Library Progress International *Print version ISSN 0970 1052*
Vol.44, No.4, Jul-Dec 2024: P. 870-889 *Online version ISSN 2320 317X*

Original Article *Available online at www.bpasjournals.com*

Detection of COVID-19 by classifying CT-Scan images using Enhanced MobilenetV2

Dr. M. Anusha[1], P. Kiruthika*[2]

[1]Assistant Professor, PG & Research Department of Computer Science, National College (Autonomous), Affiliated to Bharathidasan University, Tamil Nadu, India. Email: E-mail-anusha260505@gmail.com

[*2]Research Scholar, PG & Research Department of Computer Science, National College (Autonomous), Affiliated to Bharathidasan University, Tamil Nadu, India. Email: Kiruthikashyne@gmail.com

*Corresponding Author: E-mail-Kiruthikashyne@gmail.com

How to cite this article: Dr. M. Anusha , P. Kiruthika (2024) Detection of COVID-19 by classifying CT-Scan images using Enhanced MobilenetV2. *Library Progress International,* 44(4), 870-889

Abstract

Using images from chest CT scans, this study attempts to determine how well transfer learning performs in correctly identifying COVID-19. This research enhanced the MobileNetV2 deep learning model, decision trees, support vector machines, and logistic regression for feature extraction and classification, respectively. 13545 CT scan images of the chest were included in the dataset for this study. A smaller dataset of images from pneumonia, COVID-19, normal, Omicron and delta chest CT scans is used to fine-tune the models after they have been trained on a larger image dataset. When machine learning models based on transfer learning are applied to chest CT scan images, the study's findings demonstrate that COVID-19 can be successfully identified. The MobileNetV2 model turned out to be the top performer. This model performed well on the testing set, achieving a 97.5% accuracy rate along with good precision, recall, and F1-score across all classes. Overall, the model performed well, as evidenced by the high average precision and recall. As a result, the study shows how well deep learning works for both image classification and feature space extraction from chest CT scan images used for COVID-19 detection.

Keywords: Covid 19, Pneumonia, Normal, Omicron and Delta, CT scans images, Deep feature extraction, MobileNetV2, Transfer learning.

1. Introduction

A vital tool for both disease diagnosis and treatment is medical imaging. However, acquiring medical images for different diseases can be challenging due to the need for specialized equipment and ethical considerations (Pahar et al., 2022). Artificial intelligence (AI) tools such as Generative Adversarial Networks have demonstrated remarkable promise in producing synthetic medical images for range of uses, such as deep learning model training and evaluation. However, the effectiveness of GAN-generated images largely depends on the quality of the features extracted from them (Loey et al., 2020).

A critical phase in the creation of machine learning models is feature engineering. It entails picking and removing pertinent features from the unprocessed data to increase the precision and effectiveness of subsequent operations, like diagnosis or classification. Traditional feature engineering methods involve manually selecting and extracting features based on domain knowledge or statistical analysis. However, these techniques can be labour- and time-intensive and may not capture all the relevant information in the data (Singh et al., 2021). In this work, we develop a feature engineering a deep learning-based method to extract meaningful features from multiclass generated CT scans and assess how well it performs in subsequent tasks like diagnosis or classification. (Li et al., 2021). The proposed technique involves passing the generated CT scans through a pre-trained deep learning model to extract high-level features that can be used for training a new model (Shaik & Cherukuri, 2022).

In some years, use of GANs for medical image generation has grown in popularity. For a variety of purposes, such as data augmentation, data privacy, and the training and assessment of deep learning models, GANs are used to create synthetic medical images. However, the complexity of the underlying distribution and the GAN model's architecture can have a substantial impact on how well GAN-generated images turn out. (Umer et al., 2022). The proposed deep learning-based feature engineering technique can overcome some of the limitations of traditional feature engineering methods and improve the quality of features extracted from GAN-generated images (Mahmood Khan et al., 2022). The high-level features of the generated CT scans can be captured by the pre-trained deep learning model employed in the suggested method, which can be used to train a new model for task that come later. (Turkoglu, 2021). The performance of the suggested method will be compared with conventional feature engineering techniques to assess its effectiveness. The evaluation will be carried out on a dataset of multiclass generated CT scans, including Normal, Covid-19, Pneumonia, Omicron, and Delta. The performance of the proposed technique will be measured based on the accuracy and efficiency of downstream tasks such as classification or diagnosis (Shamsi et al., 2021).

Therefore, the proposed deep learning-based feature engineering technique which have the potential to improve the quality of features extracted from GAN-generated medical images and enhance the accuracy and efficiency of downstream tasks such as classification or diagnosis. The proposed technique can overcome some of the limitations of traditional feature engineering methods and reduce the training time and computational resources required for developing the new model.

1.1 Aim and Objectives

The objective is to create a feature engineering method based on deep learning to extract important features from multiclass generated CT scans and evaluate its effectiveness in improving downstream tasks such as classification or diagnosis.

The research goals are as follows:

- To assess how well a GAN-based method produces CT scan images of Normal, Covid-19, Pneumonia, Omicron and Delta classes.
- To find out if applying pre-processing methods like Contrast Limited Adaptive Histogram Equalisation (CLAHE) which can improve the quality of CT scan images that are produced.
- To explore the effectiveness of feature extraction using a enhanced MobileNetV2 model in extracting vectors from the generated CT-scan images.
- To evaluate the quality of CT-scan images produced for further tasks like diagnosis or classification by extracting features using MobileNetV2.
- The enhanced MobileNetV2 has been implemented to perform the categorization of COVID-19 CT scan images.

1.2 Problem Statement

When it comes to the diagnosis and treatment of many diseases, medical imaging is essential. However, acquiring medical images for different diseases can be challenging due to ethical considerations and the need for specialized equipment. Deep learning model training and evaluation are just two of the many uses for synthetic medical images that Generative Adversarial Networks (GANs) have shown promise for producing. However, the qualities of GAN-generated images depend on the effectiveness of feature extraction methods.

Henceforth, the problem statement aims to create an advanced learning-based feature engineering technique to extract important features from multiclass generated CT scans and evaluate its effectiveness in improving downstream tasks such as classification or diagnosis. The suggested method should be able to get around the drawbacks of conventional feature engineering techniques and cut down on the amount of time and computing power needed to train the new model. Creating a feature engineering method that can successfully extract significant features from GAN-generated medical images is the primary challenge in this problem. In addition to reducing potential noise and artefacts in the generated images, the suggested technique should be able to extract all the pertinent information from the data. Moreover, the technique should be able to reduce the training time and computational resources required for developing the new model.

The suggested method's efficacy should be assessed by contrasting its output on a dataset of multiclass generated CT scans with that of conventional feature engineering techniques. The findings of the study may have big impact on how effectively and accurately medical image analysis and diagnosis are performed. In the end, the suggested feature engineering method may result in improved patient outcomes and more effective healthcare administration.

2. Related Work

The literature review discusses recent research on transfer learning-based feature engineering techniques for GAN-generated medical images. It highlights the effectiveness of these techniques in improving downstream tasks such as classification or diagnosis and identifies research gaps that require further research to be done. Recent research has shown significant interest in developing effective feature engineering techniques for medical image analysis using deep learning. The goal of these projects is to raise the standard of GAN-generated medical images and improve tasks like diagnosis and classification.

