image Analysis 24(2):143-158, 2025

Integrated CNN Model for Multi-disease Classification Through Chest X-ray Images

Diwakar Diwakar* and Deepa Raj+

* *Department of Computer Science, BBA University, Lucknow, India, 226025*
+ *Department of Computer Science, BBA University, Lucknow, India, 226025*

Received 25th of november, 2023; accepted 17th of September 2025

Abstract

Rapid and accurate detection of chest pathologies from radiographs is a challenging yet critical task in medical diagnostics. This paper proposes a hybrid deep learning model that integrates two convolutional neural networks (VGG16 and VGG19) with a Principal Component Analysis (PCA) based feature fusion for multi-disease classification on chest X-ray images. The model is trained and evaluated on a balanced dataset comprising four classes namely COVID-19, Pneumonia, Tuberculosis, and Normal healthy cases using transfer learning to leverage pretrained knowledge. The fused CNN features are compressed via PCA, which reduces dimensionality and boosts efficiency. Experimental results demonstrate a high classification accuracy of approximately 97.5%, with a notably low training time and an average inference speed of about 0.036 seconds (36 ms) per image. Such performance indicates suitability for real-time diagnostic use. To enhance interpretability, the framework employs LIME to generate visual explanations, highlighting the regions of each X-ray that most influence of the model predictions. The proposed integrated approach thus offers a scalable, fast, and explainable solution for automated chest disease screening, underlining the potential of combined deep learning and statistical techniques in medical image analysis.

Key Words: Convolutional neural network, VGG16, VGG19, Principal component analysis, Transfer learning.

1 Introduction

In recent years, Convolutional Neural Networks (CNNs), a type of Artificial Intelligence (AI) approach, have played a vital role in medical imaging for tasks such as classification, localization, segmentation, registration, and many more. The ability to diagnose diseases has become increasingly dependent on the development of medical imaging technologies including Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and X-ray imaging etc. One of the most punctual and most broadly utilized medical diagnosis strategies is the X-ray, which depicts the interior body parts in black and white shades. Deep learning has already proved its ability to identify images with human-like accuracy. In especially, image classification using CNNs was quite successful. The primary benefit of CNNs is that they can extract more meaningful features from the entire

Correspondence to: diwakarmsccs0@gmail.com

Recommended for acceptance by Angel D. Sappa
ELCVIA ISSN:1577-5097
DOI:https://doi.org/10.5565/rev/elcvia.1804
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

image than can handcrafted features (like the Histogram of Oriented Gradients (HOG), Speeded Up Robust Feature (SURF), Scale Invariant Feature Transform (SIFT), etc). CNN or ConvNet architecture is a multi-layered feed-forward neural network built by sequentially stacking several hidden layers with learnable biases and weights on top of one another. Three layers are commonly present in CNNs: the convolutional layer, the pooling layer, and the dense or fully connected layer. The conv layer has a set of filters or kernels that perform convolution on the input and send the outcome to the following layer. Typically, the filter's size is less than the original image and each kernel produces an activation map after it convolves with the image. The dimension of the feature maps is reduced using pooling layers. Whereas, fully connected layers are the final few levels of the network, where every neuron in one layer connects with every neuron in every other layer. Numerous Deep Neural Network models have already been created to distinguish diseases through Chest X-rays (CXRs). Many scientists and researchers are engaged in this area and trying to make improvements to employ it in real-time clinical applications.

What sets this work apart is its integrated dual-CNN approach combined with PCA-based feature reduction. Unlike prior studies that often rely on a single deep learning model or standard transfer learning, this paper leverages two complementary CNN architectures (VGG16 and VGG19) simultaneously. The feature maps from both networks are fused (using a max-fusion strategy) to capture richer information from the X-ray images. Introducing a PCA step after fusion is a novel twist, it condenses the fused feature space to a lower dimension, preserving important patterns while eliminating redundancy. This hybrid design (CNN + PCA) yields a model that is not only highly accurate but also computationally efficient. Furthermore, the approach emphasizes explainability by incorporating Local Interpretable Model-Agnostic Explanations (LIME) for interpretability, an aspect often overlooked in related work. In summary, the novelty lies in combining multiple deep networks with statistical dimensionality reduction and explainable AI techniques to improve chest X-ray classification.

1.1 Problem Statement

The medical community faces the challenge of rapidly and accurately diagnosing multiple chest diseases from X-ray images. Chest radiographs are among the most common diagnostic tools, but interpreting them to distinguish conditions such as COVID-19, Pneumonia, and Tuberculosis from normal cases is difficult due to overlapping visual features. This can lead to misdiagnosis or delayed treatment. Therefore, there is a pressing need for an automated, reliable classification system for chest X-rays that can identify different diseases with high precision. In this paper, the above problem is addressed by developing a method that achieves fast and accurate multi-disease detection from chest X-rays, suitable for real-time use in clinical or resource-constrained settings where expert radiologists may not be readily available.

1.2 Key Contributions

- Combined VGG16 and VGG19 features using max-fusion to enhance representational power.
- Applied Principal Component Analysis to reduce feature dimensionality, improving efficiency and minimizing overfitting.
- Achieved 97.5% accuracy on a four-class CXR dataset with fast training and an average inference time of approximately 0.036 seconds per image.
- Incorporated LIME to provide interpretable visual explanations for model decisions.
- Designed a lightweight, modular architecture suitable for deployment in real-time and resource-constrained clinical environments.

