

Original paper

An adaptive convolution neural network model for tuberculosis detection and diagnosis using semantic segmentation

Sayali Abhijeet Salkade^{A,B,C,D,E,#}, Sheetal Vikram Rathil^{1,E,F,#}

Thakur College of Engineering and Technology, Mumbai, India

[#]These authors had equal contribution to this work.

Abstract

Purpose: Tuberculosis (TB) continues to be a major cause of death from infectious diseases globally. TB is treatable with antibiotics, but it is often misdiagnosed or left untreated, particularly in rural and resource-constrained regions. While chest X-rays are a key tool in TB diagnosis, their effectiveness is hindered by the variability in radiological presentations and the lack of trained radiologists in high-prevalence areas. Deep learning-based imaging techniques offer a promising approach to computer-aided diagnosis for TB, enabling precise and timely detection while alleviating the burden on healthcare professionals. This study aims to enhance TB detection in chest X-ray images by developing deep learning models. We have observed upper and lower lobe consolidation, pleural effusion, calcification, cavity formation and military nodules. A proposed preprocessing technique has been also introduced in our work based on gamma correction and gradient based technique for contrast enhancement. We leverage the Res-UNet architecture for image segmentation and introduce a novel deep learning network for classification, targeting improved accuracy and precision in diagnostic performance.

Material and methods: A Res-UNet segmentation model was trained using 704 chest X-ray images sourced from the Montgomery County and Shenzhen Hospital datasets. Following training, the model was applied to segment lung regions in 1400 chest X-ray scans, encompassing both TB cases and normal controls, obtained from the National Institute of Allergy and Infectious Diseases (NIAID) TB Portal program dataset. The segmented lung regions were subsequently classified as either TB or normal using a deep learning model. A gradient based technique was used for contrast enhancement by capturing intensity changes in image by comparing each pixel with its neighbour with pyramid reduction unique mapping and histogram matching along with gamma correction is used. This integrated approach of segmentation and classification aims to enhance the accuracy and precision of TB detection in chest X-ray images. Classification of segmented images was done using customised convolutional neural network, and visualisation was done using Grad-CAM.

Results: The Res-UNet model demonstrated excellent performance for segmentation, achieving an accuracy of 98.18%, recall of 98.40%, precision of 97.45%, F1-score of 97.97%, Dice coefficient of 96.33%, and Jaccard index of 96.05%. Similarly, the classification model exhibited outstanding results, with a classification accuracy of 99.45%, precision of 99.29%, recall of 99.29%, F1-score of 99.29%, and an AUC of 99.9%. Enhanced gradient based method showed ambe of 16.51, entropy of 6.7370, CII of 86.80, psnr of 28.71, ssim of 86.83 which are quite satisfactory.

Conclusions: The findings demonstrate the efficiency of our system in diagnosing TB from chest X-rays, potentially surpassing clinician-level precision. This underscores its effectiveness as a diagnostic tool, particularly in resource-limited settings with restricted access to radiological expertise. Additionally, the modified Res-UNet model demonstrated superior performance compared to the standard U-Net, highlighting its potential for achieving greater diagnostic accuracy.

Key words: tuberculosis, artificial intelligence, deep learning, chest X-ray segmentation, chest X-ray classification.

Correspondence address:

Prof. Sayali Abhijeet Salkade, Thakur College of Engineering and Technology, Mumbai, India, e-mail: salkadesayali6@gmail.com

Authors' contribution:

A Study design · B Data collection · C Statistical analysis · D Data interpretation · E Manuscript preparation · F Literature search · G Funds collection

Introduction

Tuberculosis (TB), caused by the bacterium *Mycobacterium tuberculosis*, is a highly infectious disease that primarily targets the lungs but can also affect other organs such as the kidneys, spine, and brain. The disease spreads through airborne particles expelled when an infected person coughs, sneezes, or talks, contributing to its high transmissibility. This study presents a novel approach to improving TB detection by leveraging chest X-ray (CXR) image datasets. The proposed method, TbCNN-net, integrates semantic segmentation with an adaptive convolutional neural network (CNN) architecture to enhance diagnostic precision. An India “TB Report 2024” by the Ministry of Health and Family Welfare reported estimated TB case numbers of 27.8 lakhs with 25.5 lakh in 2024, showing an increase from 24.2 lakh cases in 2023. There are various factors of risk of TB like undernourishment, diabetes, use of alcohol, HIV, smoking, etc. Ahmed *et al.* [1] introduced TB U-net with an attention mechanism for efficient segmentation of lungs. They also proposed a densenet-169 CNN for accurate classification of TB, COVID-19, and pneumonia. Nafisah [2] proposed TB detection by using 9 different CNN models with U-net segmentation and EfficientNetB3, achieving higher accuracy of 99.1% with receiver operating characteristic (ROC) of 99.9%, recall of 98.8%, precision of 98.8%, and f1_score of 98.8%. Also, it involved different visualisation techniques like Grad-CAM and t-SNE [3]. Sharma *et al.* [4] proposed an U-Net model for segmentation with accuracy of 96.35%, Jaccard index of 90.38%, and DCE coefficient of 94.88%, and a classification model of Xception with accuracy of 99.29%, precision of 99.30%, and recall of 99.29%. They also used Grad-CAM to show the heatmap pattern and developed a framework with 9 different CNN models, among which CheXNet was the best performing model with accuracy of 96.47%, sensitivity of 98.56%, precision of 98.57%, and f1_score of 98.56%. Rajaraman *et al.* [5] trained X-rays using a U-Net model on TBX11 CXR dataset. Shallow models like VGG16 and VGG-19 were used for classification with staple consensus region of interest for region localisation on Montgomery TB and Shenzhen TB datasets [6]. This study trained and evaluated 10 different deep CNNs for distinguishing TB cases from normal cases. The models included ResNet50, ResNet101, ResNet152, InceptionV3, VGG16, VGG19, DenseNet121, DenseNet169, DenseNet201, and Mobile Net. The approach presented here leverages histogram-matched CXR images, eliminating the need for object segmentation of regions of interest. This approach enhances the accuracy and detection performance of CNN models for TB identification by incorporating histogram matching with CXR images. Chavan *et al.* [7] proposed Res-UNet++ outperformed U-Net and Res Net architectures, achieving superior evaluation metrics, including a validation Dice coefficient of 96.36%, validation mean IoU of 94.17%, and valida-

tion binary accuracy of 98.07% [8]. A novel algorithm has been developed to refine segmented masks, enhancing final accuracy, alongside an efficient fuzzy inference system for more weighted activity assessment. Dao and Lin [9] gives a comparison between U-Net, UNet++, and Trans-UNet+ [10]. ConvNet achieved sensitivity 87.0%, f1-score of 87.0%, 88.0% precision [11]. The algorithm achieved outstanding performance in detecting pulmonary tuberculosis (PTB), with an area under the receiver operating characteristic curve (AUC) of 0.878 and a 95% confidence interval (CI) ranging from 0.854 to 0.900 in the NTUH-20 dataset [12]. Swin ResuNet3 was used for segmentation of images, and a Multi-scale Attention-based DenseNet with an Extreme Learning Machine model was used for classification of TB [13]. An attention block integrating VGG16 and VGG19 architectures is employed for classifying TB CXRs. In the study by Verma *et al.* [14], a deep learning model was proposed for TB diagnosis, incorporating deep learning features alongside Gabor and Canny edge detection in one approach, and excluding these techniques in another, achieving accuracies of 95.7% and 97.9%, respectively [15]. Le *et al.* [15] used 5 pretrained models: VGG16, EfficientB7, MobileNetV3, DenseNet121, and RegNetY040 for classification of TB, among which MobilenetV3 showed the best performance with an accuracy of 98.35%, f1-score of 98.32% on TB CXR, Montgomery accuracy of 77.81%, and f1-score of 78.92%, and on the Shenzhen dataset – an accuracy of 67.19% and f1-score of 74.86%, on the India CXR – accuracy of 86.25% and f1-score of 83.75%.