A recent study by Ismael et al. (2021) suggested a method for COVID 19 detection using chest X-ray images. The deep features of the image have been extracted in the current work using pretrained CNN models, including, ResNet50 ResNet18 and VGG19. Additionally, Support Vector Machine (SVM) has been used to classify the extracted features.

Additionally, Sachdev et al. (2021) used a pretrained DenseNet model to propose a feature engineering technique for CT scans generated by GANs. The efficacy of their suggested method in enhancing classification task accuracy—such as differentiating between non-COVID-19 and COVID-19 CT scans—was assessed in the study. The findings demonstrated that, in comparison to conventional feature engineering techniques, the suggested technique greatly increased the accuracy of classification tasks.

Like in the study Teodoro et al (2023) a feature engineering technique for GAN-generated skin lesion images using a pre-trained EfficientNet model was proposed. The study assessed how well their suggested method worked to increase the precision of tasks involving the diagnosis of skin lesions. The findings demonstrated that the suggested method greatly increased the accuracy of skin lesion diagnosis while outperforming conventional feature engineering techniques.

Likewise, Zhao et al (2021) suggested a deep learning model for classifying CT scan data to detect COVID 19 illness. A state-of-the-art method that successfully detects COVID-19 infection from CT scans is deep learning for COVID-19 detection based on images. This method uses artificial intelligence. With more accurate early diagnosis and treatment, this technology might be able to halt the disease's progression and save lives. A deep learning-based classification system called CoroDet was proposed by Hussain et al. (2021) for COVID-19 detection from chest X-ray images. This innovative method has the potential to increase diagnostic efficacy and aid in the pandemic response by reliably identifying COVID-19 cases from X-ray scans.

Therefore, the discussed research has shown that deep learning-based feature engineering techniques can effectively extract important features from GAN-generated medical images and significantly improve downstream tasks such as classification or diagnosis. These methods can enhance patient outcomes in healthcare by cutting down on the time and computational resources needed to create accurate models. However, further research is needed to explore the effectiveness of these techniques for different medical imaging modalities and diseases.

2.1 Research Gap

Although recent research has shown promising results in developing effective transfer learning-based feature engineering techniques for GAN-generated medical images, there is still a research gap in the evaluation of the generalizability and robustness of these techniques across different datasets and imaging modalities (Sachdev et al., 2021). Most of the current research focuses on specific diseases or imaging modalities, such as COVID-19 diagnosis using CT scans or Alzheimer's disease diagnosis using MRI scans. Moreover, there is a lack of research on the comparison of different pre-trained models and feature extraction techniques for GAN-generated medical images (Teodoro et al., 2023).

The interpretability and explainability of the features that are extracted from GAN-generated medical images also require more investigation. While deep learning-based feature engineering techniques can significantly improve the accuracy of downstream tasks, such as classification or diagnosis, it is essential to understand the underlying features that contribute to the model's predictions and their clinical significance. Consequently, the main goal of future research should be to assess how well deep learning-based feature engineering techniques generalise and hold up across various datasets and imaging modalities. Furthermore, more investigation is required to compare the efficacy of various feature extraction methods and pre-trained models for GAN-generated medical images. Lastly, the planned study will also include a comparison with multiple lung diseases to enhance performance.

3. Methodology

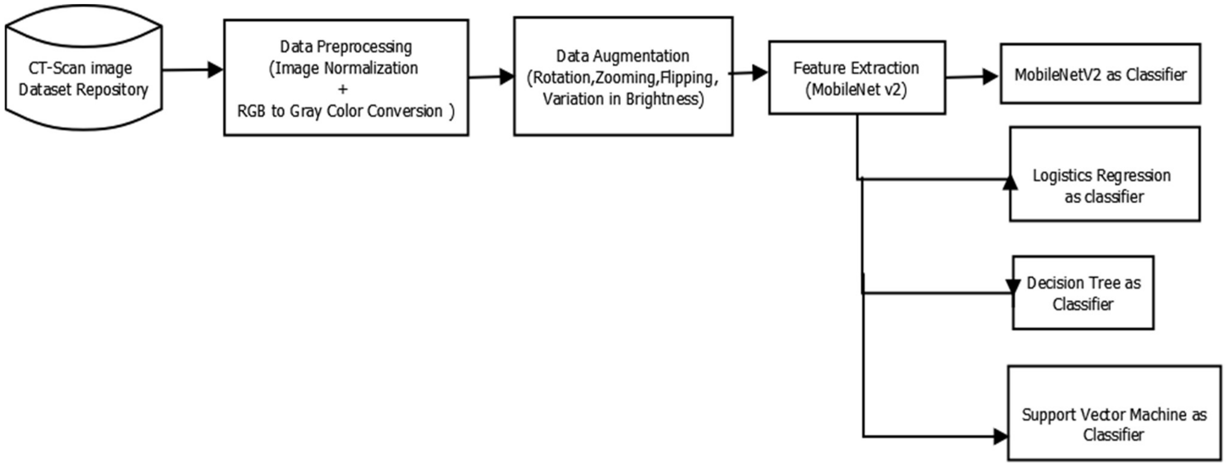


Figure 1: System Overview

The methodology section explains the procedures and techniques used to conduct the study and the functionality of each stage in the proposed methodology is demonstrated in the above figure. It offers a thorough explanation of the procedures for gathering data, preprocessing, and analysing it to meet the study's goals. The section also justifies the selection of specific methods and explains how they are suited to address the research questions.

3.1 Dataset

For this study, data was collected from the generated CT scan images using the model that was built using deep learning techniques (Contribution 1 can be referred here). The data includes various online sources and was categorized into four classes based on the specific condition or disease represented by each class: Normal, Covid-19, Pneumonia, and Omicron/Delta. The quantity of pictures in every class indicates the amount of data available for training and evaluating machine learning models. The dataset comprises a total of 61,782 normal CT scan images, 21,036 Covid-19 CT scan images, 21,191 pneumonia CT scan images, and 12,200 CT scan images for the Omicron and Delta variants of the SARS-CoV-2 virus. Figure 1 depicts the image of a normal CT scan.

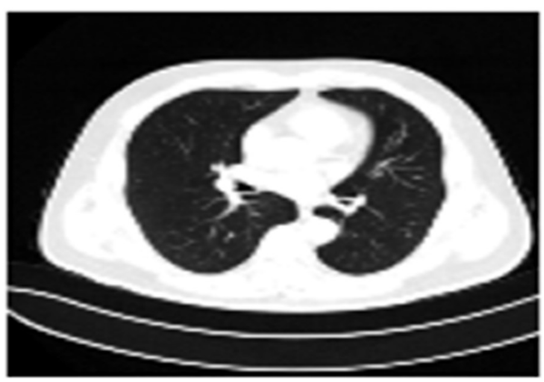


Figure 2: Image of normal CT scan

Further, to use a pre-trained network, the input data must be of the same dimensions as the input to the pre-trained network. This is because the pre-trained network has already learned a set of filters and weights that are optimized for the input dimensions it was trained on. Therefore, if the input dimensions are different, the pre-trained network may not be able to take useful characteristics out of the input data. In this model, 224,224, 3 as the input dimension of the model will be utilized.