2 Related Work

The utilization of deep learning for the diagnosis of chest disorders through CXR images has garnered significant attention in recent literature. Various studies have explored the application of CNNs in the context of medical image analysis, with a focus on differentiating between respiratory conditions. Noteworthy contributions in this field include Jiang et al. [1] proposed an enhanced VGG16 model for pneumonia disease Classification, where reduced layer depth with smaller convolutional kernels and achieved 89% accuracy. Heidari et al.[3] developed and tested the deep transfer learning CNN model for COVID-19, pneumonia, and normal CXR image classification. Used image pre-processing algorithm (histogram equalization and bilateral low-pass filter) before passing the image as input to the convolutional neural network. This CNN model with image pre-processing achieved an overall accuracy of 94.5 %. Sharma et al.[2] developed an integrated VGG16 with PCA model for image classification of four independent datasets— Fashion-MNIST dataset, ORL dataset of faces, corn leaf disease dataset, and rice leaf disease datasets and obtained an accuracy of 91.145%, 95.0%, 92.33 %, and 98.59%, respectively. Müller et al.[4] proposed a multi-disease detection pipeline in retinal imaging using an ensemble technique and transfer learning approach. In an experiment by Chen et al.[5], multiple CNN models were tested for COVID-19 detection utilizing a CXR and a transfer learning technique. Additionally, a grand CAM approach was used to visualize the infected region. Haritha et al.[6] showed COVID-19 detection through CXR in the VGG16 model using a transfer learning approach with the 2 classes dataset containing 1824 images and 5 classes dataset containing 2476 images. Qassim et al.[7] proposed the Residual Squeeze VGG16 model by modifying the fire module, which decreased the size of network, complexity, and training time with only a minor loss in accuracy for the Top-1 and Top-5 rankings. Setiawan and Damayanti [8] presented 35 layers of CNN architecture for pneumonia detection with an accuracy of f 96.3%. Hira et al.[9] presented a deep CNN-based model using transfer learning and fine-tuned approach to detect bacterial pneumonia, viral pneumonia, and normal CXR images. This has been experimented on nine CNN models (AlexNet, GoogleNet, ResNet-50, Se-ResNet-50, DenseNet121, Inception V4, Inception ResNet V2, ResNeXt-50, and Se-ResNeXt-50) and results show Se-ResNeXt-50 achieved the highest accuracy of 99.32% for binary classification and 97.55% accuracy with multiple classes. Reshi et al. [10] proposed a CNN model for COVID-19 detection using CXR images. experimented with different datasets, prepossessed the datasets, and achieved 99.5% accuracy which demonstrates the good capability of the proposed CNN model. Karacı [19] presented a modified VGG19 model for COVID-19 detection through CXR images. For binary class (COVID-19 and normal) classification model achieved an accuracy of 98.72% and for multi-class (COVID-19, pneumonia and normal) classification achieved 88.89%. Souid et al. [18] presented a paper for lung disease classification and prediction from CXR images using 'MobileNet v2' and achieved more than 90% of aacuracy. Sitaula and Hossain [11] proposed attention-based model using the attention module with VGG16 and compared with the state-of-the-art methods. Sanket et al. [12] presented a CNN based deep learning model for COVID-19 detection and achieved 98.4% of accuracy. Das et al. [13] proposed automatic COVID-19 detection using ensemble learning with CNN, achieved 91.62% of accuracy. Khan and Aslam [14] proposed deep learning based model and achieved an overall classification accuracy of 99.3%. Progga et al. [15] presented a paper based on transfer learning pre-trained VGG16 model with 98.75% of accuracy and compared with VGG-19 and Mobilnenet-V2. [21, 22]

These prior works laid the groundwork for leveraging deep learning techniques in the diagnosis of chest disorders, setting the stage for our proposed integrated model. In contrast to existing approaches, our model incorporates the VGG16 and VGG19 architectures, along with Principal Component Analysis (PCA) and feature fusion, offering a novel perspective on the classification of multiple diseases from CXR images.

3 Material and Methods

3.1 Datasets and Preprocessing

To conduct thorough experiments with the proposed approach, publicly available datasets for COVID-19, Pneumonia, Tuberculosis, and Normal CXR images were sourced from the Kaggle repository [17, 16]. The experimental dataset comprises a total of 1920 chest X-ray images, with 80 images from each category designated for validation and 400 images from each category used for training purposes.

After applying data augmentation techniques such as rotation, flipping, and scaling, the training set was expanded to 700 images per class, resulting in a more diverse and robust dataset for model learning.

3.2 Transfer Learning

Transfer learning, a fundamental aspect of machine learning, leverages acquired knowledge to tackle associated challenges. This study employed pre-trained weights from ImageNet, a dataset encompassing over 14 million images. Training deep learning models from the ground up is typically arduous due to the substantial computational demands, necessitating a robust GPU-equipped machine. The adoption of a transfer learning approach addresses this challenge while mitigating the risk of overfitting. In this research, the weights in VGG16, VGG19, and the proposed model were initialized through a transfer learning strategy.

3.3 Principal Component Analysis

In 1901, Karl Pearson developed the PCA, a statistical technique for dimensionality-reduction used to reduce the dimension (number of variables) of a large dataset with the help of orthogonal transformation. Either singular value decomposition or eigenvalues decomposition method can be used to implement PCA. In conclusion, the PCA concept is straightforward reducing the number of variables of the dataset, whereas preserving the maximum amount of information possible. The following Figure 1, depicts the PCA phase.

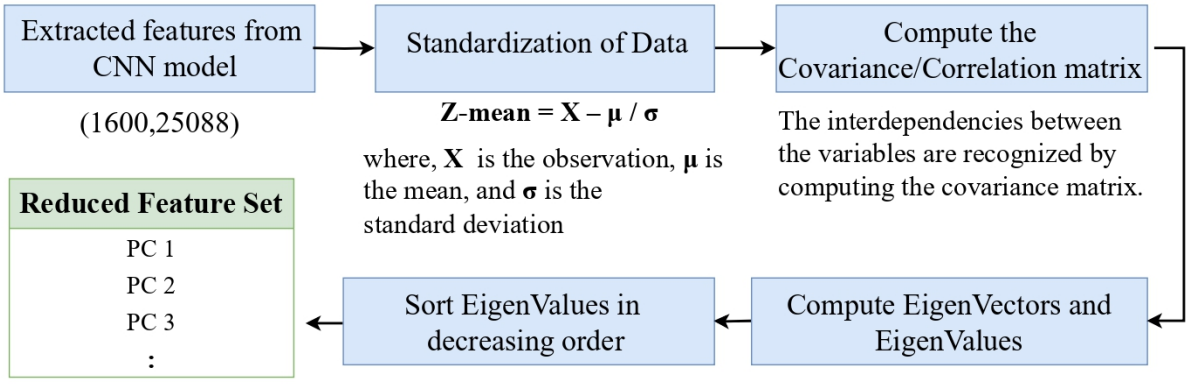


Figure 1: PCA calculation flow

Step 1: Standardize the Data

The mean and standard deviation for each feature must be determined for standardization. In this phase, the range of the continuous initial variables will be standardized to ensure that each variable contributes equally to the analysis.

$$\text{Standardization (Z-Mean)} = \frac{X - \mu(X)}{\sigma(X)} \tag{1}$$

Where X is the observation, $\mu(X)$ is the mean and $\sigma(X)$ is the standard deviation. Now, all variables are transferred to the same scale.

Step 2: Compute the Covariance/Correlation Matrix
The authors construct the covariance matrix (for all the feature dimensions) to determine the correlations between the variables. Computation for 2D data.

$$\text{Covariance Matrix } = \begin{bmatrix} \text{Cov}(x,x)\ \text{Cov}(x,y) \\ \text{Cov}(y,x)\ \text{Cov}(y,y) \end{bmatrix} \tag{2}$$

$$\text{Covariance}(m,n) = \frac{1}{N}\sum_{i=1}^{N}(m-\bar{m}) * (n-\bar{n}) \tag{3}$$

Where, m $\propto$ (1/n) if Cov (m, n) is negative (m and n are inversely correlated), while m $\propto$ n if Cov (m, n) is positive (m and n are correlated).

Remark 1. *Covariance holds the Commutative property, Cov(m, n)= Cov(n, m).*

Remark 2. *Cov(m) and Cov(n) are indicated as Cov(m, m) and Cov(n, n) respectively.*

Step 3: Compute the Eigenvectors and Eigenvalues The properties of matrix can be better understood by breaking it down into its eigenvalues and eigenvectors. The new feature space eigenvalues and eigenvectors are responsible for defining its magnitude and direction, respectively. First calculate eigenvalues by the equation 4:

$$\det(A - \lambda I) = 0 \tag{4}$$

Where 'I' is the identity matrix, A is the covariance matrix, and λ will be the eigenvalues.