Material and methods

Datasets and preprocessing

For the segmentation task, we utilised the Montgomery County and Shenzhen Hospital datasets, both obtained from the publicly available “Chest X-ray Masks and Labels” dataset on Kaggle. The combined dataset comprised 704 CXR images, each paired with corresponding segmentation masks. The data was split into 494 images (70%) for training, 140 images (20%) for validation, and 70 images (10%) for testing. To ensure a robust evaluation, we implemented a five-fold cross-validation strategy, allowing each image to be used at least once for both training and testing. For the classification task, we used the “National Institute of Allergy and Infectious Diseases (NIAID) TB Portal Program” dataset, which is part of the “Tuberculosis Chest X-ray Images” dataset on Kaggle. This dataset included 700 CXR images of TB patients and 3500 control images. For our experiment, we selected 700 TB images and 700 control images, which were divided into 1120 images (80%) for training, 140 images (10%) for validation, and 140 images (10%) for testing. Figure 1 shows sample images from this dataset.

Before being input into the model, the images were normalised by scaling the pixel values to a [0, 1] range, achieved

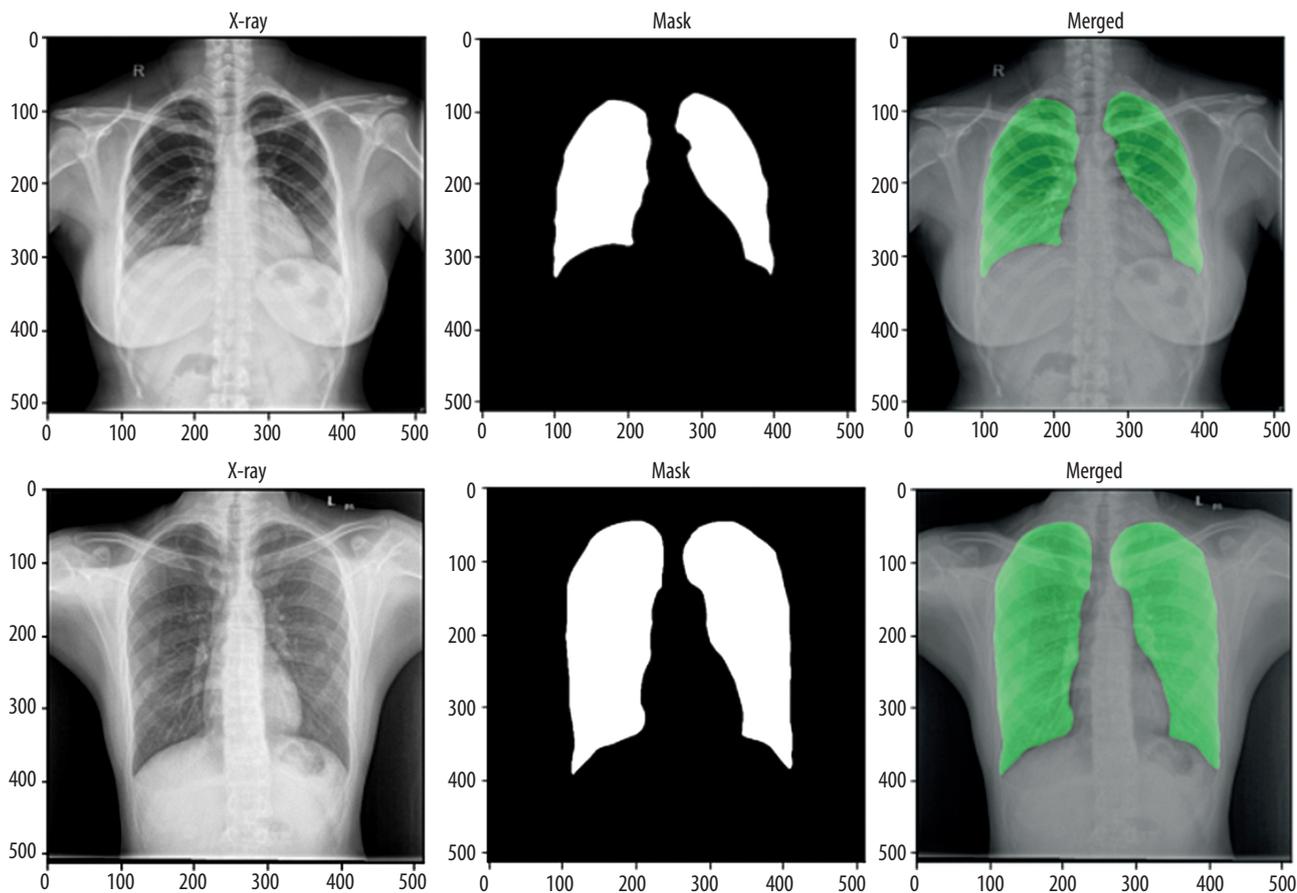


Figure 1. Sample images from datasets

by dividing each pixel by 255. This normalisation process ensures uniformity across the dataset, optimising the model's training performance. Also a gradient based contrast enhancement technique with gamma correction is used (Figure 2).

Methods

Figure 2 outlines the methodology employed in this study. We started by selecting a well-established chest X-ray (CXR) dataset, which included both the images and their corresponding segmentation masks for training the segmentation model. Before segmentation an enhanced gradient based technique for contrast enhancement was used.

Enhanced gradient-based contrast enhancement technique with gamma correction

The gradient computation function captures intensity changes by comparing each pixel with its neighbours in 4 directions, emphasising edge-like features. Pyramid reduction processes the image at multiple resolutions to capture gradients at different scales, mimicking multi-scale edge detection. Unique mapping and histogram matching adjust pixel values based on gradient information, enhancing contrast by redistributing intensities. Finally, gamma correction (with a value of 0.5) boosts brightness in darker regions, further improving contrast.

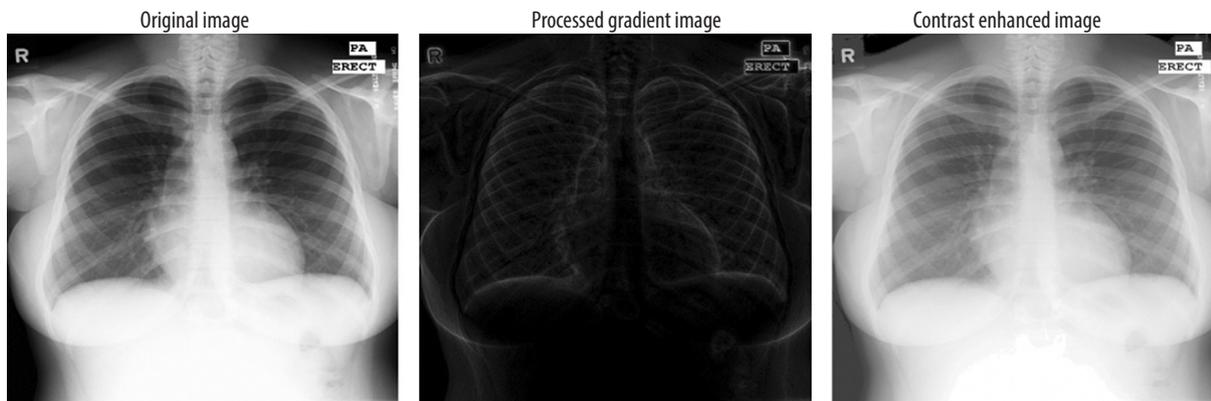


Figure 2. Gradient-based contrast enhancement

Steps for enhanced gradient-based contrast enhancement:

1. Take the original image as input and show the original image in greyscale.
2. Reduce the resolution by the `cv2.pyrDown()` function and perform padding and gradient initialisation.
3. Do gradient computation by padding the input image, traversing each pixel, extract a 3×3 sub image, and compute the differences between the centre pixel and its neighbours.
4. The gradients for original and reduced images should be computed at multiple levels.
5. Perform normalisation of the gradient by ensuring a minimum value and computing the geometric mean.
6. Display gradient-based images.
7. Unique mapping and histogram matching.
8. A final contrast image is created and displayed.