3.2 Image Normalization

Normalization is the process of scaling and transforming the pixel values of images to a consistent range. The mechanism on how this process happens is given below - Scaling the image pixel values to a range, like [0, 1] or [-1, 1], is applied. To accomplish this, either divide the pixel values for the [0, 1] range by the maximum value that can be obtained (255, for example, in the case of 8-bit images), or for the [-1, 1] range, Calculate the standard deviation by dividing the mean by the amount. Scaling the pixel values helps to ensure that they fall within a consistent range and prevents them from being too large or too small, which can impact the performance of deep learning algorithms.

3.3 Data Pre-processing

CLAHE has several parameters that can be tuned for optimal performance. One of the most important parameters is the clip limit, which sets the maximum value for the histogram pixel count. A higher clip limit can lead to better contrast enhancement but may also result in over-amplification of some areas of the image. We have also white balance helps adjusts the color temperature of the image. White balance can be used to correct color distortions in the image, which can occur due to variations in the lighting conditions during image acquisition. Table 1 shows the important object parameters that have been used to train the model.

Table 1: Parameter used to train the model

Object	Parameter
Clip limit	0.10
White balance percentage	0.05
Resize	224,224
Interpolation	Inter cubic
CLAHE - Grid size	16,16

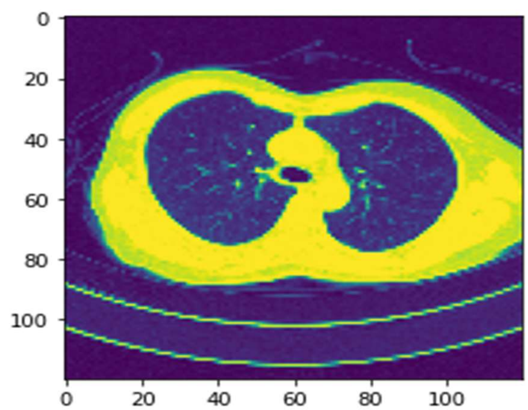


Figure:3 RGB to Gray scale conversion

The CT scans have been converted to gray scale from RGB color. The reasons to change the images into gray scale are, reduced computational complexity, simpler data preprocessing and reduced noise and complexity

- Reduced Computational Complexity: Compared to colour images with three channels (RGB), grayscale images have one channel, which lowers the computational complexity of the model. This results in faster training and inference times and requires less computational resources.

- Simpler Data Preprocessing: Gray scale images are simpler to preprocess compared to color images, as they do not require normalization or standardization of color channels. This simplifies the preprocessing pipeline and reduces the chance of errors or data corruption during preprocessing.

- Reduced Noise and Complexity: Gray scale images have reduced noise and complexity compared to color images, as they do not contain color variations or artifacts that can affect the performance of the model. Figure 3 shows the image after preprocessing.

Figure:4 Normal Gan image After Enhancement

3.4 Data Augmentation

An essential method for artificially growing a dataset in deep learning is data augmentation. When it comes to CT scans, it might make the model more resilient to adjustments made to the images. The model mentioned above made use of the following techniques for augmentation of data:

To augment the dataset and increase its diversity, several methods of data augmentation were used on the CT scans. The CT scans were randomly rotated by up to 30 degrees to simulate different orientations. They were -also randomly zoomed in or out by up to 20% to simulate different scales. To further increase diversity, the CT scans were randomly flipped horizontally to simulate mirror images and flipped vertically to simulate upside-down images. Additionally, the brightness of the CT scans was randomly adjusted to simulate different lighting conditions. By applying these data augmentation techniques, the proposed model can better generalize to new, unseen CT scans and improve its overall performance.

4. Modelling

The modelling section describes the implementation of transfer learning technique to achieve the study objectives. We will give a thorough description of the model architecture and training procedure utilised in this section.

Convolutional neural networks like MobileNetV2 are made to be effective and lightweight so they can be used on mobile devices with constrained processing power. It was created by Google as a component of their endeavours to build real-time neural networks for mobile devices. Depthwise separable convolutions are used by MobileNetV2 to minimise the number of parameters in the network without sacrificing accuracy. Each convolutional layer is split into two separate layers using this method: a pointwise convolution that uses 1x1 convolutions to combine the outputs of the depthwise convolution, which applies one filter to each input channel.

Feature extraction in the context of input image extraction can be accomplished by using MobileNetV2 as a pre-trained network. The pre-trained MobileNetV2 network can be used to extract features from new images because it has already learned a set of filters that are perfect for image classification tasks. The input images must first be preprocessed to match the dimensions of the input to the MobileNetV2 network to use MobileNetV2 for feature extraction. After that, features are extracted from the preprocessed images using the MobileNetV2 network, which has already been trained. For the target task, like segmentation or classification, a new neural network can be trained using the extracted features as inputs. The new network's overall performance can be enhanced, and the quantity of training data needed can be decreased by employing MobileNetV2 for feature extraction.

4.1 Modules – MobileNetV2

MobileNetV2 is composed of several modules that perform different operations on the input data. Here are brief explanations of each module and its functioning:

- **Input:** This module takes the input image and applies any necessary preprocessing steps to prepare it for processing by the network.

- **Convolutional layers:** MobileNetV2 uses several convolutional layers and filters them to extract features from the input image. To lower the network's parameter count, these layers employ depthwise separable convolutions.

- **Depthwise Convolution:** In a standard convolutional layer, the input tensor is convolved with a set of filters (kernels). Each filter is applied to the full depth of the input tensor. Depthwise convolution, on the other hand, applies a separate filter for each input channel. This is achieved by performing a convolution independently for each channel and then combining the results.

$$\mathbf{DW}\boldsymbol{Conv}(\mathbf{X})_{i,j,k} = \sum_{m,n} \mathbf{X}_{i+m,j+n,k} * \mathbf{K}_{m,n,k} \qquad (1)$$

where $X_{i,j,k}$ represents the value at spatial location *(i,j)* in the input tensor for channel *k*

$K_{m,n,k}$:This is the depthwise convolution kernel at spatial location *(m,n)* for channel *k*.

$DWConv(X)_{i,j,k}$ This is the result of applying the depthwise convolution at spatial location *(i,j)* for channel *k*.

∗: Denotes the convolution operation.

The summation is over all spatial locations *(m,n)* of the kernel.

The depthwise convolution is applied independently for each channel of the input tensor. So, if the input tensor has ‘C’ channels, there will be ‘C’ different depthwise convolution filters.

Pointwise Convolution:

The pointwise convolution is a standard 1x1 convolution applied to each location in the spatial domain. The equation for the pointwise convolution operation is as follows:

$$\mathbf{PWConv}(\mathbf{X})_{i,j,k} = \sum_{l} \mathbf{DWConv}(\mathbf{X})_{i,j,l} * \mathbf{K}_{l,k} \qquad (2)$$

where $DWConv(X)_{i,j,l}$: This represents the value at spatial location (i,j) for channel l in the output of the depthwise convolution.

