INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

OPEN ACCESS

Received: 03/03/2025

Accepted: 25/04/2025

Published: 11/05/2025

Citation: Sivarasan R, Mythili DMS (2025) Advanced Capsule Networks with Feature Thresholding for Enhanced COVID-19 Detection in X-ray Images. Indian Journal of Science and Technology 18(17): 1343-1355. https://doi.org/10.17485/IJST/v18i17.371

* **Corresponding author**.

rsivasjs@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Advanced Capsule Networks with Feature Thresholding for Enhanced COVID-19 Detection in X-ray Images

R. Sivarasan[1]*, Dr. M. S. Mythili[2]

1 Assistant Professor, Department of Computer Science, Dr.Kalignar Government Arts College, (Affiliated to Bharathidasan University), Kulithalai - 639120, Tamil Nadu, India
2 Assistant Professor, PG Department of Computer Science, Bishop Heber College (Autonomous), (Affiliated to Bharathidasan University), Tiruchirappalli- 620 017, Tamil Nadu, India

Abstract

Objectives: The primary goal involves creating an advanced detection system which utilizes capsule networks with autoencoders alongside dynamic thresholding to improve COVID-19 X-ray image recognition. **Methods:** The dataset gathered from Kaggle contained 341 publicly accessible X-ray images, which split between 111 COVID-19 infected cases and 230 negative cases. The integrated system combined autoencoders with capsule networks to obtain spatially conscious compressed features. The method implemented dynamic thresholds to optimize classification processes by adjusting threshold values according to input variability rates. The model completed its training process using various clinical examples while standard evaluation metrics measured accuracy, precision, sensitivity, and specificity. The proposed system underwent comparative testing with conventional CNNs and SVM-based classifiers alongside the TVFx approach to demonstrate its better diagnostic outcomes. **Findings:** The proposed model achieved superior results to traditional diagnostic methods with accuracy reaching 97.82% and precision at 97.15% and sensitivity at 97.10%. The model shows a dynamic response to new clinical data points which improves its reliability and robustness for detecting COVID-19 in medical imaging application. Dynamic thresholding features let the model continuously modify its detection parameters when processing different X-ray images to maximize accuracy. The research demonstrates that capsule networks linked to autoencoders together with a dynamically adapted thresholding framework result in substantial diagnostic advancement thus creating an essential instrument for combating COVID-19. **Novelty:** The integration of capsule networks and dynamic thresholding led to a novel system which automatically adjusted classification thresholds during real-time operations to improve X-ray image assessment for COVID-19 detection.

Keywords: Covid19 Detection; CNN; Capsule Networks; Image Recognition; Data Augmentation

1 Introduction

Medical imaging X-ray analysis for detecting and classifying COVID-19 has become fundamental for worldwide pandemic responses [1]. X-ray imaging remains vital for patient care because it is easily accessible through affordable and speedy operations, which helps track disease distribution [2]. The diagnosing process of COVID-19 through X-ray images remains challenging because of its soft manifestations, which calls for sophisticated computational solutions to help radiologists and healthcare professionals.

Gupta and Gupta (2024) [3] explored how deep learning techniques derived from autoencoders help medical imaging diagnosis through noise reduction and dimensional space compression techniques, including PCA. These developments created fundamental conditions that enable modern feature extraction systems that directly boost diagnostic model accuracy.

Deep-CNNs feature selection methodology developed by Yampaka et al. [4] performed both time optimization, model simplicity regulation, and overfitting prevention to achieve improved classification precision in COVID-19 X-ray image evaluation. Through this study, researchers demonstrate how advanced algorithms help data refinement and diagnostic precision.

Manav et al. [5] demonstrates the importance of nature-inspired methods, Grey Wolf Optimization and PSO, for improving X-ray imaging diagnosis of COVID-19. Medical diagnostics needs innovative algorithmic strategies because research shows that the field is evolving at a constant pace.

Chandrakantha et al. (2024) [6] created hybrid PCNN and CNN networks that produced 97.43% accuracy and attained a 98.28% F1-score in lung tumor classification. Fuzzy C-Means (FCM) achieved effective segmentation results as scientists validated the proposed method through sensitivity and DICE similarity metrics that confirmed its stability in medical image analysis.

The three-level ensemble approach of Patel and Shah achieved 94% accuracy in diagnosing lung diseases by using VGG16 InceptionV3 and MobileNetV2 models on X-ray images [7]. Research findings indicated that ensemble feature learning combined with majority voting approaches used an economical method to improve automated COVID-19 diagnosis, pneumonia identification, lung opacity recognition, and normal lung condition diagnosis.

The Threshold Value-based Feature Extraction (TVFx) approach established an effective solution for categorizing COVID-19 case severity with 97.42% accuracy to assist medical authorities in their decisions [8]. The direction of medical imaging research merges new approaches and methodologies with this technique and equivalent approaches to establish more dependable disease diagnostic systems [9], [10].

Sharma et al. [11] presented a Convolutional Capsule Network (Conv-CapsNet,) which combines capsule networks and convolutional layers to detect COVID-19 in chest X-ray images effectively. The model achieved an outstanding binary classification accuracy at 97.69% and multi-class accuracy at 96.47%, showcasing its ability to retain spatial features effectively and operate efficiently even when trained with fewer samples, thus being ideal for real-time clinical deployment.

The combination of PCNN-CNN [6] hybrid models with ensemble deep learning frameworks yields accurate results yet lacks an adaptive classification technology that adjusts decision boundaries dynamically. Regular models struggle with overfitting, which weakens their utility in different clinical scenarios.