Then calculate all the vectors v$\neq$0 by using eigenvalues, such that:

$$Av = \lambda v \tag{5}$$

Step 4: Sort Eigenvalues and their Corresponding Eigenvectors Now sort the eigenvalues in decreasing order $\lambda_1 > \lambda_2 > \lambda_3 > \cdots \lambda_k$ and top k eigenvalues represent the principal components (PCs) and these PCs represent the direction of data.

3.4 Convolutional Neural Network Models

3.4.1 VGG16

Simonyan et al. [20] presented the VGG16 convolutional neural network architecture in 2014. VGG16 model has 16 layers, including 13 Conv layers, 3 fully connected layers, max-pooling layers (having 2×2 filter with stride 2), and a softmax activation function followed by the last fully connected layer. VGG16 has a fixed size of the input is a 224×224 RGB image. It uses 3×3 filter convolutional layers with stride 1 and the same padding. This research work used the VGG16 base model for feature extraction and further at the classifier layer used three dense layers and in between the first two dense layers used dropout (0.4) and used 'relu' activation function moreover the last layers use 'softmax' function for multiclass classification as shown in Figure 2.

3.4.2 VGG19

The Visual Geometry Group (VGG19) model[20] is identical to VGG16 except that it allows 19 layers. Compared to VGG16, VGG19 contains three extra conv layers. There are 19 layers in the VGG19 algorithm: 16 convolutional layers, 3 fully connected layers, max-pooling levels that are used to minimise the volume size, softmax activation layers, and finally the last fully connected layer is used in this experiment. This network received a fixed size (224×224) RGB image as input, indicating that the matrix is shaped (224,224,3). The output of last pooling layer is flattened and fed into the dense layers with 'relu' activation function and the last layers use 'softmax' function to make the final decision as shown in Figure 3.

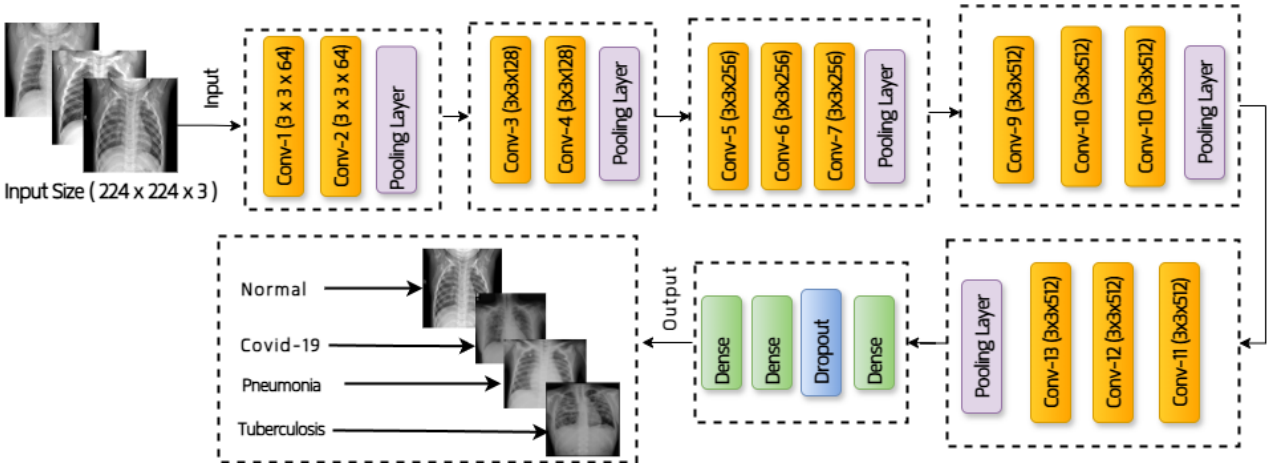


Figure 2: VGG16 Architecture

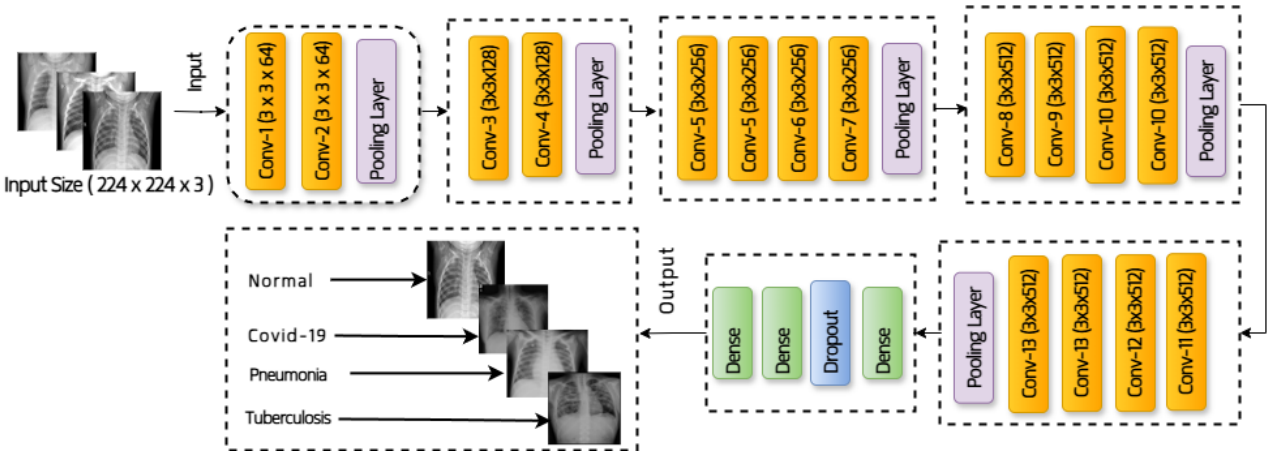


Figure 3: VGG19 Architecture

3.5 Evaluation Standards

The effectiveness of the neural-based model system is assessed using the following metrics: Accuracy, F1-score, Sensitivity or Recall, Precision, and Confusion matrix. A brief discussion of these matrices is given below.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FN} + \text{TN} + \text{FP})} \tag{6}$$

Specificity indicates the percentage of true negatives that the model accurately detects. A model with high specificity will accurately identify most of the negative outcomes, whereas one with low specificity may incorrectly classify many negative results as positive.

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})} \tag{7}$$

Precision indicates what percentage of positive identifications were actually correct.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{8}$$

Recall indicates the percentage of actual positives that were detected correctly.

$$\text{Sensitivity or Recall } = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{9}$$

F1 score is computed by calculating the harmonic mean of two matrices (Precision and recall). It is used to compare the performance of two classifiers.

$$\text{F1 score } = \frac{2 \times (\text{Precision} \times recall)}{(\text{Precision} + \text{recall})} \tag{10}$$

Here, TP stands for "True Positive," FP for "False Positive," TN for "True Negative," and FN for "False Negative."

4 Proposed Model

The proposed model is based on feature fusion from the well-known VGG16 and VGG19 models, and the additional dimensionality of the merged features is reduced by using PCA. Feature fusion is accomplished by selecting the maximum value for each pixel from the feature maps of both models. Additionally, feature fusion experimented with max, min, and average methods and max fusion outperformed min and average strategies. Attention-based fusion was skipped to keep the model lightweight for real-time applications. The architecture of the proposed model is shown in Figure 4.