Next, we modified and optimised the model to improve its performance. The process began with the selection of a well-established CXR dataset, which included both images and their corresponding segmentation masks, for training of the segmentation model. The U-Net architecture, a popular model for segmentation tasks, was modified and optimised to align with the study's objectives. The performance of the trained segmentation model was assessed using metrics such as accuracy, Dice coefficient, Jaccard index, and AUC. The model was then applied to CXR datasets containing both TB and normal cases to

segment the lung regions while excluding background information. Following segmentation, a deep learning CNN was developed and fine-tuned to classify the segmented lung images. The classification model's performance was evaluated using various metrics, including accuracy, precision, recall, F1-score, AUC, and ROC curves.

U-Net model (segmentation model)

The U-Net model is a widely recognised and highly effective architecture for image segmentation, particularly in medical applications such as segmenting organs, tissues, or lesions in medical scans like CT, MRI, and CXR. It was originally designed for **bio-medical image segmentation** but has since become widely adopted for various segmentation tasks. Originally developed by Olaf Ronneberger in 2015, U-Net has become one of the most popular and effective models for segmentation tasks in medical imaging, especially for analysing radiological images.

Figure 3 illustrates the U-Net architecture, which comprises 2 main components: the encoder and the decoder. The encoder employs a sequence of convolutional layers combined with ReLU activation and max-pooling operations. Conversely, the decoder reconstructs segmented images using up-convolution layers, depth concatenation, softmax activation, and max pooling. The process starts with a 224×224 pixel input image, which is processed by the encoder to extract features represented as numerical