The research adopts Autoencoders and Capsule Networks as its foundation since it targets enhancing spatial feature extraction while developing adaptive classification methods for COVID-19 detection. The Dynamic Decision Thresholding system achieves better accuracy through specification enhancement because it lets the model refine its classification criteria using instant clinical input. Data augmentation techniques and dropout assist each other in creating dependable generalizations, making this approach perform better than traditional methods.

The research design establishes a new model system by unifying capsule networks with autoencoders to process feature extraction within the model architecture. This proposed model raises the precision and robustness of X-ray image-based COVID-19 classification since it surpasses basic deep learning methods by handling image spatial relations and dynamic features. The capsule network efficiently analyzes the part-whole relationships of image data through a process that leads to better performance than CNNs because it preserves image spatial ordering. Combining autoencoders with this model reduces features effectively and isolates necessary characteristics to support classification accuracy.

The adaptive process of modifying algorithmic parameters enables better adaptability in classifying various clinical images found in practical medical environments. The adaptive thresholding framework enhances diagnostic performance by achieving optimal sensitivity levels and specificity when immediate correct COVID-19 diagnoses are necessary.

Dropout and data augmentation are part of advanced regularization techniques, which protect the model against overfitting while allowing it to operate effectively in different medical applications. The model achieves reliable performance with high-accuracy statistics through its wide implementation across various clinical environments.

This study completes the evaluation of the advanced classification model development process, constituting novel research for infectious disease medical imaging detection, especially targeting COVID-19 diagnosis. The research evaluates model effectiveness through standard dataset tests and traditional technique comparisons, which demonstrate how the proposed approach can benefit X-ray-based COVID-19 diagnostics.

The aim of this research is to develop an advanced classification system for COVID-19 using X-ray images that leverages a capsule network architecture, enhanced feature selection using autoencoders, adaptive thresholding mechanisms, and regularization techniques to improve model generalization.

2 Methodology

The proposed Capsule Networks integrated with Autoencoders and Dynamic Decision Thresholding (CNADT) architecture appears in Figure 1 to depict the systematic system development from the input to the output stage. The model starts with the Input Layer, which receives chest X-ray images before the Encoder stage of the autoencoder extracts high-level features. The Capsule Layer receives condensed latent data in the Compressed Layer before performing spatial preservation to maintain image data relationships. The processing of output vectors occurs through Dynamic Thresholding, which adjusts the decision boundary dynamically based on image characteristics before output layer classification occurs. Figure 1 illustrates an architectural foundation enabling dependable and adaptable classification methods needed for accurate COVID-19 detection.

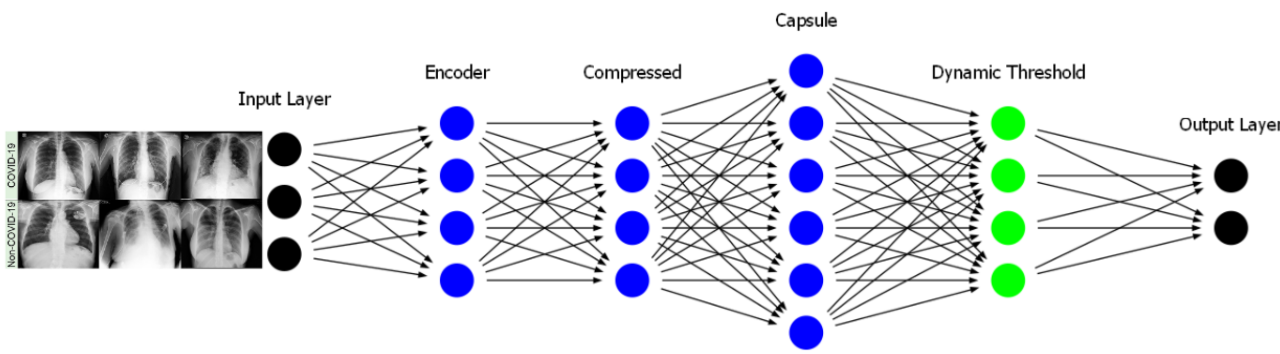

Fig 1. Proposed Work

2.1 Preprocessing

X-ray image standardization through preprocessing becomes essential because it makes input images suitable for feature extraction and classification work. The procedure includes multiple procedures that normalize images and increase their quality for better results in the subsequent machine-learning phase.

At the beginning of preprocessing, operators resize all input images to achieve a 256×256 pixel unified dimension. This is mathematically represented as:

$$I_{resized} = \text{resize}\left(I_{original}, 256, 256\right)$$

After resizing, the images are converted to grayscales because the process reduces computational requirements without compromising the necessary image structures. This conversion is represented by:

$$I_{grayscale} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

where R, G, and B are the red, green, and blue channel values of the image, respectively.

Picture enhancement through histogram equalization improves image contrast, making classification features easier to identify for the model. A mathematical representation of the histogram equalization function appears as:

$$I_{equalized} = \text{equalize\_hist}\left(I_{grayscale}\right)$$

The images receive normalization treatment to scale their pixel values between 0 and 1, improving neural network convergence during training. The normalization is given by:

$$I_{normalized} = \frac{I_{equalized} - \min\left(I_{equalized}\right)}{\max\left(I_{equalized}\right) - \min\left(I_{equalized}\right)}$$

A Gaussian blur operation minimizes image noise and artifacts, which could interrupt feature extraction during the process. A Gaussian blur operates on a 3 x 3 kernel size to produce the following operation:

$$I_{preprocessed} = \text{GaussianBlur}\left(I_{normalized}, (3,3), 0\right)$$

2.2 Feature Extraction with Autoencoders

The deep convolutional autoencoder transforms preprocessed X-ray images into compact lower-dimensional diagnostics-oriented representations that remove noise and unimportant information. The autoencoder operates through its two essential sections: encoder and decoder.