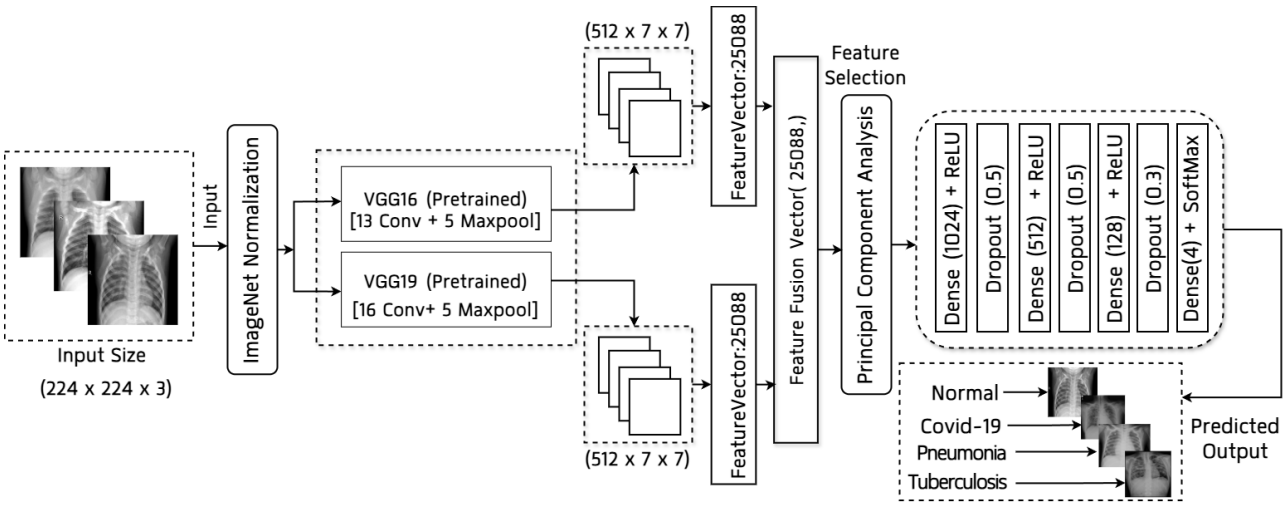


Figure 4: Proposed Architecture

In the proposed model, features are first, extracted with the dimension of $7 \times 7 \times 512$ from both the base models VGG16 & VGG19. The extracted features from both models are fused using the fusion method and were passed to PCA for feature selection. To obtain the most relevant features, it is necessary to examine the optimal number of components to be selected based on the explained variance value, which covers the important reduced features from the extracted features (high dimensional feature set). Finally, the resulting reduced relevant features were passed through the fully connected layers or classifier to classify the diseases. stepwise implementation process of the proposed model shown in algorithm 1.

5 Experimental Setup

All computational experiments were executed on RTX4060 GPUs with 12GB memory. The complete system architecture was implemented in Python using TensorFlow 2.6.0 and the Keras high-level API. Pretrained ImageNet weights were loaded for both VGG16 and VGG19 modules, with all layers frozen except for the top-level

Algorithm 1 Implementation Process

Step 1: First Load VGG16 and VGG19 models with 'Imagenet' trained weights and without classifier/fully connected layers. The input tensor size for VGG16 and VGG19 is 224×224 with three RGB channels. Pass the dataset (containing 1920 images for four classes) to both models.

Step 2: Pre-trained VGG16 and VGG19 models extract the features from the last Conv layer of $7 \times 7 \times 512$ features, then reshape or flatten it to 25088 features.

Step 3: Now combine the features of both models (picking the maximum value for each pixel) and apply PCA on the combined features to extract the components and ensure that the most information is retained.

Step 4: Reduced components are further, sent to the classifier layer for classification after training the classifier layer.

Step 5: When a fresh test image (X-ray) is provided to the proposed model. VGG16 and VGG19 first extract the features, then merge both features and apply PCA. Now reduces the components and subsequently passes to the classification layer to categorize the diseases.

classifier layers to enable task-specific adaptation. A fixed learning rate of 0.001 was used in conjunction with the Adam optimizer to minimize classification loss and accelerate convergence. The system was trained over 30 epochs, with each batch of input images resized to 224×224 pixels to comply with the input constraints of the CNN modules. To ensure reliable evaluation of system performance under constrained computational resources, the training process was deliberately confined to low-latency GPU environments with transfer learning to mitigate overfitting and reduce convergence time.

6 Experimental Results and Discussion

This work proposed a model for multi-disease detection. VGG16 and VGG19 base models are used for feature extraction which takes input size 224×224 images and it is pre-trained by the Imagenet dataset. Features are extracted as the output of both the models and further fused feature map passes to PCA which reduces the dimension and further, reduced components are passed to the classifier layer to perform classification. The stepwise experimentation results of the proposed methodology are discussed below:

6.1 Feature Selection

To determine the appropriate number of principal components for dimensionality reduction, the authors analysed the cumulative variance retained across different PCA configurations. As shown in Figure 5, 100 components capture approximately 65% of the total variance, 200 capture around 78%, while 300 and 500 components retain over 85% and 90% respectively. Although higher component sizes retained more variance, the model achieved the best classification accuracy with 200 components, balancing performance and efficiency. This analysis guided our selection of PCA dimensionality in the proposed pipeline. Table 1, illustrates the outcomes with the various PCA components.

The maximum accuracy is 97.50% in the proposed integrated model, which had 200 components. Accuracy is reduced as the components are increased more. Therefore, 1600 images with 25088 features make up the training dataset (1600×25088). After applying PCA on this, features were reduced to 200 (1600×200).

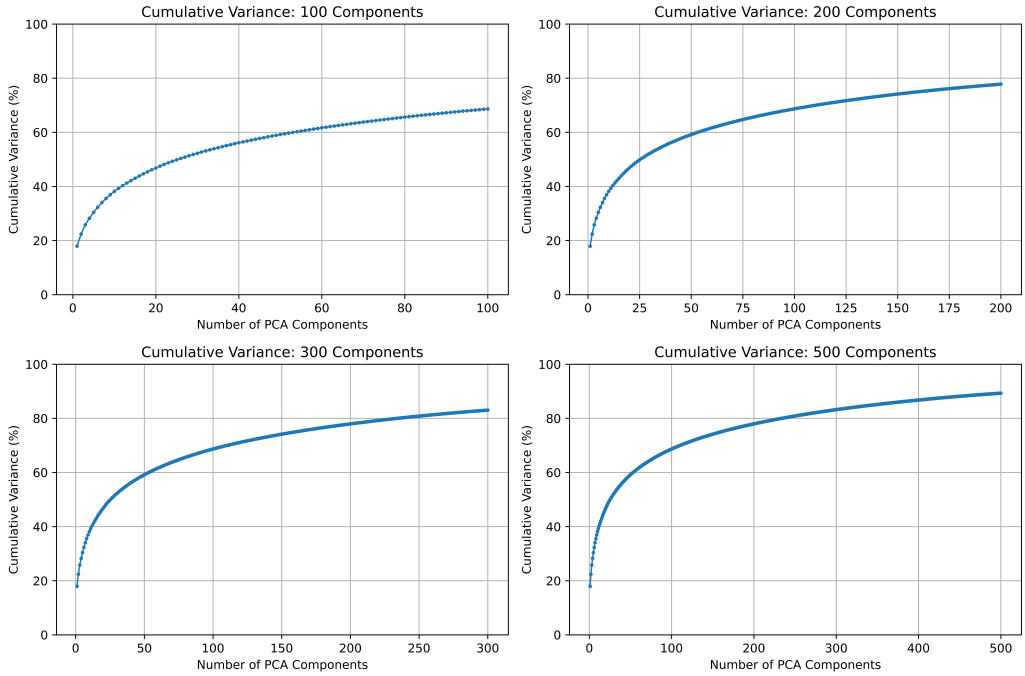


Figure 5: Cumulative variance retention plots for PCA with different component sizes (100, 200, 300, and 500).