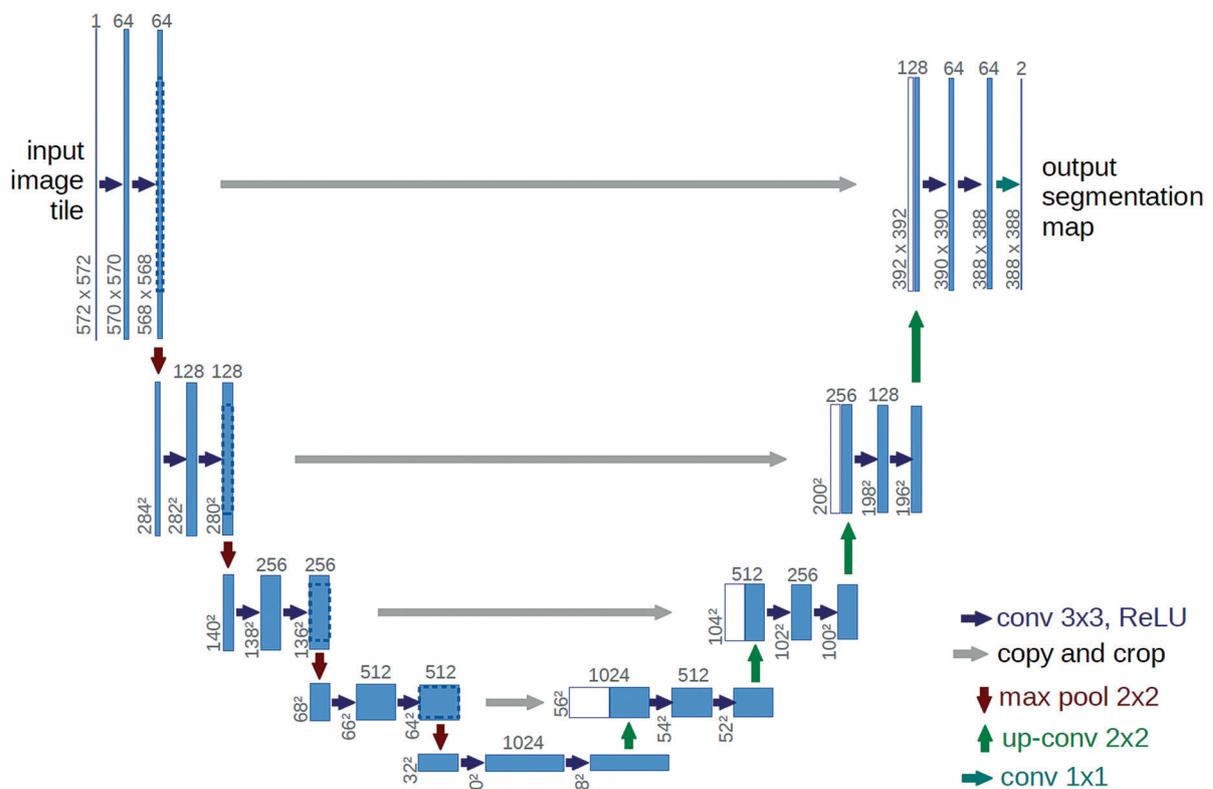


Figure 3. Architecture of U-Net

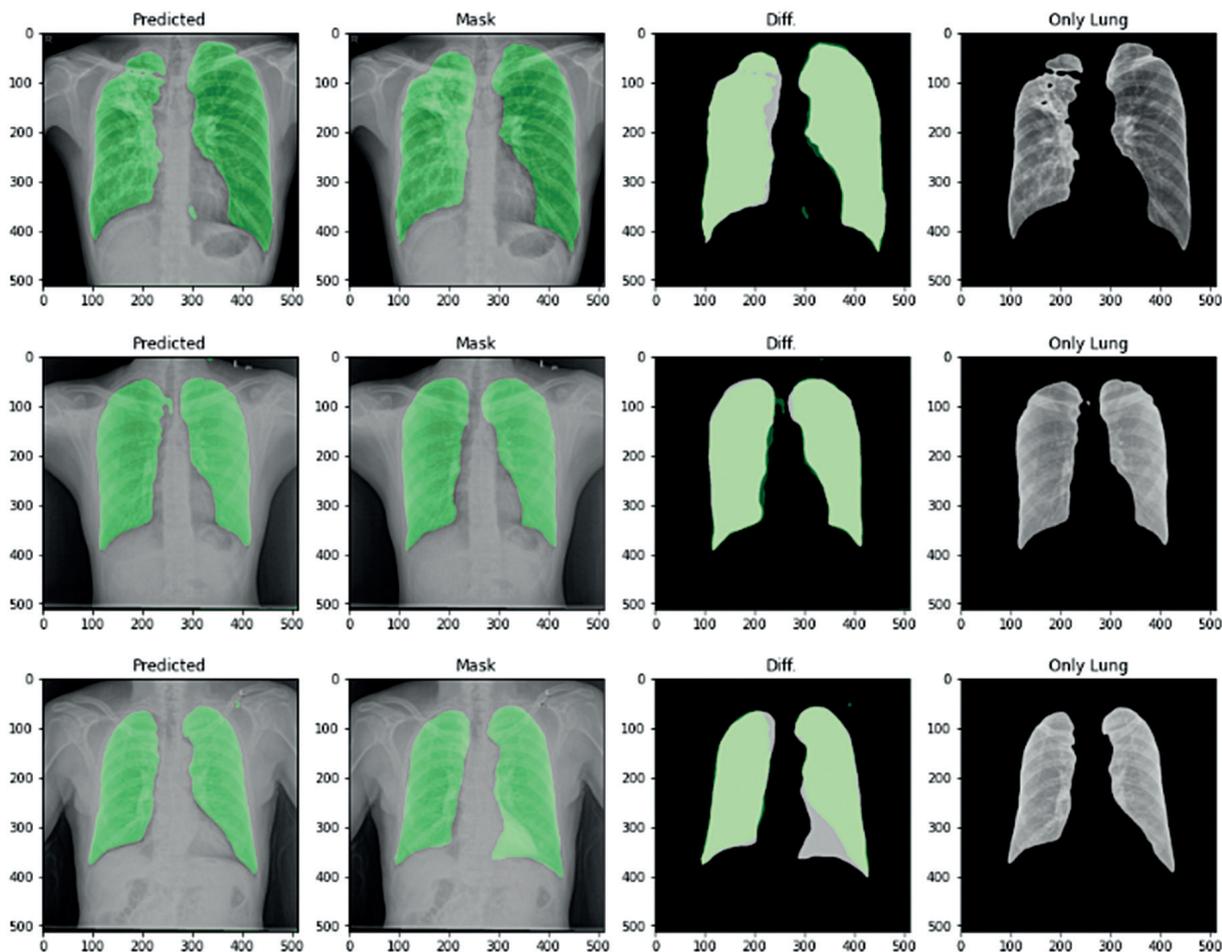


Figure 4. Segmentation of chest X-ray using U-Net

values between 0 and 255. These values are normalised to a scale of 0 to 1. As the image progresses through the encoder, its dimensions decrease after each max pooling operation. A bridge channel links the encoder and decoder, facilitating the transfer of extracted features. The decoder reconstructs the image by interpreting these numerical values, assigning pixel values below 0.5 to 0 and those equal to or greater than 0.5 to 1. Skip connections between the encoder and decoder enable the direct transfer of image features, ensuring minimal information loss during neural node processing. The decoder’s final output comprises binary images with pixel values of 0 (black) and 1 (white), producing black-and-white masks that emphasise regions of interest, such as the lung area in CXR. The U-Net model was trained for 50 epochs per fold, using a learning rate of 0.001 and a dropout rate of 0.25. The batch size was set to 4 images per batch for both training and validation. The cross-entropy loss function, LCE, is expressed as:

$$LCE = y_i \times \log(ai) + (1 - y_i) \log(1 - ai) \tag{1}$$

where i represents the index of the sample (e.g. image), and y_i is the ground truth label, with $y_i = 1$ for the positive class and $1 - y_i = 0$ for the negative class (Figure 4).

Modified U-Net model

For lung image segmentation, the researcher adopted a deep learning approach using U-Net with residual connections. U-Net, developed by Olaf Ronneberger *et al.* for biomedical image segmentation, can be further optimised by replacing standard convolutional units with residual units. Figure 5 shows the architecture of the residual U-net Network. U-Net (ResU-Net) combines the advantages of both U-Net and residual neural networks to achieve better segmentation. results. The ResU-Net architecture is designed to simplify network training while preserving information flow through its skip connections in residual units, preventing degradation. It consists of 3 key components: encoding, bridge, and decoding (Figure 5). Figure 6 illustrates chest Xray images with actual and predicted masks from Res-Unet model

Classification model

The classification model employs a CNN architecture specifically designed for image classification. It processes

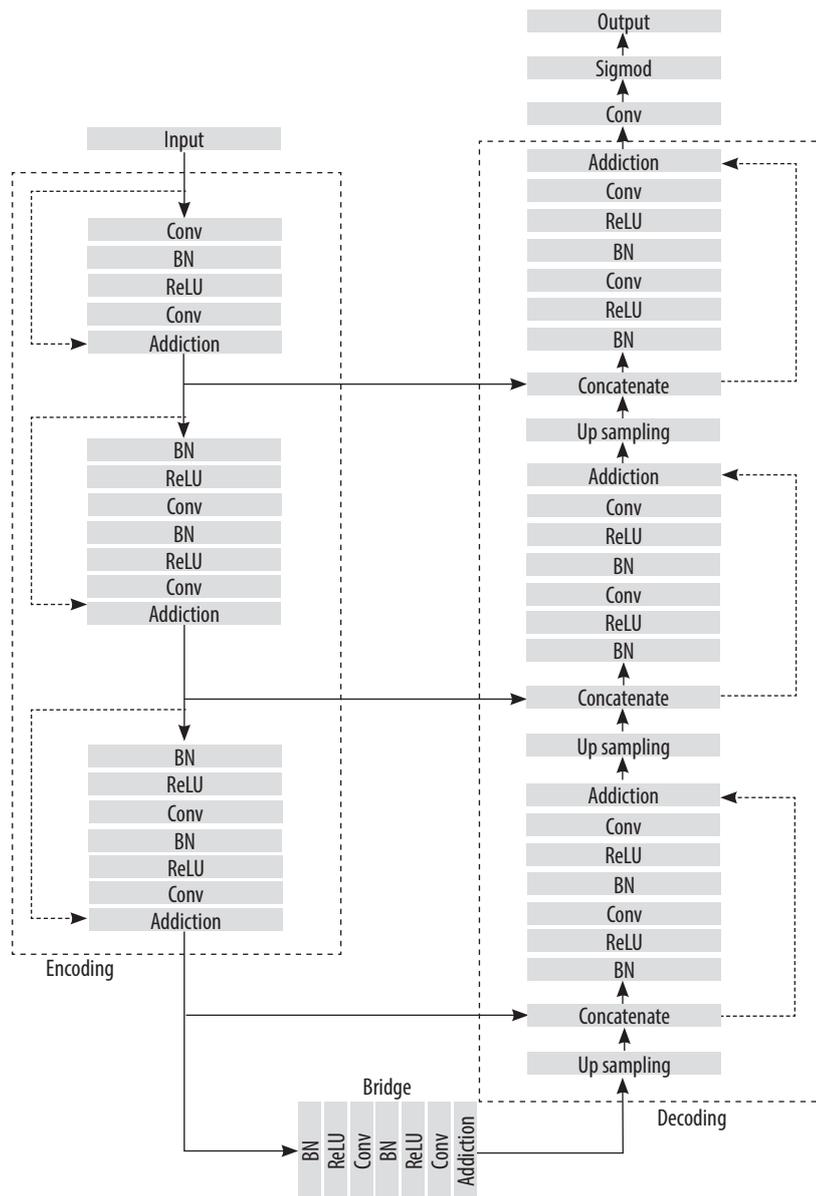


Figure 5. Architecture of Res-UNet

input images through multiple convolutional layers that extract hierarchical features, with each layer followed by max pooling. The convolutional layers detect localised patterns and features, while max pooling reduces spatial dimensions, enhancing computational efficiency and feature abstraction. The features extracted by these layers are then flattened into vector representations and passed through multiple fully connected (dense) layers to complete the final classification. These dense layers enable the model to capture complex features.

The equation for the dense layer is given below:

$$y = \sigma(Wx + b) \tag{2}$$

where x – the input vector or tensor, W – the weight matrix, b – the bias vector, σ – the activation function applied

elementwise, y – the output of the dense (fully connected) layer.

The output layer employs the sigmoid activation function (as shown in Equation 3) to estimate the probability of the input image belonging to a specific class.

$$\sigma(x) = 1/(1 + e^{-x}) \tag{3}$$

Hybrid model

The hybrid model enables the simultaneous segmentation and classification of input images by leveraging the strengths of both approaches. First, the segmentation model identifies and generates masks that highlight specific regions or objects within the image. The segmenta-