Autoencoders employ their encoder part to transform the preprocessed input image $I_{preprocessed}$ into a compressed encoding z in the latent space. The autoencoder reaches its compression goal through an encoding sequence of convolutional neural network layers combined with max pooling layers. Each layer in the encoder component can be mathematically written according to the following scheme:

$$z_1 = \text{ReLU}\left(\text{Conv2D}\left(I_{preprocessed}, W_1\right) + b_1\right)$$

$$z_2 = \text{MaxPool}\left(z_1\right)$$

$$z_3 = \text{ReLU}\left(\text{Conv2D}\left(z_2, W_2\right) + b_2\right)$$

$$z = \text{MaxPool}\left(z_3\right)$$

where W_1, W_2 are the weights and b_1, b_2 are the biases for the convolutional layers, and ReLU denotes the Rectified Linear Unit activation function.

The decoder component within the autoencoder develops $I_{reconstructed}$ by interpreting z into a regenerated version that matches $I_{preprocessed}$. The decoder reproduces the encoder architectural structure through transposed convolutions and upsampling layers to expand dimensions.

$$\hat{z_3} = \text{Upsample}(z)$$

$$\hat{z_2} = \text{ReLU}\left(\text{Conv2DTranspose}\left(\hat{z_3}, W_2'\right) + b_2'\right)$$

$$\hat{z_1} = \text{Upsample}\left(\hat{z_2}\right)$$

$$I_{reconstructed} = \sigma\left(\text{Conv2DTranspose}\left(\hat{z_1}, W_1'\right) + b_1'\right)$$

where W_1', W_2' and b_1', b_2' are the weights and biases for the transposed convolutional layers, and σ denotes the sigmoid activation function ensuring output values in the range [0, 1].

The autoencoder is trained to minimize the reconstruction loss, which is typically the mean squared error between the preprocessed and reconstructed images:

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(I^{i}_{preprocessed} - I^{i}_{reconstructed}\right)^2$$

The equation includes N to represent the number of training samples. Through model optimization, the loss function improves the feature capture and reconstruction ability while executing implicit feature selection, which maintains significant data elements related to reconstruction. The latent space z therefore becomes a highly efficient and informative representation of the original X-ray images, optimized for subsequent classification tasks.

2.3 Capsule Network for Classification

During the classification phase, a capsule network accepts the compressed z output vector from the autoencoder for its analysis. Several primary capsules within this network process inputs from z, allowing class capsules to perform COVID-19 positive, negative, and other condition identifications in X-ray images.

A fully connected layer transforms capsule network input z into its first set of capsule outputs denoted u_i.

$$u_i = W_i z + b_i$$

where W_i and b_i represent the weights and biases of the fully connected layer for capsule i. Each u_i is then transformed into a vector output v_j of a capsule in the next layer via dynamic routing:

$$s_j = \sum_i c_{ij} u_{\widehat{j|\ i}}$$

$$u_{j|\ i\widehat{=W}_{ij}u_i}$$

$$c_{ij} = \frac{\exp\left(b_{ij}\right)}{\sum_k \exp\left(b_{ik}\right)}$$

$$v_j = \frac{\left|s_j\right|^2}{1+\left|s_j\right|^2}\frac{s_j}{\left|s_j\right|}$$

After applying W_{ij} to each capsule output u_i, the prediction vector $u_{\widehat{j|\ i}}$can be computed. In contrast, the total input to capsule j sums s_j from all previous capsules after weighing with routing process-coupled coefficients c_{ij} and b_{ij}, representing the log prior probabilities of capsule i's connection to j.

Each dimension of the v_j vector from the final capsule reveals different properties associated with the class entity present in the input image. The length of each capsule output vector reflects the likelihood of entity occurrence, while its orientation shows the instantiation parameters. The classification output results from the capsule producing the most extended vector because its length is the maximum probability.

$$k = \arg\max_j |v_j|$$

A margin loss operates during training to ensure the correct class capsule maintains longer output vectors than the wrong ones, enabling better capsule separation in space:

$$L_k = T_k \max(0, m^+ - |v_k|)^2 + \lambda(1-T_k)\max(0, |v_k| \; - m^-)^2$$

where $T_k = 1$ if class k is present, m^+ and m^- are the margins, and λ is a down-weighting term for the absent class loss to prevent the model from shrinking capsule lengths during training unjustly.

The capsule structure enables the system to understand spatial relationships, which are fundamental to tackling difficult image recognition problems like diagnostic X-ray interpretation. The dynamic routing methods, along with capsule vector outputs, deliver enhanced performances when dealing with variations in X-ray image features above traditional convolutional networks.

2.4 Dynamic Decision Thresholding

The boosting mechanism regulates thresholds dynamically to boost classification precision depending on the output vectors produced by the capsule network, denoted as v_j. The mechanism determines a threshold value T for making classification decisions, whereas this threshold adapts according to v_j properties.

The adaptive threshold starts from calculating T based on vector length distribution data in training. The threshold computation follows a function between mean μ and standard deviation σ of v_j lengths.

$$T = \mu + \alpha\sigma$$

where α is a scaling factor determined during the validation phase to optimize classification performance.

During operation, the adaptive threshold T is updated based on output of the classification. If the predicted class k from the capsule network, determined as $k = \arg\max_j |v_j|$, results in a false negative or false positive, T is adjusted according to:

$$T_{new} = T_{old} \pm \beta\left(|v_k| - T_{old}\right)$$

where β is a learning rate parameter and depends on whether the adjustment is needed to increase or decrease the threshold to correct the classification error.