Table 1: Different PCA components and analysis results

Model	Components	Training time (In seconds)	F1-Score	Precision	Recall	Accuracy
Proposed model	100	03.596728	94%	94%	94%	93.75%
	150	03.544028	94%	94%	94%	94.06%
	200	**04.062956**	**97%**	**98%**	**97%**	**97.50%**
	250	03.778688	96%	96%	96%	96.25%
	300	04.480812	95%	95%	95%	95.00%

6.2 Classification

This study tested deep neural network models VGG16, VGG16 with PCA, VGG19, and VGG19 with PCA. This study proposes an integrated CNN model on the ImageNet dataset, utilizing a transfer learning strategy to detect four classes: COVID-19, Normal, Pneumonia, and Tuberculosis. The experimental results and performance matrices are presented below.

6.2.1 Predicted Output

The proposed model uses new X-ray pictures as input to assess the performance of the model. The model predicts diseases for all four classes as demonstrated in below Figure 6.

To optimize the model, we performed a small-scale grid search over key hyperparameters such as learning rate, dropout, and optimizer type. Based on validation accuracy, the best combination was selected. Table 2 shows the combinations tested. The final model achieved 97.5% accuracy using a learning rate of 0.0001, dropout 0.5, and AdamW optimizer.

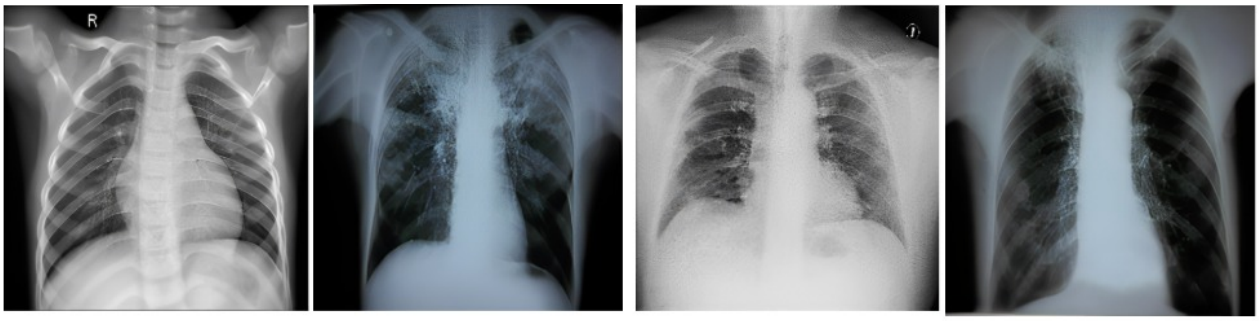


Figure 6: Predicted Output

Table 2: Grid Search Results on Validation Set

Learning Rate	Dropout	Optimizer	Accuracy (%)
0.001	0.3	Adam	95.2
0.001	0.5	AdamW	96.8
0.0001	0.3	Adam	96.9
0.0001	0.5	AdamW	**97.5**
0.0001	0.5	Adam	96.4
0.0001	0.5	SGD+Momentum	96.0

6.2.2 Training and Validation Accuracy

Figure 7 displays the training and validation accuracy graphs for both VGG16 and VGG19 models, considering variations with and without PCA, along with the proposed approach. The performance of the training accuracy is represented by the red line, and the validation accuracy is represented by the yellow line. The X-axis shows each epoch, and the Y-axis shows growing accuracy.

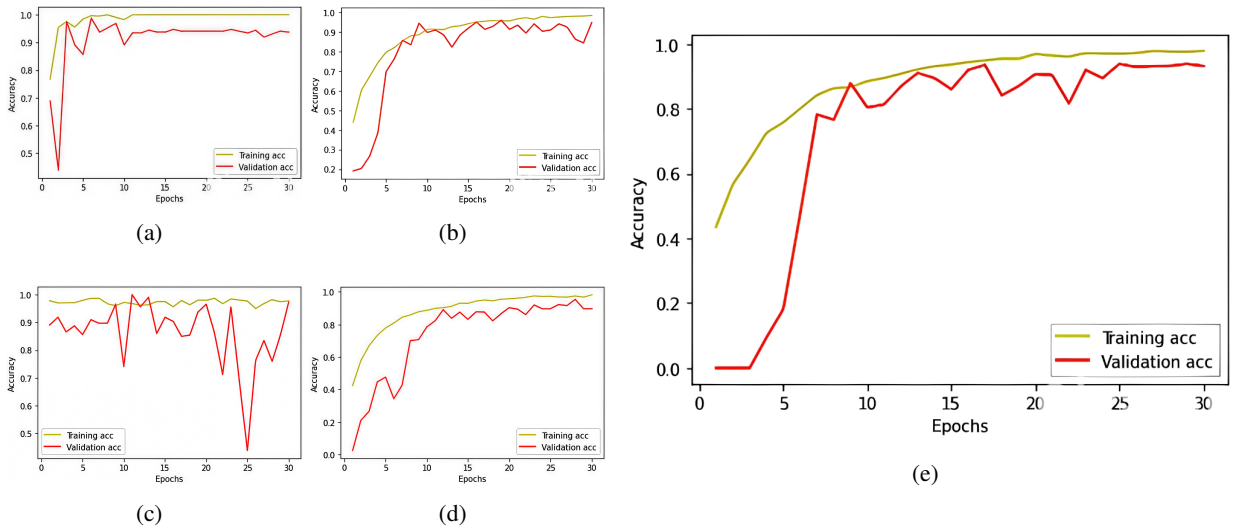


Figure 7: Training and validation accuracy: (a) VGG16 (b) VGG16 with PCA (c) VGG19 (d) VGG19 with PCA (e) Proposed model

In comparison to the various models, we utilised in our research, the proposed technique has the best ac-

curacy. Validation accuracy for the VGG19 base model decreases after 20 epochs but increases again after 25 epochs, whereas VGG19 training and validation accuracy with PCA improves with epochs.

6.2.3 Training and Validation Loss

Training loss and validation loss curves provide insight into the training procedure and how the network learns, as shown in Figure 8.