$K_{l,k}$: This is the pointwise convolution kernel connecting channel l of the depthwise convolution output to channel k in the final output.

$PWConv(X)_{i,j,k}$ *:* This is the result of applying the pointwise convolution at spatial location (i,j) for channel k

.∗: Denotes the convolution operation.

The summation is over all channels l in the depthwise convolution output.

Bottleneck layers: Bottleneck layers are used to reduce the computational cost of the network. They consist of 1x1 convolutions that are used to compress the input feature maps before passing them through the more computationally expensive depthwise separable convolutions.

$$\mathbf{F_{out}=PWConv(DWConv(PWConv(X)))+X} \qquad (3)$$

where X is the input feature map,

DWConv is the depth wise separable convolution, and

PWConv is the pointwise convolution.

$F_{out}(X)$ is the final output feature map after passing through the bottleneck layer.

Blocks with inverted residuals: These blocks are used to deepen the network while minimising the number of parameters. They consist of a bottleneck layer followed by a 1x1 convolutional layer that expands the depth of the feature maps, and another bottleneck layer that compresses the feature maps again. The block is defined as follows:

$$\mathbf{F_{out}=PWConv(DWConv(PWConv(X)))+X} \qquad (4)$$

$$\mathbf{F_{out}=ReLU6(F_{out}+Shortcut(X))} \qquad (5)$$

where ReLU6 is the Rectified Linear Unit with clipping at 6

Shortcut is the shortcut connection.

$F_{out}(X)$ is the final output feature map after passing through the inverted residual layer.

Global Average Pooling (GAP): MobileNetV2 typically uses global average pooling to reduce spatial dimensions before the final fully connected layer. The equation for global average pooling is:

$$Fgap(x) = \frac{1}{HxW} \sum_{i=1}^{H} \sum_{j=1}^{H} W x_{ijk} \qquad (6)$$

Here, the terms are defined as follows:

where H and W stand for the input feature map's height and width, respectively.
Fgap: The output value for channel ‘x’ after global average pooling.
xijk : The value at the (i,j,k)-th position in the input feature map.

Final convolutional layer: The final convolutional layer takes the output of the inverted residual blocks and applies a set of 1x1 convolutions to produce the final feature maps.

4.2 Model Pseudocode

In this section, we present the pseudocode of our model implementation. The pseudocode provides a high-level overview of the implementation details of our approach, encompassing the essential phases of inference and model training.

```
Algorithm 1: Building the CT Image Classification Model
// Initialize Input Shape
Input Shape: Image size with 3 channels (RGB);
// Initialize MobileNetV2 as Base Model
Base Model: MobileNetV2 with pre-trained ImageNet weights;
// Configure Base Model
   1. Set include_top to False ;          // Exclude classification layer
   2. Set base model as non-trainable ;   // Freeze pre-trained weights
// Define Input Layer
Input Layer: Specify input shape;
// Apply Data Augmentation
Apply data augmentation using data_augmentation function;
// Build Model Architecture
   1. Input: Image data X with shape H × W × C ;          // Input
   2. Apply convolutional layer (256 filters, 3x3 kernel, ReLU activation) ;
      // Initial Convolutional Layer
   3. for i = 1 to N do
      Apply depthwise separable convolution block i;
      end
      ;          // Stacked Depthwise Separable Convolution Blocks
   Apply max pooling layer (2x2 pooling) // Max Pooling Layer
   4. Flatten the output // Flatten Layer
   5. for j = 1 to M do
      Apply fully connected layer with L_j units and ReLU activation;
      if j is the first dense layer then
         Apply dropout layer with dropout rate 0.5;
      end
      if j is the second dense layer before output then
         Apply dropout layer with dropout rate 0.5;
      end
      end
      ;          // Fully Connected Layers
   6. Apply output layer with softmax activation for 4 classes ;   // Output
      Layer
// Create the Model
Model: Use Model class with defined inputs and outputs;
// Compile Model
Compile the CNN model with appropriate loss and optimizer;
// Training
Train the model on annotated CT scan image with backpropagation
  and gradient descent;
// Testing
Test the model on new, unseen CT Scan Image;
// Output
Output: Class of CT Scan Images;
```

4.3. Model Tuning

In the following sub sections, the preprocessing techniques which have been used in this proposed methodology has discussed briefly

1. Adaptive learning rate:

Adaptive learning rate techniques are essential for training deep neural networks effectively. They adjust the learning rate during training to speed up convergence and prevent overshooting the minimum of the loss function. When using MobileNetV2, especially as both feature extractor and classifier for a specific task, it's common to use an adaptive learning rate algorithm like Adam optimizer. A smaller learning rate is often used during fine-tuning to ensure that the model does not deviate too far from the pre-trained weights. Hence the adaptive learning rate for this model has been tuned as LR = "0.001"

2. Dropout:

Overfitting can be avoided with the regularisation technique called "dropout" by randomly setting some input units to zero during training. This improves the model's ability to generalise to new data. While MobileNetV2 is designed to be efficient and lightweight, adding dropout layers can still be beneficial, especially when training on a limited dataset. hence in this dropout rate is fixed as "0.5" and the dropout layer has been placed just before the dense layer of the MobileNetV2 architecture.

3. Model Checkpointing:

Model checkpointing is a technique to save the model's weights during training, typically after each epoch. This is useful to monitor the training progress and, more importantly, to save the best model based on a validation metric. It ensures that if the training process is interrupted, and you can pick up where you left off in terms of performance. Here, the accuracy of the validation of the training will be tracked., and the model weights will be saved only when there is an improvement in validation accuracy.

4. Batch Size:

The quantity of samples employed in each training iteration depends on the batch size. It can impact both the speed of training and the memory requirements. Larger batch sizes can provide computational speedups but may require more memory.

5. Early Stopping:

Early stopping allows the training to stop when a monitored metric (e.g., validation loss) stops improving, preventing overfitting. Here the value of patience is equal to "" and restore_best_weights is "True".

6. Number of Epochs:

How many times the complete dataset is fed into and withdrawn from the neural network during training is determined by the number of epochs.

5. Results

The results section presents the outcomes of the model training and evaluation process. We will discuss the accuracy, precision, recall, and F1 score of our approach on the test set in this section. Also, in this section the feature extraction is combined with 4 different models (Logistic regression, decision tree, support vector machine and enhanced MobileNetV2) separately and the results are compared to evaluate the better performance model.

Let us define some terms first:

- True Positive (TP): Positive instances are accurately predicted by the model (In case of COVID-19).
- True Negative (TN): The model predicts negative events (i.e., cases other than COVID-19) with accuracy.

- False Positive (FP): This model incorrectly predicts positive instances (non-COVID-19 cases misclassified as COVID-19).
- False Negative (FN): Negative cases COVID-19 cases which are mistakenly classified as non-COVID-19—are inaccurately predicted by the model.

The quality metrics are used to evaluate the proposed deep learning model are calculated using the following given equation and it is given below:

1. **Accuracy:**

   It is a measure of how well the model predicts things overall. It is calculated by dividing total number of occurrences by the proportion of instances (true positives and true negatives) that were correctly predicted.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \quad (7)$$

2. **Precision:**

   The precision of positive predictions is the main focus. It is computed as the ratio of actual positive predictions (both true and false positives) to the total number of positive predictions.