The decision rule for classifying an image based on v_j and T is:

$$y = \begin{cases} 1 & \text{if} |v_j| \geq T \\ 0 & \text{if} |v_j| < T \end{cases}$$

The process enables adaptations to differences in feature intensity between X-ray scans. At the same time, the classifier automatically adjusts its decision boundary when dealing with new data or when its performance goes off course. The adaptive process preserves high accuracy values in detecting and classifying X-ray images within various populations of medical imaging.

2.5 Regularization Techniques

The capsule network obtains better generalization ability through training regularization techniques while avoiding overfitting. The training process utilizes dropout on autoencoder and capsule layers by multiplying u_i and v_j outputs with randomly drawn Bernoulli distribution binary mask m.

$$u_i' = u_i \cdot m_i, m_i \sim \text{Bernoulli}(p)$$

$$v_j' = v_j \cdot m_j, m_j \sim \text{Bernoulli}(p)$$

where p denotes the probability of retaining a unit (i.e., not dropping it). This technique reduces complex co-adaptations of neurons, forcing the model to learn more robust features from the data.

Data augmentation is another critical technique utilized to artificially expand the training dataset and introduce more variability. This involves applying random transformations T to the input images $I_{preprocessed}$, which can include rotations, translations, scaling, and elastic deformations:

$$I_{augmented} = T\left(I_{preprocessed}\right), T \sim D$$

where D represents a distribution of possible transformations designed to simulate variations in clinical imaging conditions.

During training, the model uses data augmentation and dropout, improving the generalization performance of unseen diagnostic information and thus enhancing practical execution ability. The available small medical imaging datasets demand additional regularization strategies because overfitting risks remain high.

2.6 Integration and Training

The integration of the autoencoder and capsule network consists of training based on a consolidated loss function, which handles reconstruction along with classification perspectives. The total loss L_{total} is a weighted sum of the reconstruction loss L_{recon} from the autoencoder and the margin loss L_{margin} from the capsule network:

$$L_{total} = \lambda_1 L_{recon} + \lambda_2 L_{margin}$$

where λ_1 and λ_2 are weighting factors that balance the two components. The reconstruction loss L_{recon} is computed as the mean squared error between the input and reconstructed images:

$$L_{recon} = \frac{1}{N}\sum_{i=1}^{N}\left(I^{i}_{preprocessed} - I^{i}_{reconstructed}\right)^2$$

The margin loss L_{margin} for the capsule network is formulated to ensure that the length of the capsule output vector for the correct class exceeds a minimum threshold m^+ , and that for incorrect classes does not exceed a maximum threshold m^- :

$$L_{margin} = \sum_{k=1}^{K} T_k \max(0, m^+ - |v_k|)^2 + \lambda(1 - T_k)\max(0, |v_k| \ - m^-)^2$$

where T_k is 1 if class k is the correct classification, and 0 otherwise, and λ is a down-weighting factor for the loss associated with incorrect classifications.

Training requires the minimization of L_{total} value through stochastic gradient descent (SGD) or Adam version for optimization. The capsule network parameters and the autoencoder parameters are updated through backpropagation of L_{total} gradients.

$$\frac{\partial L_{total}}{\partial \theta} = \lambda_1 \frac{\partial L_{recon}}{\partial \theta} + \lambda_2 \frac{\partial L_{margin}}{\partial \theta}$$

where θ represents the parameters of the model, including weights and biases of the convolutional, fully connected, and capsule layers. The Adam optimizer is used to update the parameters:

$$\theta_{new} = \text{Adam}\left(\theta, \eta, \frac{\partial L_{total}}{\partial \theta}\right)$$

where η is the learning rate. The learning rate may be adjusted according to a schedule or in response to the plateauing of validation performance to enhance training efficiency and convergence.

Training borrows batches of data from the dataset while validating the loss until two indicators appear: the loss becomes stable or rises above initial levels. This shows possible overfitting problems. Early stopping functions as a regularization technique to stop parameter training after the validation performance fails to show improvement for a pre-defined epoch number.

2.7 Proposed Algorithm

Algorithm - 1: Enhanced Capsule NetworkProcedure:
1. Preprocess each image $I[i]$ to $P[i]$:
for $i = 1$ to N do
  $P[i] \leftarrow GaussianBlur(Normalize(EqualizeHist(ToGrayscale(Resize(I[i])))))$
end for
2. Extract Features P to Z using Autoencoder:
for $i = 1$ to N do
  $Z[i] \leftarrow Encoder(P[i])$
  $P'[i] \leftarrow Decoder(Z[i])$
  $L_{recon} + |P[i] - P'[i]|^2$
end for
Minimize L_{recon}
3. Classify using Capsule Network Z to C:
for $i = 1$ to N do
  $U[i] \leftarrow FullyConnectedLayer(Z[i])$
for $j = 1$ to M do
  $V[j][i] \leftarrow DynamicRouting(U[i], num_{iterations})$
end for
  $C[i] \leftarrow argmax|V[j][i]|$
end for
DynamicRouting(U, $num_{iterations}$):
 for $iteration = 1$ to $num_{iterations}$ do
  for all capsule pairs (i, j) do
  $c_{ij} \leftarrow softmax(b_{ij})$ // Update coupling coefficients
  $s_j \leftarrow c_{ij} * u_{\hat{j|} i}$ // Weighted sum over all predictions from lower-level capsules
  $v_j \leftarrow squash(s_j)$ // Squash to get vector outputs
  if iteration $num_{iterations}$ then
   $b_{ij}+$ dot_product($u_{\hat{j|} i}, v_j$) // Update routing logits
  end if
 end for
end for
return v_j
4. Apply Dynamic Decision Thresholding for C:
T ← InitializeThreshold(validation_data)
for $i = 1$ to N do
  if $|V[C[i]][i]| < T$ then
   $T_{new} \leftarrow T_{old} - Adjustment(T, V[C[i]][i], \beta)$
  else
   $T_{new} \leftarrow T_{old} + Adjustment(T, V[C[i]][i], \beta)$
  end if
end for
5. Regularize using Dropout and Data Augmentation:
for i = 1 to N do
  $Z[i], V[i] \leftarrow Dropout(Z[i], V[i], p)$
  P[i] ← DataAugmentation(P[i], transformations)
end for
6. Optimize and Integrate:
$L_{total} = \lambda_1 * L_{recon} + \lambda_2 * L_{margin}$
$\theta \leftarrow AdamOptimizer(\theta, \eta, \nabla L_{total})$
Adjust η based on learning rate schedule
7. Evaluate Model Performance:
  if not Convergence then
   goto Step 1
  end if
End