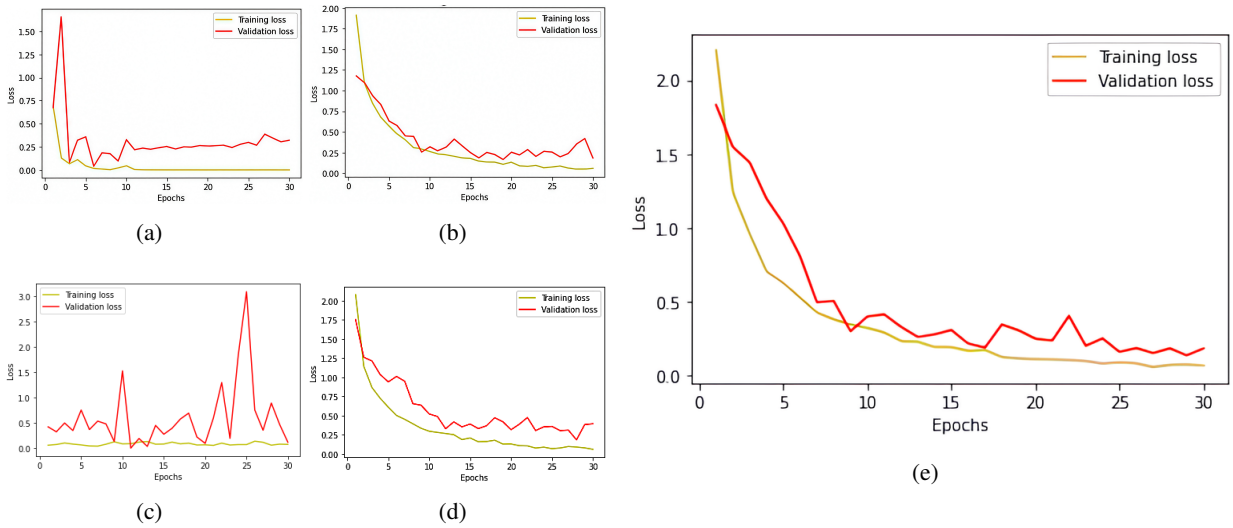


Figure 8: Training and validation loss: (a) VGG16 (b) VGG16 with PCA (c) VGG19 (d) VGG19 with PCA (e) Proposed model

Validation loss in the VGG16 and VGG19 models showed overfitting, however, both models using PCA showed improved validation loss. The training of proposed model and validation losses are almost identical, resulting in a superior validation loss graph.

6.2.4 Confusion Matrix

The effectiveness of a classification model or classifier is described by a confusion matrix. Essentially, it summarises and visualises the outcomes of a classification algorithm. 80 images were utilised in this experiment for each class. Non-diagonal numbers indicate incorrect predictions from each class, whereas diagonal values indicate the correct predictions as shown in Figure 9. Moreover, analysis of confusion matrix shown in Table 3.

Here, Class-0 represents COVID-19, Class-1 represents normal, Class-2 represents pneumonia, and Class-3 represents TB (tuberculosis).

Out of 320 images, the VGG16 basic model predicts 15 incorrect predictions. VGG16 with PCA makes 11 incorrect predictions. Similarly, the VGG19 basic model predicts 15 incorrectly, whereas VGG19 with PCA predicts 19 incorrect predictions. Furthermore, the proposed approach produces just 8 incorrect predictions.

6.3 Feature Space Visualization

To understand the separability of learned features after dimensionality reduction, both t-SNE and UMAP visualizations were applied on the CNN feature vectors post-PCA. Figure 10 illustrates the distribution of samples across the four classes. Distinct clusters in both 2D and 3D spaces confirm the discriminative capability of the extracted features, validating the effectiveness of feature learning pipeline of proposed model.

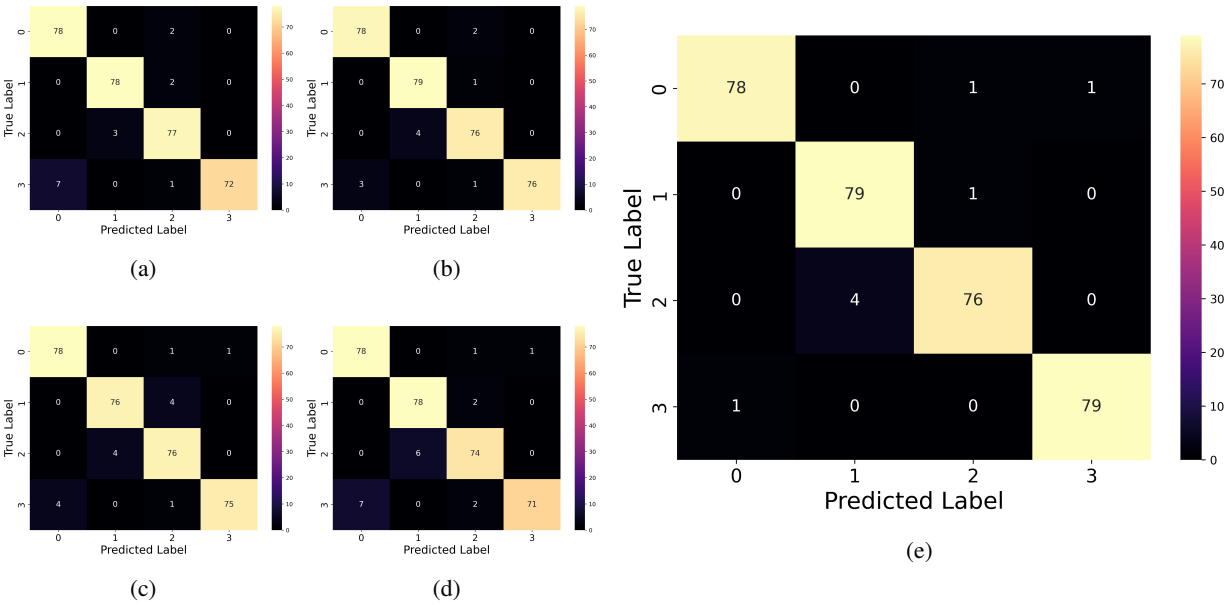
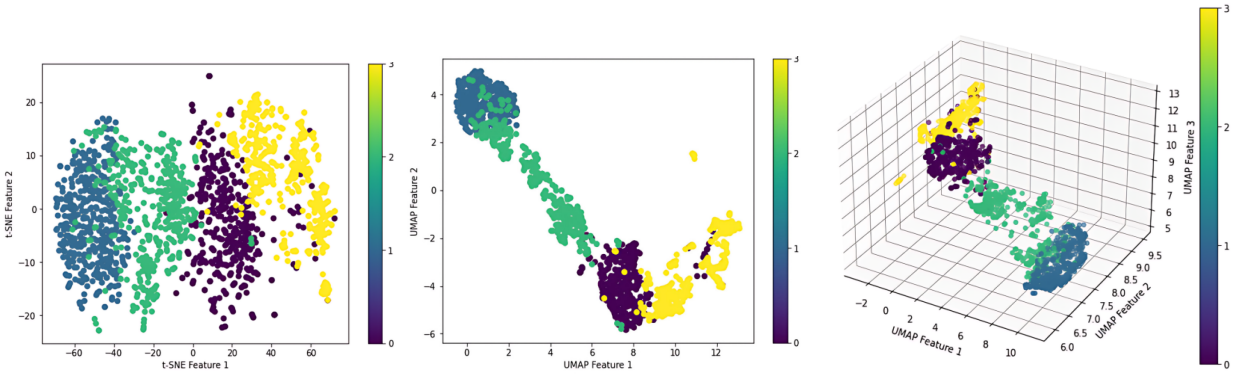