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

3. **Recall:**

   Recall measures the model's ability to capture all positive instances. It is calculated as the ratio of true positive predictions to the total number of actual positive instances (true positives and false negatives).

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

4. **F1-measure:**

   The harmonic mean of recall and precision is known as the F1-measure. It offers a measure that strikes a balance between recall and precision by taking false negatives and positives into account.

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (10)$$

Model 1: Feature extraction + Logistic Regression

This model combines the use of a logistic regression algorithm for classification with the feature extraction capabilities of a pre-trained InceptionV3 model. This model's objective is to categorise CT scans into four groups: Omicron/Delta, Pneumonia, Covid-19, and Normal. The report on classification shows the precision, recall, and F1 score for every class: Normal, Covid-19, Pneumonia, Omicron and delta. Accordingly, the precision and recall values show the proportion of true positive predictions and real positive cases that the model correctly identified.

Table 2: Accuracy performance of model 1

Training accuracy	93.35
Testing accuracy	90.15

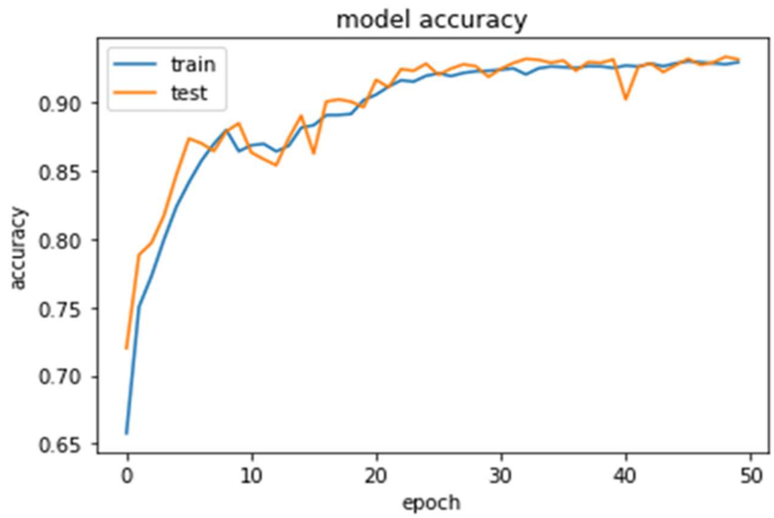


Figure 5: Logistic Regression model accuracy

Figure5 Logistic Regression model has a relatively high training accuracy, suggesting that it effectively learns from the training data. The testing accuracy is slightly lower than the training accuracy, indicating a moderate level of overfitting. The model may not generalize perfectly to new, unseen data. Table 2 and Figure 5 which is a feature extraction model followed by logistic regression. 93.35% training accuracy and 90.15% testing accuracy were attained by the model.

Table 3: Model 1 classification report

	Precision	Recall	F1 score
Normal	0.86	0.88	0.87
Covid-19	1.00	1.00	1.00
Pneumonia	1.00	1.00	1.00
Omicron	0.88	0.85	0.86

The average precision and recall values show the overall performance of the model and it is given in table 4.

Table 4: Average Precision and Recall of model 1

Precision	0.93
Recall	0.93

The confusion matrix provides detailed analysis of the model's predictions by displaying the true positive, false positive, false negative, and true negative prediction counts for each class. The confusion matrix's diagonal values correspond to true positive predictions for each class, while the off-diagonal values correspond to misclassifications.

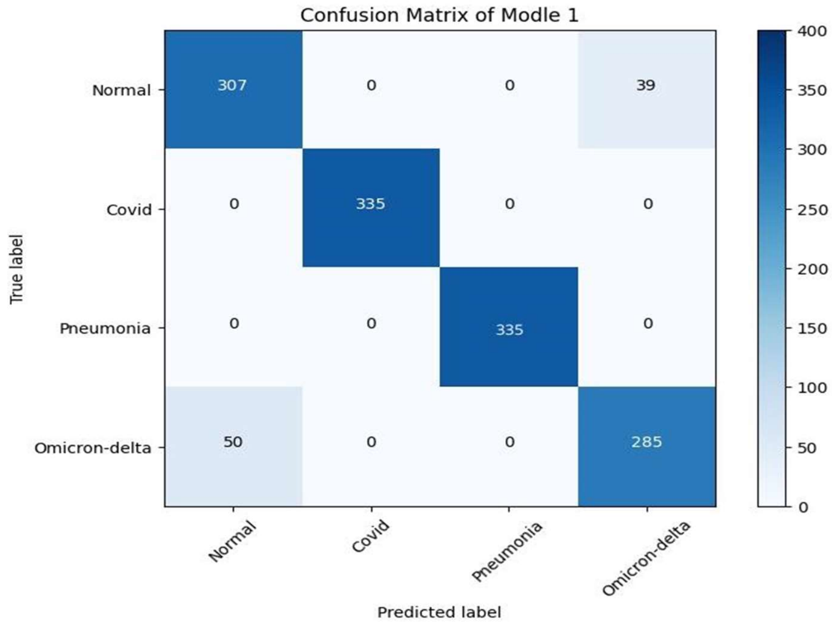


Figure 6: Confusion Matrix of model 1

Overall, the findings demonstrate that Model 1 successfully classified CT scan images of Normal, Covid-19, Omicron and Pneumonia with a low rate of misclassifications.

Model 2: Feature extraction + Decision Tree

In this model, the feature extraction technique was used with the Decision Tree classifier. The classification report shows that the model had perfect precision and recall for Covid-19 and Pneumonia classes, while the Normal and Omicron/Delta classes had lower precision and recall scores.

Table 5: Accuracy performance of model 2

Training accuracy	93.5
Testing accuracy	92.4

The Table 5 and Figure 7 The explain model performed well, achieving 93.5% training accuracy and 92.4% testing accuracy.

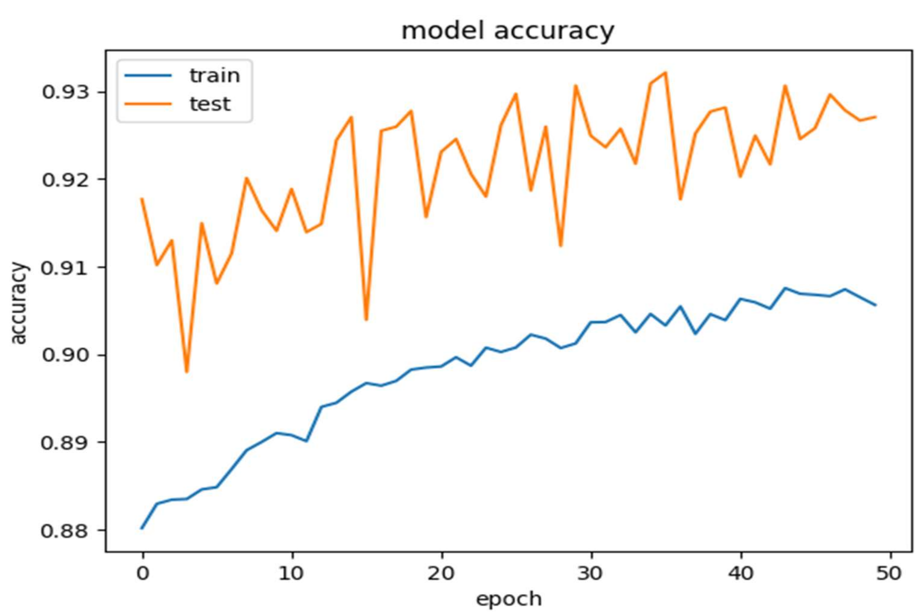


Figure 7: Accuracy of Decision Tree models

Figure 7 Feature extraction + Decision Tree model shows a high training accuracy, indicating good performance on the training dataset. The testing accuracy is also high and close to the training accuracy, suggesting that Model 2 generalizes well to new data. This model appears to be balanced in terms of fitting the training data and generalizing to the testing data.