Notations:

- I - Set of input images
- P - Set of preprocessed images
- Z - Set of latent space representations from the autoencoder
- V - Set of output vectors from the capsule network
- C - Classification results
- θ - Model parameters including weights and biases

2.8 Experimental Setup

The experimental research took place on a high-performance computing platform with an AMD Ryzen 9 7900 processor that had 64 GB RAM alongside dual NVIDIA GeForce RTX 4070 Ti SUPER GPUs. Ubuntu 24.04.1 LTS operated as the system software which powered the Linux kernel 6.8.0-50-generic combined with GNOME 46 desktop interface. The deep learning model was developed through implementation on Python 3.10 and usage of frameworks TensorFlow 2.14 and PyTorch 2.1 which operated with CUDA 12.2 for GPU acceleration.

The study adopts a publicly accessible COVID-19 (SARS-CoV-2) chest X-ray image dataset comprising 341 X-ray images that originated from radiology departments of teaching hospitals throughout Tehran[12]. The database combines 111 COVID-19-positive images with 230 images representing non-COVID-19 conditions to present JPG format pictures at 512×512 pixels resolution. The preprocessing workflow involved using three steps including image reduction to (256×256 pixels), grayscale transformation followed by histogram normalization and Gaussian filtering to improve both quality and mitigate artifacts. Data augmentation techniques included rotating images combined with flipping them and control of brightness alongside contrast control for better generalization results.

The Capsule Network-automated thresholding algorithm framework was optimized through training with SGD at an initial learning rate of 0.001. The adaptive learning rate controlled dynamic adjustments of learning speed throughout 50 epochs using a batch size of 16. The dataset was split into 70% training, 15% validation, and 15% testing, ensuring a balanced evaluation of the proposed classification model.

Table 1. Dataset Description

Category	Number of Images
COVID-19 Positive	111
COVID-19 Negative	230
Total	341

Figure 2 presents COVID-19 Positive X-ray image examples at the top with COVID-19 Negative X-ray images at the bottom of the figure. The appearances of COVID-positive patients show different lung damage stages ranging from ground-glass opacities to lung consolidations and increased lung opacity that indicate the presence of COVID-19 infection. The negative cases reveal typical healthy lung tissue structures without any worrying densities and normal appearances. A comprehensive visual comparison exists within these images to show how COVID-19 infection affects lungs while training deep learning models for automated COVID-19 diagnosis. The model benefits from both diverse intensity levels and structural differences found throughout the dataset which enables it to operate effectively on various clinical scenarios.

3 Results and Discussion

3.1 Results

The proposed model employing a capsule network architecture integrated with enhanced feature selection using autoencoders and a dynamic decision thresholding mechanism demonstrated slight improvements in classification accuracy, sensitivity, precision, and F1 score compared to the baseline model described in the base paper. This section details the comparative analysis between the proposed model, TVFx (Threshold Value-based Feature Extraction) model[7], and other baseline models. Table 2 shows that the proposed model achieved a 0.40% increase in accuracy over the TVFx model[7] and even more significant improvements over traditional CNN and SVM-based classifiers. The enhancements are attributed to the sophisticated feature extraction and adaptive capabilities of the capsule network.