Figure 9: Confusion matrices: (a) VGG16 (b) VGG16 with PCA (c) VGG19 (d) VGG19 with PCA (e) Proposed model

Table 3: Prediction Table

	VGG16		VGG16-PCA		VGG19		VGG19-PCA		Proposed	
Disease	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
Class-0 COVID-19	78	2	78	2	78	2	78	2	**78**	**2**
Class-1 NORNAL	78	2	79	1	76	4	78	2	**79**	**1**
Class-2 PNEUMONIA	77	3	76	4	76	4	74	6	**76**	**4**
Class-3 TUBERCULOSIS	72	8	76	4	75	5	71	9	**79**	**1**

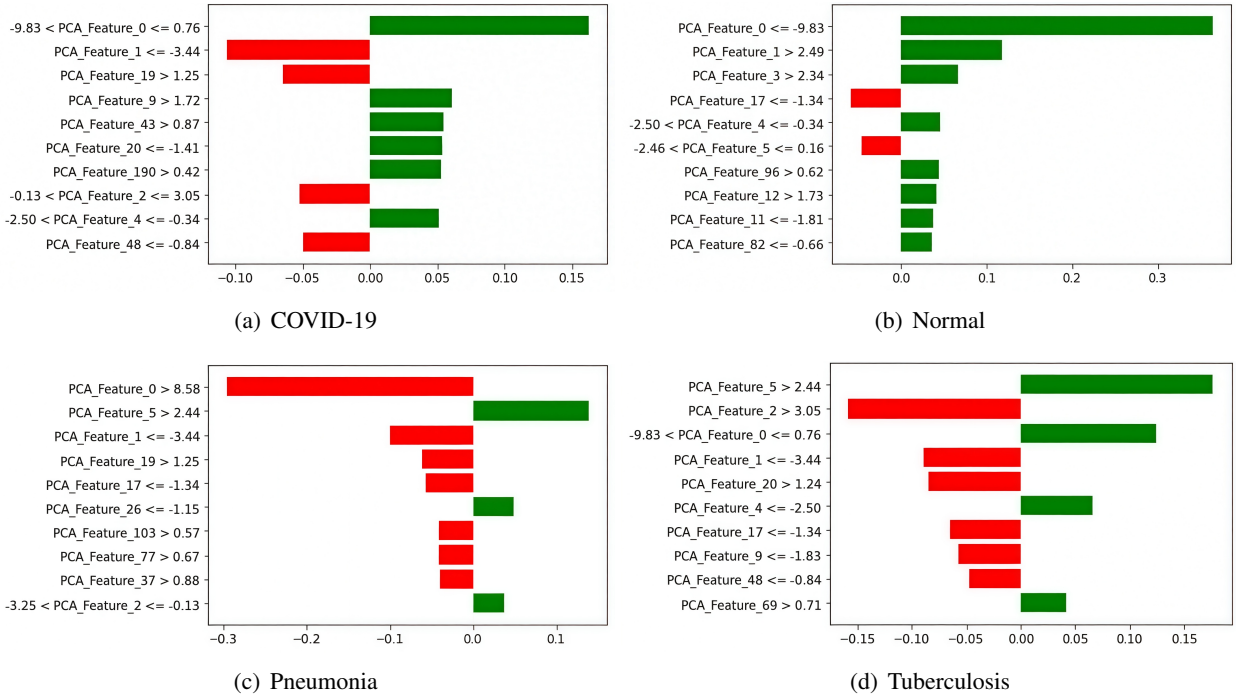


(a) t-SNE 2D (b) UMAP 2D (c) UMAP 3D

Figure 10: Visualization of CNN features after PCA using (a) t-SNE 2D, (b) UMAP 2D, and (c) UMAP 3D. Color denotes the class labels (COVID-19, Pneumonia, Normal, Tuberculosis).

6.4 Model Explainability Using LIME

To enhance transparency and interpretability, LIME (Local Interpretable Model-agnostic Explanations) was applied to understand which PCA-based features influenced the predictions of model. Figure 11 shows the LIME explanations for all four classes. Green bars represent features contributing positively towards the prediction, while red bars indicate negative influence. These visual insights help explain why the model predicted a particular disease and which features it relied on the most, supporting its potential use in clinical decision-making.



(a) COVID-19 (b) Normal

(c) Pneumonia (d) Tuberculosis

Figure 11: LIME explanations showing top PCA features influencing predictions for each class. Green bars indicate positive contributions, red bars indicate negative contributions.

6.5 System Scalability and Adaptability

6.5.1 Theoretical Complexity

Let n represent the dimensionality of the feature vector extracted from each CNN backbone (e.g., 25088 after flattening), and let k denote the number of principal components selected through PCA (e.g., 200 in the proposed model). The feature fusion step, performed using an element-wise maximum between the output vectors of VGG16 and VGG19, compares corresponding elements and selects the maximum. This operation has a linear computational complexity of $\mathcal{O}(n)$.

Following fusion, the PCA transformation projects the fused vector into a reduced subspace of size k. This involves matrix multiplication between the input vector and the k principal components (eigenvectors), resulting in a complexity of $\mathcal{O}(nk)$. Additionally, eigen decomposition during PCA contributes $\mathcal{O}(k^2)$. Therefore, the total complexity for the PCA step is $\mathcal{O}(nk + k^2)$.

In combination, the overall transformation from dual CNN feature extraction to PCA-based reduction remains efficient. This layered structure allows the system to handle high-dimensional data while maintaining low computational load, making it suitable for real-time medical diagnostics on resource-limited platforms.

Observation: Incorporating PCA drastically reduces training time and GPU load, enabling deployment in lightweight environments without sacrificing accuracy.

7 Comparison of the proposed model with different classifiers and CNNs

7.1 Comparison with Classifiers

After feature extraction from the combined VGG16 and VGG19 base model, applied PCA (unitized 200 components) with several classifiers, including SVM, Perceptron, Random Forest, K-Neighbors, and the proposed CNN-Dense layer were utilised to attain the maximum accuracy as shown in the Table 4.

Table 4: Comparison Table

Proposed model with different classifiers	Epochs	F1-Score	Precision	Recall	Accuracy
SVM	30	88%	89%	88%	87.50%
Perceptron	30	89%	91%	89%	89.37%
Random Forest	30	93%	93%	93%	92.81%
K-Neighbors	30	90%	90%	90%	90.31%
CNN-Dense layers	**30**	**97%**	**98%**	**97%**	**97.50%**

Among all classifiers, the proposed technique uses the CNN-Dense layer to produce the greatest accuracy.

7.2 Comparison with CNN Models

Try out all the models shown in Table 4, trained with the same dataset with 30 epochs at the learning rate of 0.001 using Adam optimizer, where the proposed model attained the highest accuracy (97.50%) in around 4 seconds of training. Table 5 shows the performance of different CNN models. sharma et al.[2] suggested an efficient model in which the features were extracted from VGG16 and reduced using PCA Following that, the extracted features were classified using a multilayer perceptron classifier, a support vector machine, and a random forest technique. Corn leaf disease and rice leaf disease training time 10.95 seconds and 14.23 seconds respectively. while our proposed model required just around 4 seconds of training.