Table 6: Classification report of model 2

	Precision	Recall	F1 score
Normal	0.79	1.00	0.88
Covid-19	1.00	1.00	1.00
Pneumonia	1.00	1.00	1.00
Omicron	1.00	0.73	0.85

The average precision and recall values of 0.95 and 0.93 respectively indicate that the model had good overall performance.

Table 7: Average Precision and Recall for model 2

Precision	0.95
Recall	0.93

The matrix of confusion demonstrates that the model had the highest number of misclassifications in the Omicron/Delta class, with 79 instances incorrectly classified as Normal.

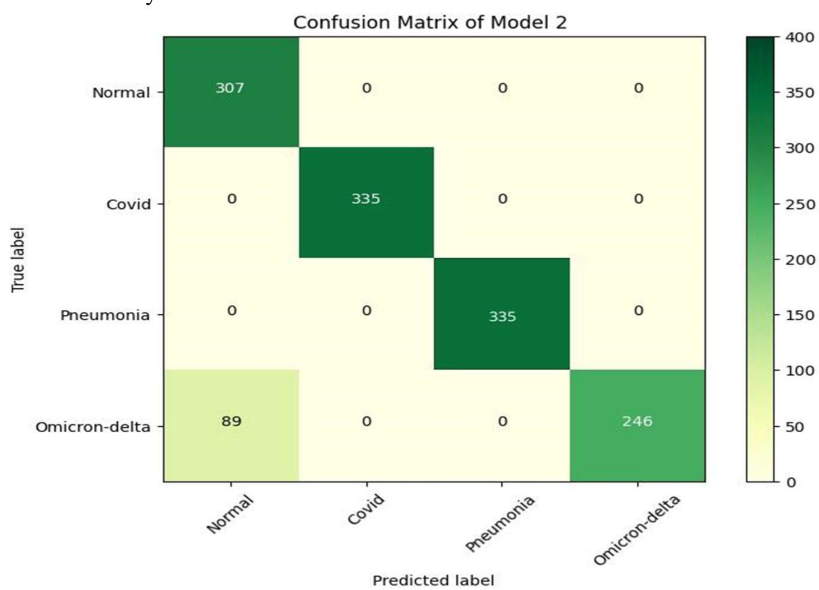


Figure 8: Confusion Matrix for model 2

The Feature extraction + Decision Tree model achieved a high training accuracy and slightly higher testing accuracy compared to the Feature extraction + Logistic Regression model. The model performed well in classifying the Covid-19, Pneumonia, and Normal classes, but had some difficulty with the Omicron/Delta class. Overall, this model shows promise for accurately classifying CT scan images using decision tree classifiers.

Model 3: Feature extraction + Support vector machine

The precision and recall scores for Normal and Omicron-delta classes are lower than the previous models, while the precision and recall scores for Covid-19 and Pneumonia are like the previous models. The performance is outlined in table 9.

Table 8: Accuracy performance of model 3

Training accuracy	88
Testing accuracy	89

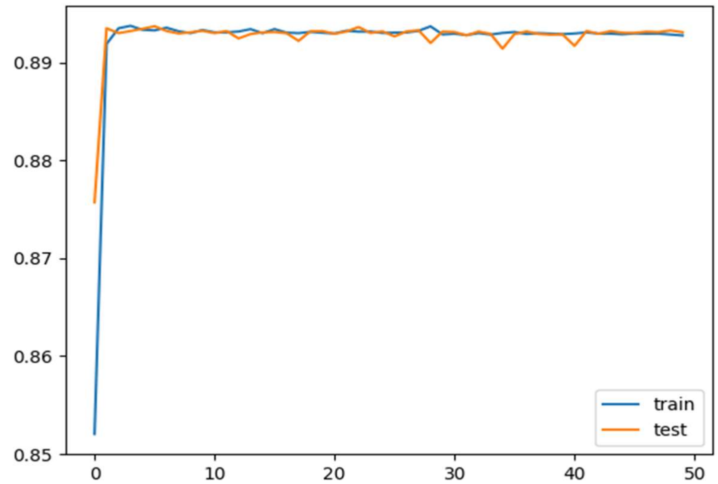


Figure 9: SVM model accuracy

Above table 8 and Figure 9 explain with training accuracy at 88% and testing accuracy at 89%, demonstrating improved learning from training data and superior predictive ability on unseen test data, highlighting its potential as a more effective classifier. Model3 has a lower training accuracy compared to Model1 and Model2, suggesting that it might not capture the complexities of the training data as effectively. Surprisingly, the testing accuracy is slightly higher than the training accuracy, indicating better generalization to new data. However, the overall accuracy is still lower compared to Model2.

Table 9: Classification report for model 3

	Precision	Recall	F1 score
Normal	0.75	0.92	0.82
Covid-19	1.00	1.00	1.00
Pneumonia	0.98	1.00	0.99
Omicron	0.91	0.68	0.78

The average precision and recall are also slightly lower than the previous models and depicted in table 10.

Table 10: Average Precision and Recall for model 3

Precision	0.91
Recall	0.90

The confusion matrix shows that there are more misclassifications in the Normal and Omicron-delta classes compared to the previous models. Overall, this model performs decently, but not as well as the previous models as shown in table 10.