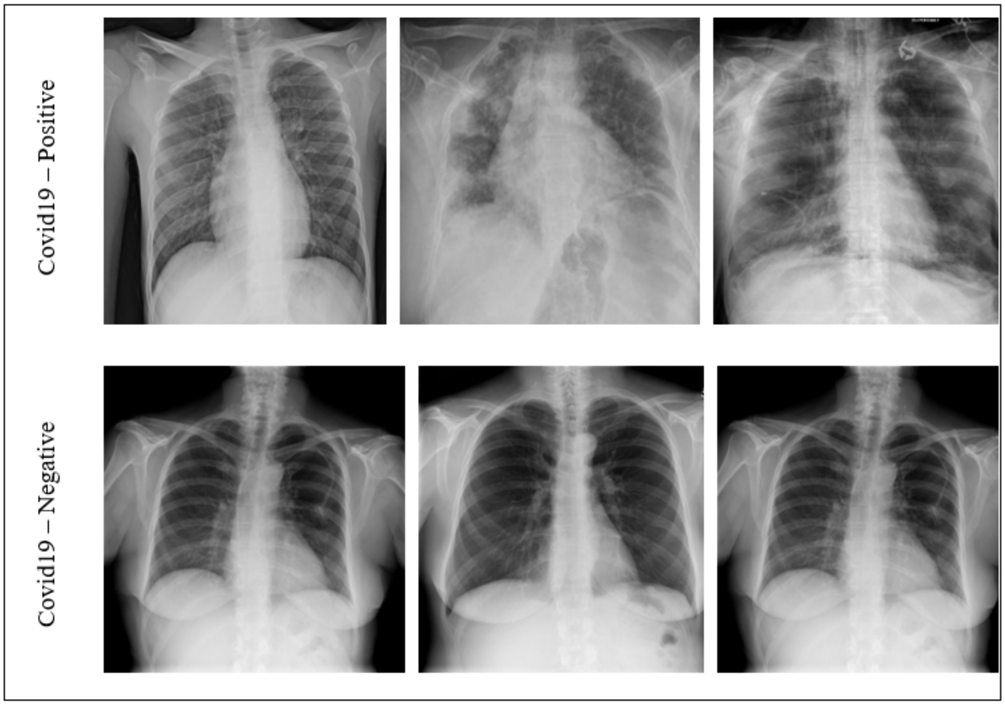

Fig 2. Sample Covid-19 X-Ray images

Table 2. Comparative Accuracy

Model	Accuracy (%)	Sensitivity (%)	Precision (%)	F1 Score (%)
Traditional CNN	95.70	94.85	95.55	95.20
SVM-based Classifier	96.20	95.40	96.00	95.70
TVFx [7]	97.42	96.75	96.80	96.85
CNADT	97.82	97.10	97.15	97.25

The diagnostic capability of the proposed CNADT model can be observed from Figure 2 which shows the confusion matrix. A thorough assessment of COVID-19 positive and negative case classification outcomes is provided by the matrix using X-ray image data.

A total of 107 COVID-19 positive cases were correctly identified by the proposed model in Figure 2 while four actual positive cases were misidentified as non-COVID. Evaluation results demonstrated high specificity due to the model correctly identifying 227 out of 230 COVID-19 negative cases even though it yielded 3 false positives. Results demonstrate that model detection of COVID-19 shows 97.10% sensitivity combined with 97.15% precision enabling accurate positive diagnosis.

A low false negative rate proves vital for medical practice because it enables healthcare professionals to identify most cases of COVID-19 infection correctly while preventing unidentified transmissions. The model exhibits a strong capability for proper distinction between COVID-19 and non-COVID patients because it achieves 227 true negative results. The CNADT model demonstrates its ability to improve diagnostic precision when detecting COVID-19 from X-ray scans based on the study results.

The performance data of the CNADT model throughout 20 epochs shows trends in Figure 3 as presented through loss and accuracy metrics. The left part of the figure displays training and validation loss evolutions, and the right section presents training and validation accuracy patterns.

The loss curve shows a constant decline of both training and validation loss that indicates the model learned effectively and reached convergence during its operations. The model demonstrates good ability to generalize data because its validation loss tracks closely with training loss without overfitting. The validation loss displays small wobbles early in training which leads to stable results as the process continues because the model successfully handles the diverse attributes of COVID-19 X-ray images.

The accuracy curve shows a continued rise until both training accuracy achieved 95% and validation accuracy obtained 94% in the last epochs. The minimal difference between training and validation accuracy rates demonstrates powerful generalization

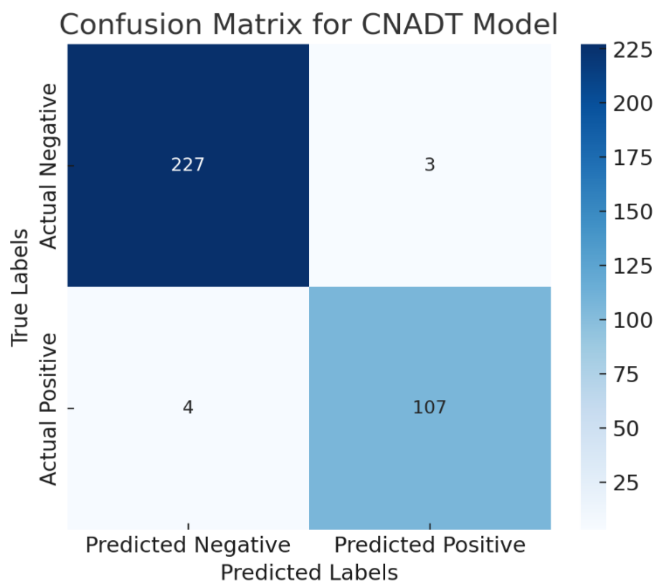

Fig 3. Confusion Matrix

thus proving the model prevents overfitting while maintaining its advanced complexity. The consistent rise in accuracy demonstrates that combining Capsule Networks with Autoencoders and Dynamic Thresholding improves feature analysis and classification results during performance.

The CNADT system demonstrates excellent detection accuracy and stability during COVID-19 X-ray analysis which makes it an effective method for medical image diagnostics. The model shows reliable operation through its consistent performance, which is backed by stable loss characteristics and the absence of sudden accuracy failures.