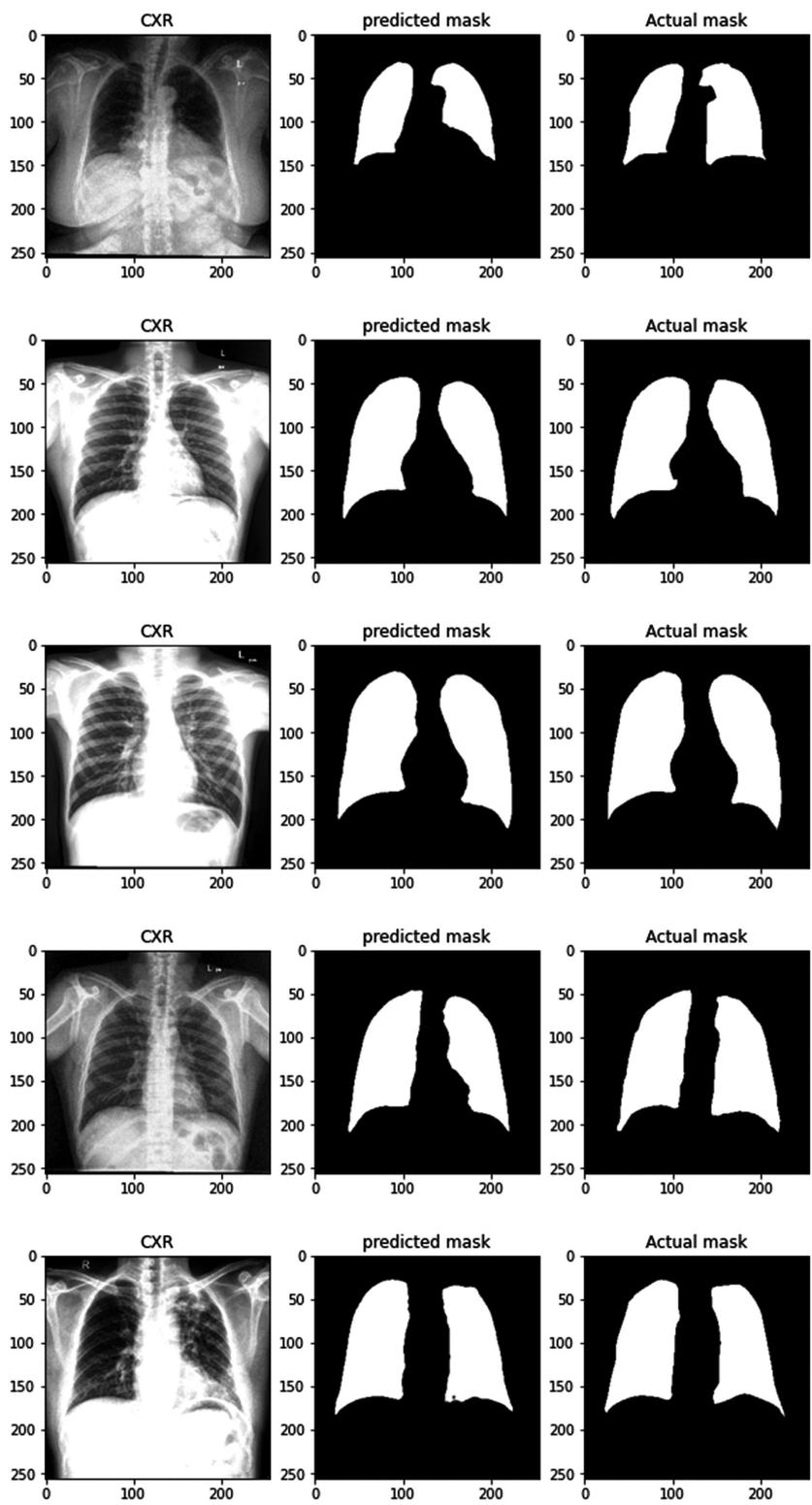


Figure 6. Predicted and actual mask using Res-Unet

tion masks are subsequently input into the classification model, which conducts binary classification using the segmented regions. This approach allows the model to integrate global context with localised details, improving its capability to identify the presence or absence of specific features in the image. The model employs the binary cross-entropy loss function and the Adam optimiser to

ensure efficient learning across both segmentation and classification tasks. The accuracy metric further highlights the model’s effectiveness in executing both tasks with precision. This hybrid approach provides a flexible and reliable solution for complex image analysis, particularly in scenarios requiring continuous segmentation and classification.

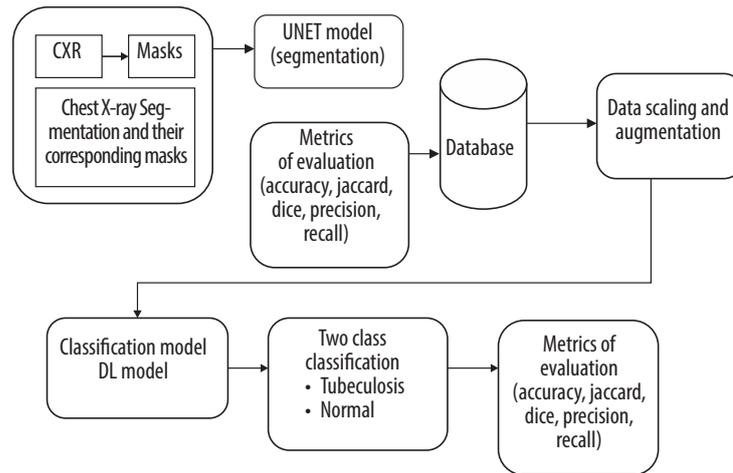


Figure 7. TBNNet-CNN model

TBNNet-CNN model

The proposed TBNNet-CNN model integrates a segmentation model and a classification model into a single hybrid architecture, as illustrated in Figure 7. The process for developing and training this hybrid model using Tensor flow and Keras is detailed in Algorithm 1. The segmentation component is initially defined, consisting of convolutional layers for feature extraction and up-sampling layers to generate segmentation masks. Next, the classification model is defined, incorporating convolutional and dense layers specifically designed for image classification tasks. These 2 components are then integrated to form the hybrid model. The model is trained using labelled data, with training data used for model construction and validation data for performance monitoring.

Input: Input shape = (256, 256, 3)

I. Define the segmentation model

1. Initialize input layer:
inputs \leftarrow Input (input_shape)
2. First convolutional layer:
convolution1 \leftarrow Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
3. Max pooling layer:
pool1 \leftarrow MaxPooling2D(pool_size=(2, 2))(convolution1)
4. Upsampling layer:
up1 \leftarrow UpSampling2D(size=(2, 2))(pool1)
5. Final convolutional layer for mask generation:
decoded \leftarrow Conv2D(3, (3, 3), activation='sigmoid', padding='same')(up1)
6. Define segmentation model:
segmentation_model \leftarrow Model(inputs, decoded, name='segmentation_model')

II. Define the classification model

1. Initialize input layer:
inputs \leftarrow Input(input_shape)

2. First convolutional layer:

convolution1 \leftarrow Conv2D(16, (3, 3), activation='relu')(inputs)

3. Max pooling layer:

pool1 \leftarrow MaxPooling2D(pool_size=(2, 2))(convolution1)

4. Second convolutional layer:

convolution2 \leftarrow Conv2D(32, (3, 3), activation='relu')(pool1)

5. Max pooling layer:

pool2 \leftarrow MaxPooling2D(pool_size=(2, 2))(convolution2)

6. Third convolutional layer:

convolution3 \leftarrow Conv2D(64, (3, 3), activation='relu')(pool2)

7. Max pooling layer:

pool3 \leftarrow MaxPooling2D(pool_size=(2, 2))(convolution3)

8. Fourth convolutional layer:

convolution4 \leftarrow Conv2D(64, (3, 3), activation='relu')(pool3)

9. Max pooling layer:

pool4 \leftarrow MaxPooling2D(pool_size=(2, 2))(convolution4)

10. Fifth convolutional layer:

convolution5 \leftarrow Conv2D(64, (3, 3), activation='relu')(pool4)

11. Max pooling layer:

pool5 \leftarrow MaxPooling2D(pool_size=(2, 2))(convolution5)

12. Flatten the output:

flatten \leftarrow Flatten()(pool5)

13. First dense layer:

dense1 \leftarrow Dense(256, activation='relu')(flatten)

14. Output layer (sigmoid):

output \leftarrow Dense(1, activation='sigmoid')(dense1)

III. Combine the segmentation and classification models

1. Initialise input tensor:

```

input_tensor ← Input(input_shape)
2. Apply segmentation model:
segmentation_output ← segmentation_model(input_tensor)
3. Apply classification model:
classification_output ← classification_model(segmentation_output)
4. Create hybrid model:
hybrid_model ← Model(inputs=input_tensor,
outputs=classification_output, name='hybrid_model')

```

IV. Compile and train the hybrid model

```

1. Compile the hybrid model:
hybrid_model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy','dice_
coeff','jaccard_index'])
2. Define log directory for tensorboard:
logdir = 'logs'
3. Train the model:
hist ← hybrid_model.fit(train, epochs=25, validation_
data=val, callbacks=[tensorboard_callback])

```

Pseudo-code for TB detection system

The pseudocode in the proposed model section outlines the system's workflow, including input processing, segmentation, classification, and the final output. The process begins with an input image of size $256 \times 256 \times 3$, which is first processed by the segmentation model. This involves encoding the image using Conv2D and MaxPooling2D layers, followed by decoding with Upsampling2D and Conv2D layers to generate the segmented output. The segmented images are then passed to the classification model for further analysis. During classification, the images are processed through several Conv2D and MaxPooling2D layers, followed by a Flatten layer and 2 Dense layers to produce the final classification. This hybrid approach seamlessly integrates segmentation and classification, offering a robust solution.