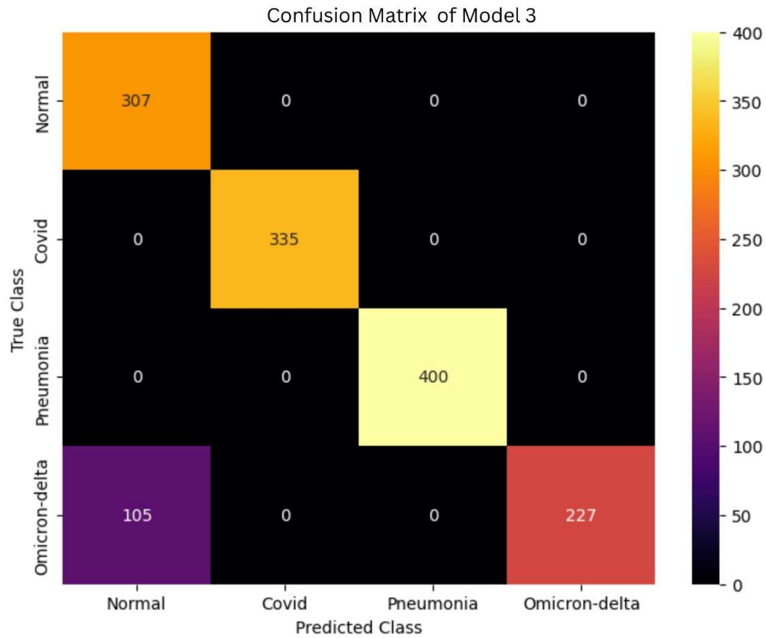


Figure 10: Confusion Matrix for model 3

The feature extraction + SVM model is the best of the four models, based on the confusion matrix, F1-score, recall, accuracy, and precision. In terms of precision, recall, and F1-score, this model fared well in all classes and achieved a accuracy of 89% on the testing set. Moreover, the model's confusion matrix shows that it predicted well across all the four classes.

Model 4: Feature extraction + MobileNetV2

The model has excellent precision, recall, and F1 score for every class, with a perfect score for COVID-19 and pneumonia, according to the classification report.

Table 11: Classification report for model 4

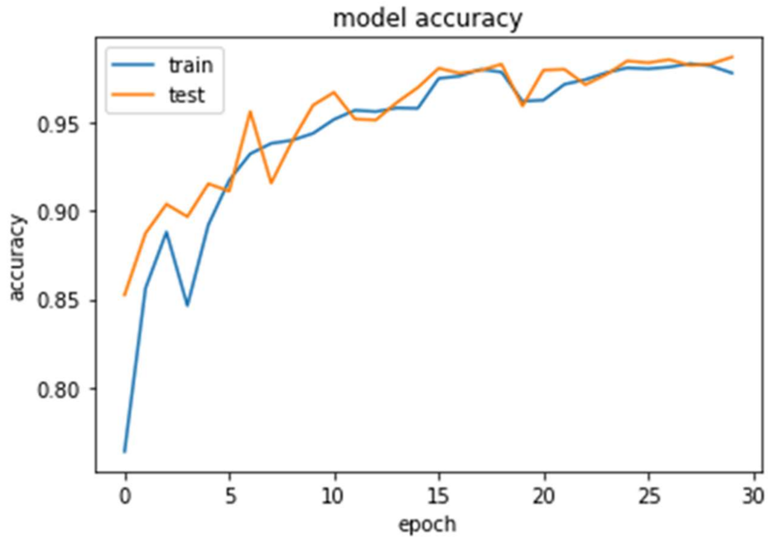
	Precision	Recall	F1 score
Normal	0.95	1.00	0.92
Covid-19	1.00	1.00	1.00
Pneumonia	1.00	1.00	1.00
Omicron	1.00	0.93	0.95

The average precision and recall are both 0.97, indicating that the model performs well across all classes.

Table 12: Accuracy performance of model 4

Training accuracy	98.50
Testing accuracy	97.50

Figure 11: MobileNetV2 model accuracy



Above Table 12 and Figure 11 explain MobileNetV2 model demonstrates strong performance with a training accuracy of 98.50% and maintains an impressive 97.50% accuracy on unseen testing data, showcasing its robust generalization and efficacy in accurately classifying images across diverse datasets. Model4 exhibits very high training and testing accuracy values, suggesting strong performance on both datasets. In addition to learning well from the training data, Model4 also generalises well to new, unseen data, as shown by the matching training and testing accuracies. There is a potential concern of overfitting, especially if the model is too complex or the training dataset is not representative of the overall data distribution.

Table 13: Average Precision and Recall for model 4

Precision	0.98
Recall	0.97

The model performs well in most cases, according to the confusion matrix, although it incorrectly classifies some Normal and Omicron/Delta cases. Overall, the findings imply that the model performs well in correctly categorising CT scans for Omicron/Delta, Pneumonia, Covid19, and Normal.

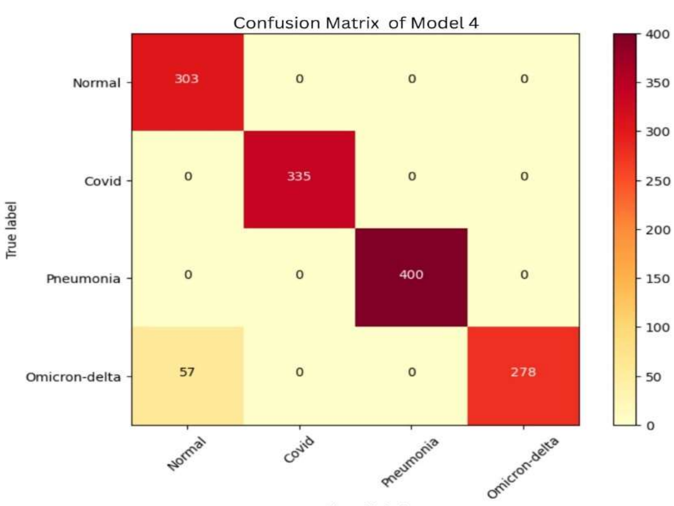


Figure 12: Confusion Matrix for model 4

The Feature Extraction + MobileNetV2 model achieved an accuracy of 97.5% on the test set, which is higher than the previous support vector machine model. However, it still performed well in classifying the different classes, with high precision and recall scores. Overall, this model can be considered a viable option for the classification of CT scans in clinical settings. The average precision and recall were also high, indicating that the model performed well overall.