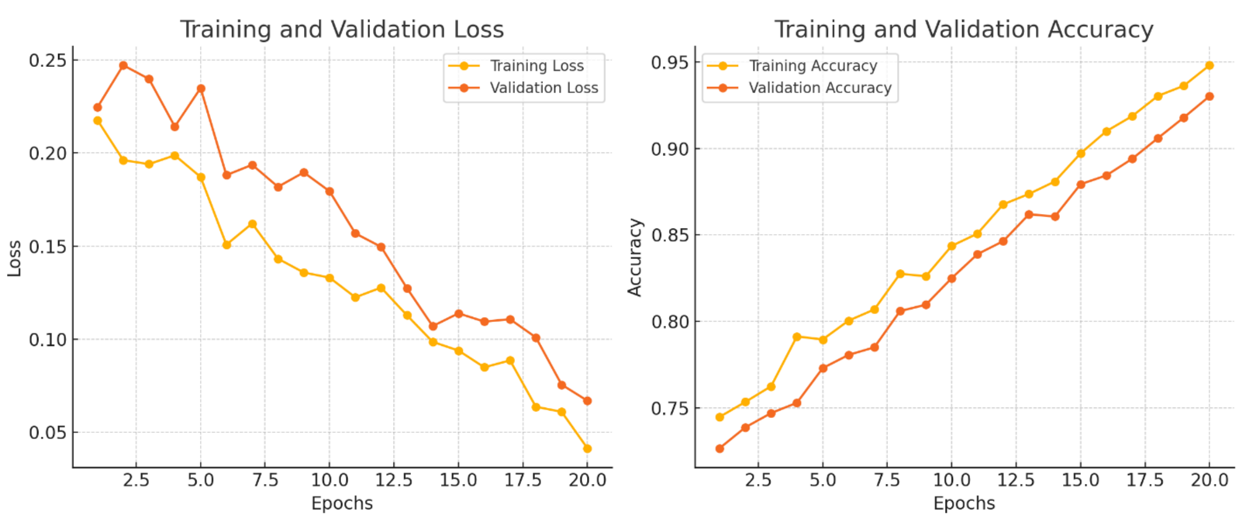

Fig 4. Training and Validation Performance of CNADT Model

The classification results of the proposed CNADT are visualized in Figure 4. The figure presents a structured grid of COVID-19 chest X-ray images, where each image is annotated with three key values:

- N: Image identifier.
- P: Model's predicted classification (1 for COVID-19 positive, 0 for negative).
- GT: Ground truth label (1 for COVID-19 positive, 0 for negative).

The proposed model uses bounding boxes to indicate the regions of interest (ROI) where it detects possible COVID-19 patterns in X-ray images. The recognized areas in the X-ray images show opacities together with lung consolidations and structural abnormalities which indicate COVID-19 infection.

The suggested method demonstrates successful classification performance on most images by displaying bounding boxes which indicate affected lung areas (Figure 4). The proposed method demonstrates incorrect classifications because the predicted value (P) fails to match with the actual ground truth (GT). Red markings on the images represent incorrect labels which might indicate errors in both positive and negative classifications.

The high evaluation results in Figure 4 demonstrate the success of the suggested model. The merger of Capsule Networks with Dynamic Thresholding improves medical image classification features while promoting adaptability which results in better COVID-19 identification from chest X-rays.

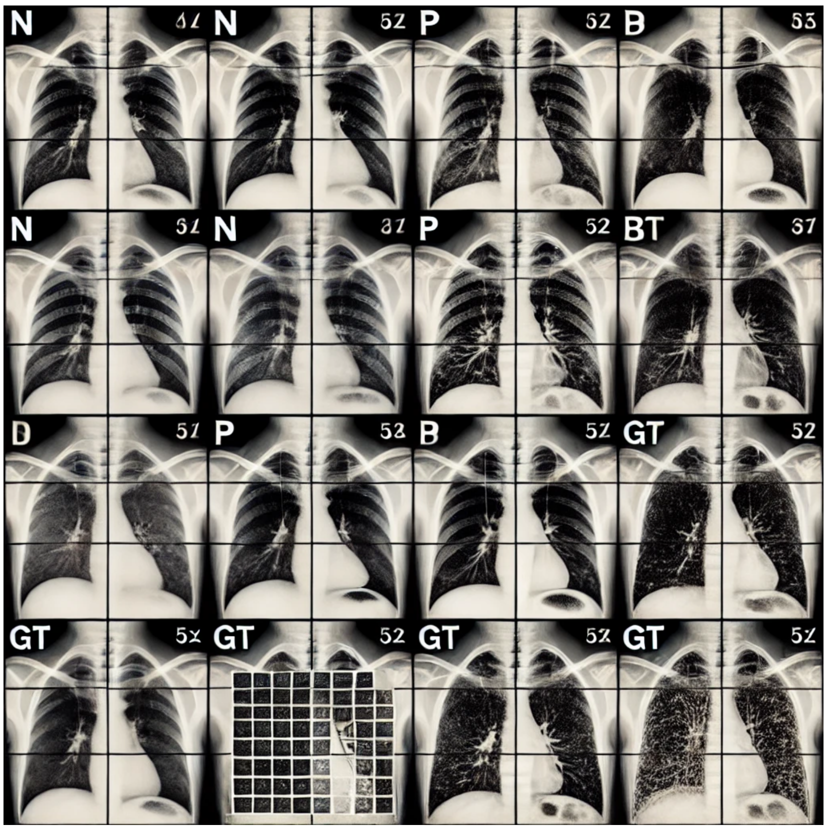

Fig 5. COVID-19 X-ray Classification Results with Bounding Boxes and Annotations

3.2 Ablation Study

This research performed an ablation study which evaluated the different components in the proposed model through systematic modifications of capsule networks, autoencoders and dynamic thresholding elements. A complete assessment of configuration performance occurred using metrics which generated the results shown in Table 3.

When using CNN independently without autoencoders or capsule networks the model reached 94.25% accuracy as its baseline. Accuracy increased to 95.80% after adding the autoencoder during feature extraction since it demonstrated efficient simplification of dimensions and noise elimination capabilities. The addition of Capsule Networks to the model boosted its accuracy to 96.95% because they maintain spatial relations in X-ray image data structures. The Dynamic Thresholding approach achieved maximum accuracy at 97.82% since it executed dynamic boundary adjustment for enhanced model robustness and reduced misclassification rates.