1. Specify the input image shape:
 - Set the input shape to $256 \times 256 \times 256$ (height, width, and RGB channels).
2. Construct the segmentation model:
 - a) Input layer: Establish the input layer for the model.
 - b) Encoder: Employ a series of Conv2D and MaxPooling2D layers to extract hierarchical features.
 - c) Decoder: Use UpSampling2D and Conv2D layers to recover spatial dimensions and refine the segmented output.
 - d) Output layer: Use a sigmoid activation function for pixel-wise binary classification.
3. Build the classification model:
 - a) Input layer: Define the input layer for the classification model.
 - b) Convolutional layers: Use a sequence of Conv2D and MaxPooling2D layers to extract important features.
 - c) Flatten layer: Flatten the feature maps into a 1D vector.

- d) Dense layers: Use fully connected layers with Relu activation for complex feature interactions.
 - e) Output layer: Use a sigmoid activation function for binary classification.
4. Integrate the models:
 - a) Use the output from the segmentation model as input for the classification model.
 - b) Define the hybrid model by combining the segmentation and classification components into a unified framework.
 5. Compile and summarise the hybrid model: Compile the hybrid model with a suitable optimiser and loss function, and generate a summary of its architecture.

Results

This section offers a detailed overview of the evaluation metrics employed to measure the performance of the TBNet-CNN model. These metrics are crucial for determining how effectively the model identifies patterns and makes predictions, serving as key indicators of its overall performance. Accuracy is a key metric for classification models, representing the ratio of correctly predicted instances to the total number of instances, providing an overall assessment of prediction reliability. Recall measures the model's ability to accurately identify positive instances, which is especially important when detecting all positive cases is critical. The F1-score, which combines precision and recall into a harmonic mean, provides a balanced evaluation of the model's performance, especially when both metrics are equally important or when dealing with imbalanced classes. Additionally, the confusion matrix offers a comprehensive breakdown of the model's predictions compared to the actual ground truth labels, consisting of 4 key elements: false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). This enables a deeper analysis of the model's predictive accuracy and error patterns. The equations for f1_score, recall, and precision are given below:

$$f1_score = 2 * (Precision * Recall) / (Precision + Recall) \quad (4)$$

$$Recall = True\ Positive / (True\ Positive + False\ Negative) \quad (5)$$

$$Precision = True\ Positive / (True\ Positive + False\ Positive) \quad (6)$$

$$Accuracy = True\ Positive + True\ Negative / (True\ Positive + True\ Negative + False\ Positive + False\ Negative) \quad (7)$$

Loss curves and accuracy curves for pre-trained and TBNet-net models are shown in Figure 8.

Tb-CNN net accuracy seemed to be higher than other pre-trained models, and the loss is also low compared to other models.

U-Net and Res-UNet loss and accuracy curves are shown in Figure 9. From this figure we can see that Res-UNet performs better than U-Net.

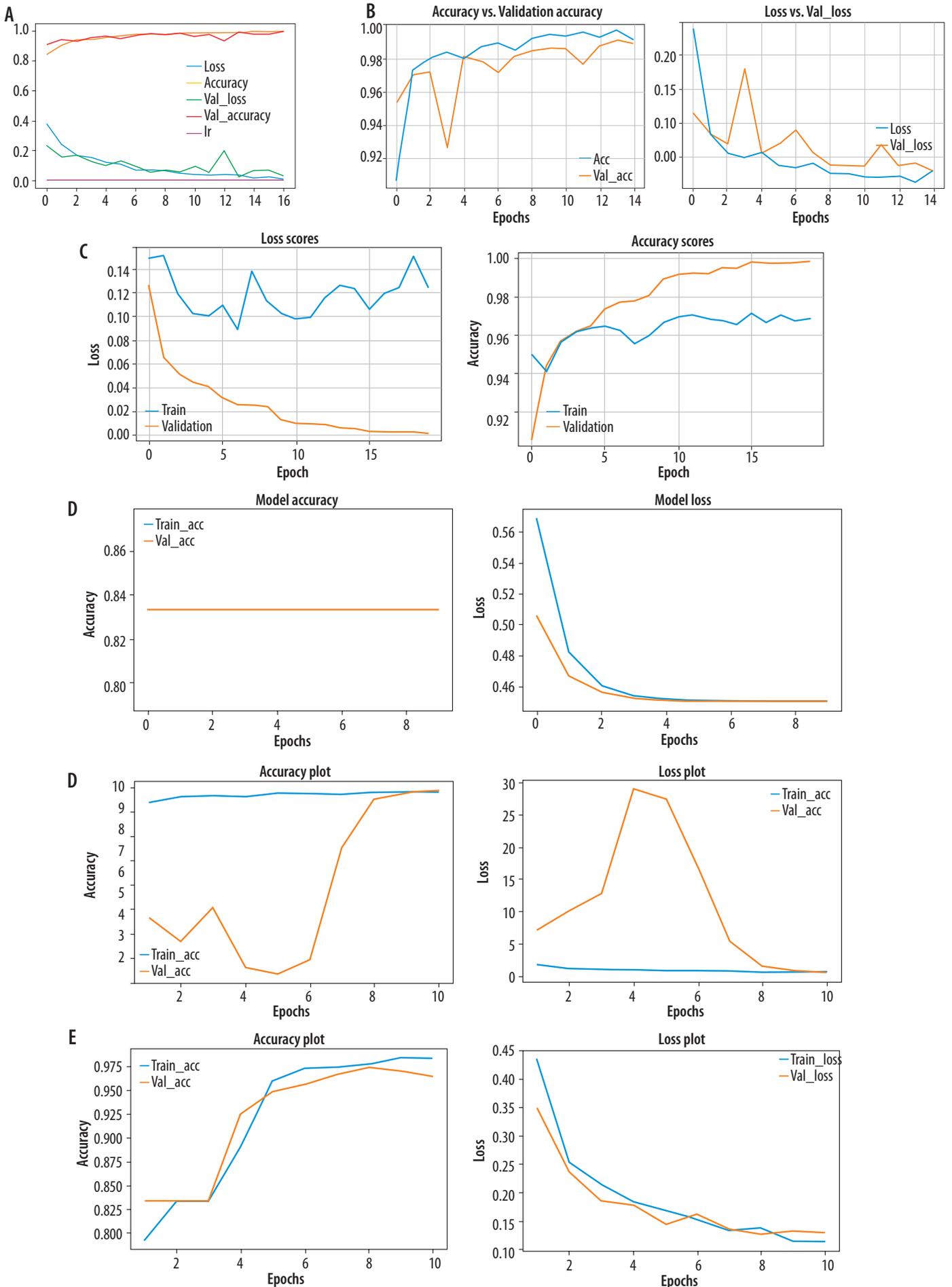


Figure 8. Loss and accuracy curves of different CNN models. A) Loss and accuracy curves for TBCNN-net. B) Loss and accuracy curves for VGG16. C) Loss and accuracy curves for DenseNet121. D) Loss and accuracy curves for VGG19. E) Loss and accuracy for ResNet50. F) Loss and accuracy curves for Xception