Table 5: Performance comparison table

CNN Models	Epochs	Training time (In seconds)	F1-Score	Precision	Recall	Accuracy
VGG16	30	56.668942	95%	95%	95%	95.31%
VGg19	30	135.042444	96%	97%	96%	96.87%
VGG16- PCA	30	04.027283	95%	95%	95%	95.00%
VGG19-PCA	30	08.030868	94%	94%	94%	94.06%
Proposed model	**30**	**04.062956**	**97%**	**98%**	**97%**	**97.50%**

8 Conclusion

Deep learning is increasingly being used in several fields nowadays. CNNs widely used for object or image detection/classification. It is capable of detecting important features automatically without the need of human supervision. In addition, as compared to a standard neural network, it reduces computation. This research work used a deep learning approach to develop a model using PCA and a transfer learning approach. In the proposed approach, two well-known models, VGG16 and VGG19, were combined and 97.50% accuracy was achieved

in around four seconds of training time. However, VGG16, VGG19, VGG16 with PCA, and VGG19 with PCA achieved 95.31%, 96.87%, 95% and 94% respectively. Grid search hyperparameter tuning can enhance this model in further research.

This study presents a modular and efficient deep learning system for multi-disease classification from CXRs images. By integrating VGG16 and VGG19 architectures with Principal Component Analysis (PCA), the proposed model forms a composite information-processing pipeline capable of accurately detecting COVID-19, pneumonia, tuberculosis, and normal cases. The fusion of features followed by dimensionality reduction resulted in a 97.50% classification accuracy with only 4 seconds of training time, outperforming both standalone CNNs and traditional classifiers.

The performance of system remained robust under computational constraints, validating its scalability for real-time diagnostic use. The proposed approach highlights the benefit of using a layered design with attention to dimensional complexity and subsystem interaction.

Future improvements may explore hyperparameter tuning, advanced fusion strategies, and integration with explainability modules to enhance clinical trust and interpretability. This architecture sets the groundwork for scalable, intelligent diagnostic systems adaptable to other imaging domains.

References

[1] Z.-P. Jiang, Y.-Y. Liu, Z.-E. Shao, K.-W. Huang, "An Improved VGG16 Model for Pneumonia Image Classification", *Applied Sciences* 11(23):11185, 2021.

[2] R. Sharma, A. Singh, "An Integrated Approach towards Efficient Image Classification Using Deep CNN with Transfer Learning and PCA", *Advances in Technology Innovation* 2:105-117, 2022.

[3] M. Heidari, S. M. Harikandehei, A. Z. Khuzani, G. Danala, Y. Qiu, B. Zheng, "Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms", *International Journal of Medical Informatics* 144:104284, 2020.

[4] D. Müller, I. Soto-Rey, F. Kramer, "Multi-disease detection in retinal imaging based on ensembling heterogeneous deep learning models", *German Medical Data Sciences 2021: Digital Medicine: Recognize–Understand–Heal*, pages=23-31, IOS Press, 2021.

[5] A. Chen, J. Jaegerman, D. Matic, H. Inayatali, N. Charoenkitkarn, J. Chan, "Detecting COVID-19 in CXRs using transfer learning with VGG16", *CSBio'20: Proceedings of the Eleventh International Conference on Computational Systems-Biology and Bioinformatics*, pages=93-96, 2020.

[6] D. Haritha, C. Praneeth, M. K. Pranathi, "Covid prediction from x-ray images", *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pages=1-5, IEEE, 2020.

[7] H. Qassim, D. Feinzimer, A. Verma, "Residual squeeze VGG16", *arXiv preprint arXiv:1705.03004*, 2017.

[8] W. Setiawan, F. Damayanti, "Layers modification of convolutional neural network for pneumonia detection", *Journal of Physics: Conference Series*, volume=1477, number=5, pages=052055, IOP Publishing, 2020.

[9] S. Hira, A. Bai, S. Hira, "An automatic approach based on CNN architecture to detect COVID-19 disease from chest X-ray images", *Applied Intelligence* 51(5):2864-2889, 2021.

[10] A. A. Reshi, F. Rustam, A. Mehmood, A. Alhossan, Z. Alrabiah, A. Ahmad, H. Alsuwailem, G. S. Choi, "An efficient CNN model for COVID-19 disease detection based on X-ray image classification", *Complexity* 2021, Hindawi.

[11] C. Sitaula, M. B. Hossain, "Attention-based VGG-16 model for COVID-19 chest X-ray image classification", *Applied Intelligence* 51(5):2850-2863, 2021.

[12] S. Sanket, M. V. R. Sarobin, L. J. Anbarasi, J. Thakor, U. Singh, S. Narayanan, "Detection of novel coronavirus from chest X-rays using deep convolutional neural networks", *Multimedia Tools and Applications* 81(16):22263-22288, 2022.

[13] A. K. Das, S. Ghosh, S. Thunder, R. Dutta, S. Agarwal, A. Chakrabarti, "Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network", *Pattern Analysis and Applications* 24(3):1111-1124, 2021.

[14] I. U. Khan, N. Aslam, "A deep-learning-based framework for automated diagnosis of COVID-19 using X-ray images", *Information* 11(9):419, 2020.

[15] N. I. Progga, M. S. Hossain, K. Andersson, "A deep transfer learning approach to diagnose COVID-19 using X-ray images", *IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, pages=177-182, IEEE, 2020.

[16] S. Jaeger, S. Candemir, S. Antani, Y. X. Wáng, P. X. Lu, G. Thoma, "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases", *Quantitative Imaging in Medicine and Surgery*, December 2014. DOI: 10.3978/j.issn.2223-4292.2014.11.20., PMID: 25525580. NIH publication. URL: https://www.kaggle.com/datasets/raddar/tuberculosis-chest-xrays-shenzhen?select=images

[17] P. Viradiya, "Chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images", *Kaggle Repository*, 2021. URL: https://www.kaggle.com/datasets/preetviradiya/covid19-radiography-dataset. Accessed on: June 20, 2022.

[18] A. Souid, N. Sakli, H. Sakli, "Classification and predictions of lung diseases from chest X-rays using MobileNet V2", *Applied Sciences* 11(6):2751, 2021. MDPI.

[19] A. Karacı, "VGGCOV19-NET: automatic detection of COVID-19 cases from X-ray images using modified VGG19 CNN architecture and YOLO algorithm", *Neural Computing and Applications* 34(10):8253-8274, 2022. Springer.

[20] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[21] R. Kumar, C.-T. Pan, Y.-M. Lin, S. Yow-Ling, T.-S. Chung, and U. G. S. Janesha, "Enhanced multimodel deep learning for rapid and precise diagnosis of pulmonary diseases using chest X-ray imaging", *Diagnostics*, vol. 15, no. 3, p. 248, 2025.

[22] C. C. Ukwuoma, D. Cai, M. B. B. Heyat, O. Bamisile, H. Adun, Z. Al-Huda, and M. A. Al-Antari, "Deep learning framework for rapid and accurate respiratory COVID-19 prediction using chest X-ray images", *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 7, p. 101596, 2023.