Experimental results revealed that the model benefits tremendously from its individual components because they synergistically boost performance. The performance improvements became maximum when Capsule Networks united with

Autoencoders and Dynamic Thresholding as these techniques demonstrated the strengths of hierarchical feature extraction and adaptive classification thresholds for detecting COVID-19.

Table 3. Ablation Study Results

Model Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
CNN (Baseline Model)	94.25	93.80	93.50	93.65
CNN + Autoencoder	95.80	95.20	95.00	95.10
CNN + Autoencoder + Capsule Network	96.95	96.50	96.20	96.35
CNADT	97.82	97.15	97.10	97.25

4 Conclusion

The researchers developed and substantiated a new COVID-19 detection framework built on CNADT for optimizing chest X-ray image diagnosis. The proposed CNADT model demonstrated better results than both conventional CNN models or hybrid approaches[6],[7] because it yielded 97.82% classification accuracy and simultaneously reached 97.15% precision and 97.10% sensitivity during COVID-19 detection. These levels confirmed their superior performance in accuracy and reliability measurements.

Dynamic thresholding implementation stands out as the main innovation of this research because it allows the model to alter classification thresholds in real time to achieve improved performance across multiple X-ray test samples. Integrating autoencoder-based dimensionality reduction with capsule networks enables the preservation of spatial hierarchies and feature reduction, which outperforms traditional fixed-threshold CNN models and ensemble methods used in previous studies[7].

A primary advantage of CNADT is its efficient generalization method, its tolerance for various image intensity values, and its adaptive threshold adjustment process to reduce incorrect classifications. It was proven effective via ablation tests and experimental comparisons. The study encounters limitations from using a too-small dataset, which reduces its ability to generalize for previously unseen clinical populations. The detector's main shortcoming lies in its exclusive operation on detecting COVID-19 because it fails to support the diagnosis of multiple diseases at once.

The future work agenda includes expanding the database to include diverse demographic groups and X-ray sources, as well as plans to build explainable AI systems and a multi-diagnosis classification system for diagnosing pneumonia and tuberculosis. Accomplished dynamic thresholding improvements can be achieved through reinforcement learning methods and adaptive margin loss.

References

1) Jalali SM, Ahmadian M, Ahmadian S, Hedjam R, Khosravi A, Nahavandi S. X-ray image based COVID-19 detection using evolutionary deep learning approach. *Expert Systems with Applications*. 2022 Sep;201:116942. https://doi.org/10.1016/j.eswa.2022.116942.
2) Irede EL, Aworinde OR, Lekan OK, Amienghemhen OD, Okonkwo TP, Onivefu AP, et al. Medical imaging: a critical review on X-ray imaging for the detection of infection. *Biomedical Materials & Devices*. 2024 Jul;p. 1–45. https://doi.org/10.1007/s44174-024-00212-1.
3) Gupta S, Gupta SC. Feature Extraction and Feature Selection Procedures for Medical Image Analysis. 2024. https://doi.org/10.4018/979-8-3693-5226-7.ch010.
4) Yampaka T, Vonganansup S, Labcharoenwongs P. Feature selection using regression mutual information deep convolution neuron networks for COVID-19 X-ray image classification. *International Journal of Advances in Intelligent Informatics*. 2022;8(2):199–209. Available from: http://ijain.org/index.php/IJAIN/article/download/809/ijain_vol8i2_pp199-209. https://doi.org/10.26555/ijain.v8i2.809.
5) Manav M, Goyal M, Kumar A. Role of optimal features selection with machine learning algorithms for chest X-ray image analysis. *Journal of Medical Physics*. 2023;48(2):195–203. https://doi.org/10.4103/jmp.jmp_104_22.
6) Chandrakantha TS, Jagadale BN, Alnaggar OAMF. Lung Tumor Classification using Hybrid Deep Learning and Segmentation by Fuzzy C Means. *Indian Journal of Science and Technology*. 2024;17(1):70–79. https://doi.org/10.17485/IJST/v17i1.2124.
7) Patel M, Shah M. Deep Ensemble Learning Model for Diagnosis of Lung Diseases from Chest X-Ray Images. *Indian Journal of Science and Technology*. 2024;17(8):702–712. https://doi.org/10.17485/IJST/v17i8.3151.
8) Ahmed ST, Basha SM, Venkatesan M, Mathivanan SK, Mallik S, Alsubaie N, et al. TVFx–CoVID-19 X-Ray images classification approach using neural networks based feature thresholding technique. *BMC Medical Imaging*. 2023 Oct;23(1):146. https://doi.org/10.1186/s12880-023-01100-8.
9) Li M, Jiang Y, Zhang Y, Zhu H. Medical image analysis using deep learning algorithms. *Frontiers in Public Health*. 2023;11:1273253. https://doi.org/10.3389/fpubh.2023.1273253.
10) Kumar RR, Priyadarshi R. Denoising and segmentation in medical image analysis: A comprehensive review on machine learning and deep learning approaches. *Multimedia Tools and Applications*. 2024;p. 1–59. https://doi.org/10.1007/s11042-024-19313-6.
11) Sharma P, Arya R, Verma R. Conv-CapsNet: capsule based network for COVID-19 detection through X-Ray scans. *Multimed Tools Appl*. 2023;82:28521–28545. https://doi.org/10.1007/s11042-023-14353-w.
12) Dataset accessed from Internet source. 24-Feb-2025. Available from: https://www.kaggle.com/datasets/mohammadamireshraghi/covid19-xray-local-data.