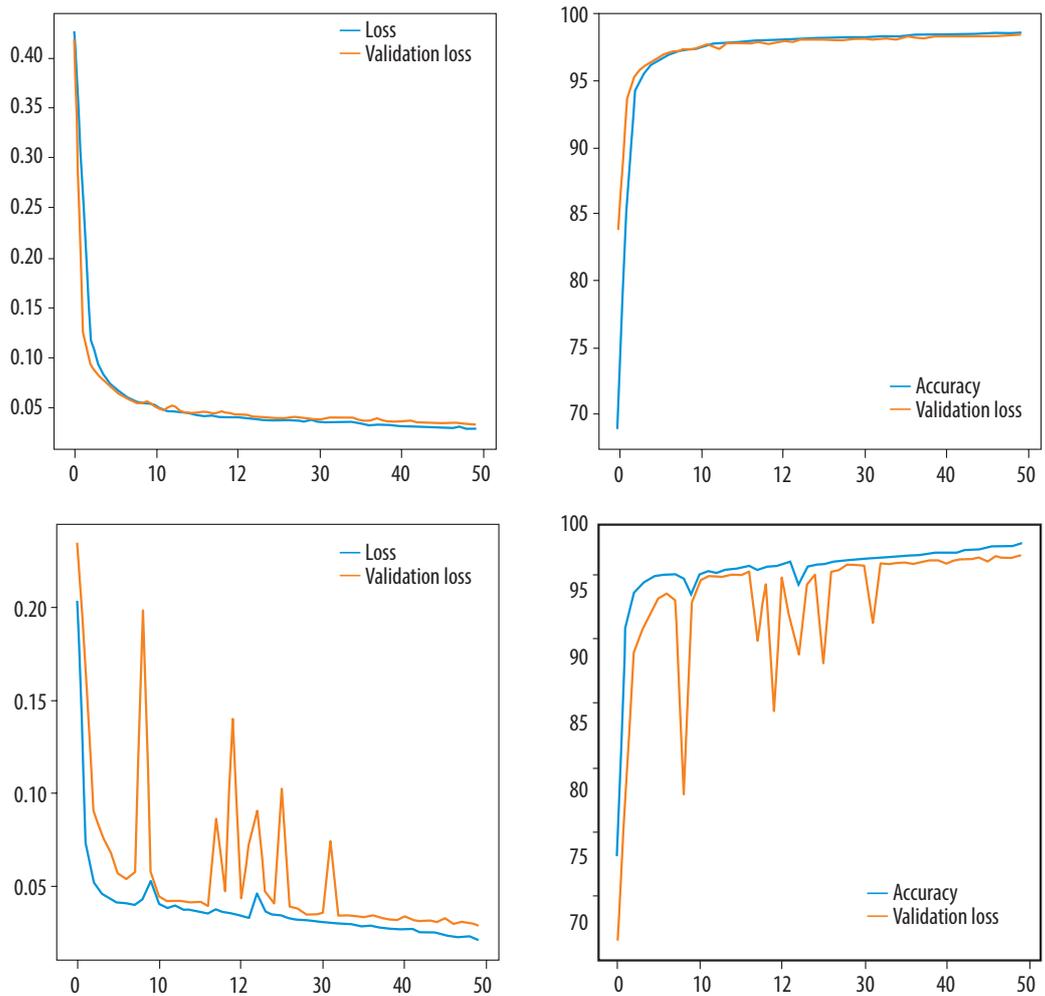


Figure 9. Loss and accuracy curves of U-Net and ResUNet

Table 1. ResUNet and U-Net model performance metrics

Model name	Accuracy	Recall	f1_score	Precision	Dice_coeff
U-Net	97.77	98.08	97.8	97.52	96.53
Res-Unet	98.18	98.40	91.97	98.39	96.33

Table 2. Metrics of performance between TBCNN-net and other pre-trained models

Model name	Accuracy	Precision	Recall	f1_score
VGG16	81.35	55	52	50
VGG19	80.59	51	51	49
DenseNet121	80.24	53	51	50
Xception	79.65	49	50	48
InceptionV3	79.17	49	50	48
ResNet50	83.35	42	50	45
TBCNN-net	99.45	99.03	99.29	99.29

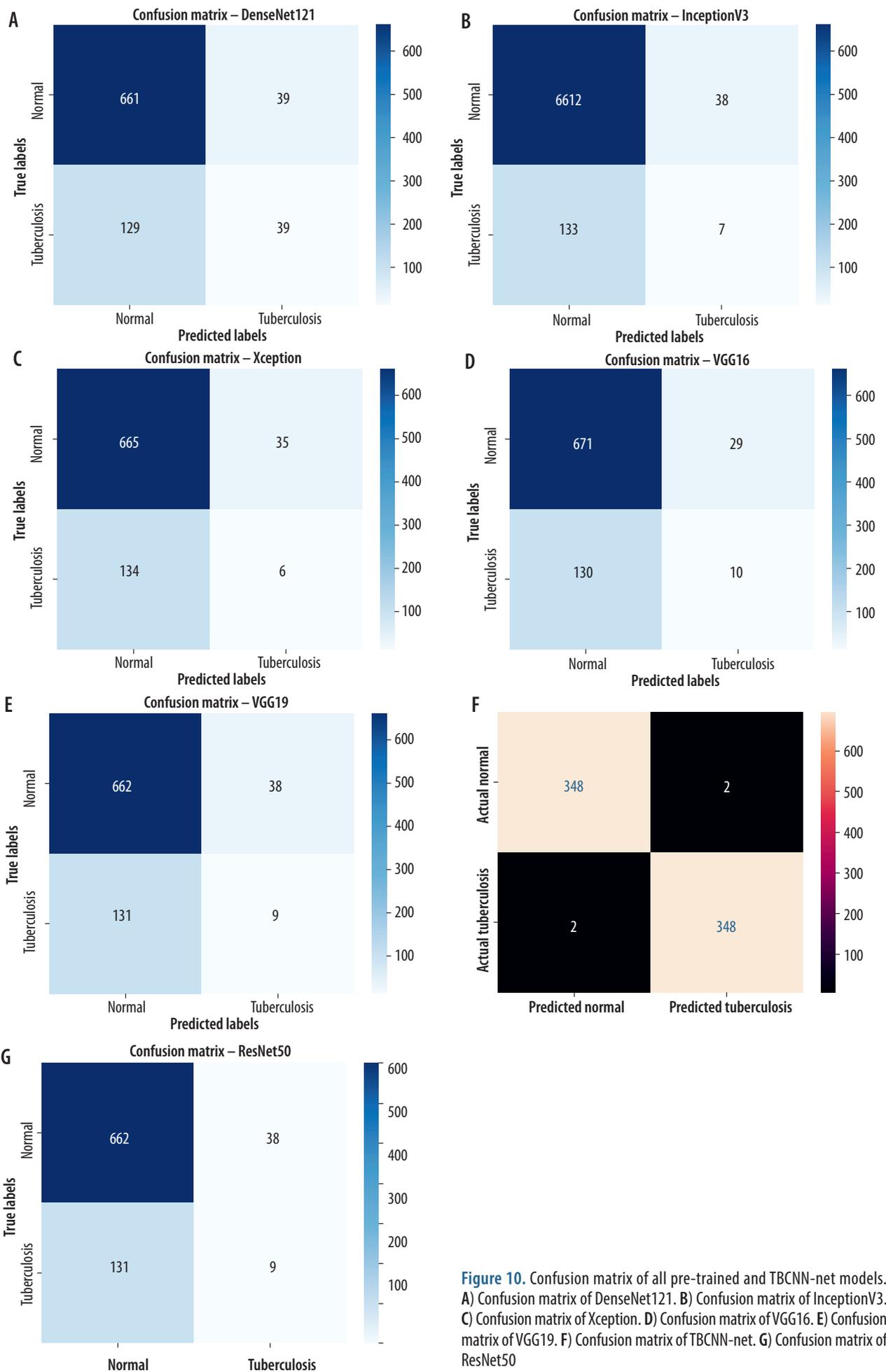


Figure 10. Confusion matrix of all pre-trained and TBCNN-net models. A) Confusion matrix of DenseNet121. B) Confusion matrix of InceptionV3. C) Confusion matrix of Xception. D) Confusion matrix of VGG16. E) Confusion matrix of VGG19. F) Confusion matrix of TBCNN-net. G) Confusion matrix of ResNet50