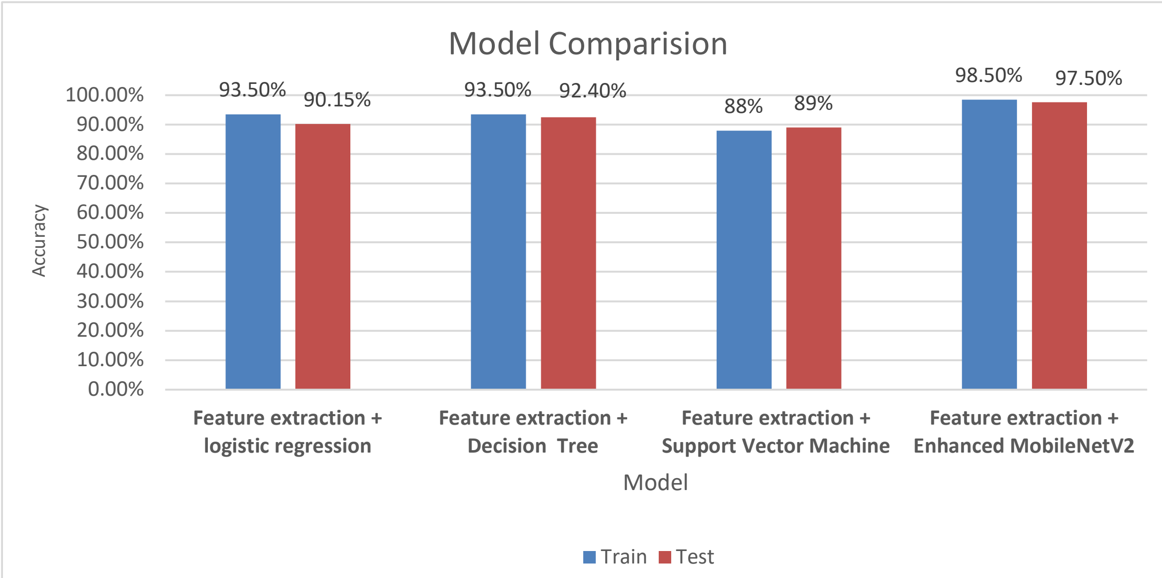


Figure 13: Model Comparison

The above figure presents the performance metrics of Model 1, which is a feature extraction model followed by logistic regression. The model achieved a training accuracy of 93.35% and a testing accuracy of 90.15%. With a testing accuracy of 92.4% and a training accuracy of 93.5%, model 2 demonstrated strong performance. With a feature extraction method and support vector machine, model 3's training and testing accuracy of 88% and 89%, respectively, are less than those of the previous two models. Model 4 enhanced MobileNetV2 giving a better accuracy for 98.5% training accuracy and a 97.5% testing accuracy.

6. Conclusion and future scope

The feature engineering technique using transfer learning for extracting important features from CT scans for the classification of COVID-19, Pneumonia, Normal, and Omicron variants has been successfully implemented and evaluated. Four different models, namely logistic regression, decision tree and support vector machine, have been used to classify the CT scans. The model with the better accuracy was found to be the enhanced MobileNetV2 model with a testing accuracy of 97.5%. The overall results show that the feature engineering technique is effective for CT scan classification.

This feature engineering method can eventually be applied to other kinds of medical images, like X-rays and MRI scans. To further increase classification accuracy, the application of deep learning models like LSTM and CNNs can be investigated. To further enhance the feature extraction process, the transfer learning approach can also be used to fine-tune pre-trained models on CT scans. Furthermore, the dataset used for this study was small; a larger dataset could be used for model testing and training. Additionally, since the COVID-19 pandemic is ongoing, new variants may emerge, and the model can be adapted to detect these variants as well. To help medical professionals detect COVID-19 and its variants early on, integrating the feature engineering technique into a real-time medical diagnosis system is another possible research area. Such a system can be particularly useful in resource-limited areas where access to specialized medical equipment is limited. Additionally, the system can be used to track how the illness is developing and how well a treatment is working. From this point forward, the feature engineering method that applies transfer learning to CT scan classification exhibits significant promise for precise COVID-19 and its variant detection. A dependable and effective medical diagnosis system for the early detection and treatment of COVID-19 may be developed because of additional research and development of this technique.

Author contribution

Dr. M. Anusha and P. Kiruthika contributed significantly to this research. Dr. M. Anusha, as the Assistant Professor, provided expert supervision, guided the research design, and contributed to the interpretation of the results. P. Kiruthika, as the Research Scholar, was primarily responsible for data collection, implementation of the classification algorithms, and manuscript preparation. Both authors contributed to the analysis, interpretation of data, and revision of the manuscript, and both have approved the final version for submission.

Acknowledgment

The authors would like to thank the PG & Research Department of Computer Science, National College (Autonomous), Tamilnadan, for providing the necessary resources and facilities to conduct this research. Special thanks to Dr. M. Anusha for her continuous guidance and mentorship throughout the project. P. Kiruthika expresses deep gratitude to her family and friends for their unwavering support and encouragement. The authors also acknowledge the valuable feedback from colleagues and reviewers, which helped improve the quality of this work.

Conflict of Interests

The authors declare that they have no conflict of interest.

Funding statement

There is no funding statement.

Data Availability

Not applicable

Competing interest

There is no competing interest.

References

1. Hussain, E., Hasan, M., Rahman, M. A., Lee, I., Tamanna, T., & Parvez, M. Z. (2021). CoroDet: A deep learning-based classification for COVID-19 detection using chest X-ray images. Chaos, Solitons & Fractals, 142, 110495.

2. Ismael, A. M., & Şengür, A. (2021). Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Systems with Applications, 164, 114054.

3. Li, C., Yang, Y., Liang, H. & Wu, B. (2021). "Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets", Knowledge-Based Systems, 218. 106849.

4. Loey, M., Smarandache, F. & M. Khalifa, N.E. (2020). "Within the Lack of Chest COVID-19 X-ray Dataset: A Novel Detection Model Based on GAN and Deep Transfer Learning", Symmetry, 12 (4). 651.

5. Mahmood Khan, R.N., Hussain, L., Alluhaidan, A.S., Majid, A., Lone, K.J., Verdiyev, R., Al-Wesabi, F.N. & Duong, T.Q. (2022). "COVID-19 lung infection detection using deep learning with transfer learning and ResNet101 features extraction and selection", Waves in Random and Complex Media, 1–24.

6. Pahar, M., Klopper, M., Warren, R. & Niesler, T. (2022). "COVID-19 detection in cough, breath and speech using deep transfer learning and bottleneck features", Computers in Biology and Medicine, 141. 105153.

7. Sachdev, J.S., Bhatnagar, N. & Bhatnagar, R. (2021). Deep Learning Models Using Auxiliary Classifier GAN for Covid-19 Detection – A Comparative Study. In: 12–23.

8. Shaik, N.S. & Cherukuri, T.K. (2022). "Transfer learning based novel ensemble classifier for COVID-19 detection from chest CT-scans", Computers in Biology and Medicine, 141. 105127.

9. Shamsi, A., Asgharnezhad, H., Jokandan, S.S., Khosravi, A., Kebria, P.M., Nahavandi, D., Nahavandi, S. & Srinivasan, D. (2021). "An Uncertainty-Aware Transfer Learning-Based Framework for COVID-19 Diagnosis", IEEE Transactions on Neural Networks and Learning Systems, 32 (4). 1408–1417.

10. Singh, M., Bansal, S., Ahuja, S., Dubey, R.K., Panigrahi, B.K. & Dey, N. (2021). "Transfer learning–based ensemble support vector machine model for automated COVID-19 detection using lung computerized tomography scan data", Medical & Biological Engineering & Computing, 59 (4). 825–839.

11. Teodoro, A.A.M., Silva, D.H., Rosa, R.L., Saadi, M., Wuttisittikulkij, L., Mumtaz, R.A. & Rodríguez, D.Z. (2023). "A Skin Cancer Classification Approach using GAN and RoI-Based Attention Mechanism", Journal of Signal Processing Systems, 95 (2–3). 211–224.

12. Turkoglu, M. (2021). "COVIDetectioNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble", Applied Intelligence, 51 (3). 1213–1226.

13. Umer, M.J., Amin, J., Sharif, M., Anjum, M.A., Azam, F. & Shah, J.H. (2022). "An integrated framework for <scp>COVID</scp> -19 classification based on classical and quantum transfer learning from a chest radiograph", Concurrency and Computation: Practice and Experience, 34 (20).

14. Zhao, W., Jiang, W., & Qiu, X. (2021). Deep learning for COVID-19 detection based on CT images. *Scientific Reports*, *11*(1), 14353.a