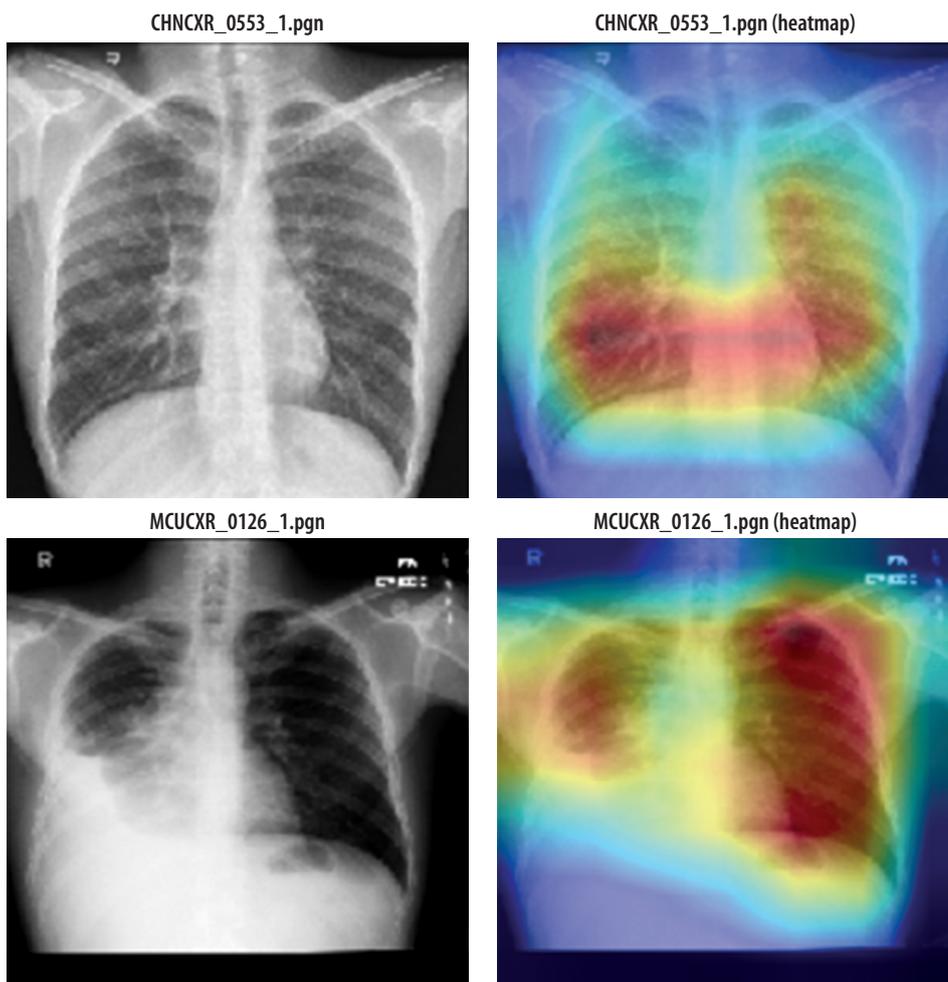


Figure 11. Heatmap for segmented images using gradCAM

Table 1 shows the accuracy of ResUnet is 98.18, recall is 98.40, precision is 98.39 which is quite more as compared to that of Unet. Only f1_score and dice coefficient is little less. But on an average the performance of ResUnet is quite good as compared with Unet Model.

Discussion

Confusion matrices for TBCNN-net and other pretrained models are shown in Figure 10.

As compared to other pre-trained models, TBCNN-net performed with 348 predicted and actual TB cases, followed by DenseNet121 with 11, VGG16 with 10 cases, VGG19 with 9, InceptionV3 with 7, and Xception with 6 cases.

Table 2 shows that the TBCNN-net has the greatest accuracy.

Figure 11 shows heatmap for segmented images indicating presence of disease infected regions.

Conclusions

This study introduces a novel deep learning-based CNN model, TBCNN-net, designed to detect TB from

X-ray images. The model employs a two-stage architecture, combining a segmentation module with a classification module to enhance prediction accuracy. Comprehensive evaluations highlight the system's reliability and practicality, outperforming previous methods in terms of accuracy of 99.45% and recall of 99.29%. Additionally, its superior Jaccard index of 96.05% and Dice coefficient of 96.33% further validate its performance using residual U-net architecture. When tested on unseen images, TBCNN-net demonstrated robustness and strong generalisation across diverse datasets, maintaining high accuracy. Future research will focus on integrating multimodal data, including genetic information and clinical metadata, to create more advanced diagnostic tools and deepen insights into it. TB detection values further validate its performance.

Disclosures

1. Institutional review board statement: Not applicable.
2. Assistance with the article: None.
3. Financial support and sponsorship: None.
4. Conflicts of interest: None.

References

- Iqbal A, Usman M, Ahmed Z. Tuberculosis chest X-ray detection using CNN-based hybrid segmentation and classification approach. *Biomedical Signal Processing and Control* 2023; 84. DOI: <https://doi.org/10.1016/j.bspc.2023.104667>.
- Nafisah SL, Muhammad G. Tuberculosis detection in chest radiograph using convolution neural architecture and explainable artificial intelligence. *Neural Computing and Applications* 2022; 36: 111-131.
- Sharma V, et al. Deep learning models for tuberculosis detection and infected region visualization in chest X-ray images. *Intelligent Medicine* 2024; 4: 104-113.
- Rahman T, et al. Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. *IEEE Access* 2020. DOI: 10.1109/ACCESS.2020.3031384
- Rajaraman S, Folio LR, Dimperio J, Alderson PO, Antani SK. Improved semantic segmentation of tuberculosis – consistent findings in chest X-rays using augmented training of modality-specific U-Net models with weak localizations. *Diagnostics (Basel)* 2021; 11: 616. DOI: <https://doi.org/10.3390/diagnostics11040616>.
- Ignatius JLP. Histogram matched chest X-rays based tuberculosis detection using CNN. *Comput Sci Engineering* 2021. DOI: 10.32604/csse.2023.025195.
- Chavan M, Varadarajan V, Gite S, Kotecha K. Deep neural network for lung image segmentation on chest X-ray. *Technologies* 2022; 10: 105. DOI: <https://doi.org/10.3390/technologies10050105>.
- Sineglazav V. A novel algorithm has been developed to refine segmented masks, enhancing final accuracy, alongside an efficient fuzzy inference system for more weighted activity assessment. *Comput Biol Med* 2022; 147: 1-18.
- Dao L, Lyn NQ. A Comprehensive study on medical image segmentation using deep neural networks. *Int J Adv Computer Sci Appl* 2023; 14: 167-184.
- Showkatie E. Deep learning-based automatic detection of tuberculosis disease in chest X-ray images. *Pol J Radiol* 2022; 87: e118-e124. DOI: 10.5114/pjr.2022.113435.
- Wang CH, Chang W, Lee MR, Tay J, Wu CY, Wu MC, et al. Deep learning-based diagnosis of pulmonary tuberculosis on chest X-ray in the emergency department: a retrospective study. *J Imaging Inform Med* 2024; 37: 589-600.
- Dipali H, Patil DH, Gadekar A. Tuberculosis disease detection through CXR images based on deep neural network approach. *Int J Recent Innovation Trends Comp Commun* 2023; 11(7s): 96-106.
- Guia SS, Laouid A, Kara M, Hammoudeh M. Tuberculosis detection using chest X-ray image classification by deep learning. *Res Square* 2023. DOI: <https://doi.org/10.21203/rs.3.rs-2509525/v2>.
- Verma G, Kumar A, Dixit S. Early detection of tuberculosis using hybrid feature descriptors and deep learning network. *Pol J Radiol* 2023; 88: e445-e454. DOI: 10.5114/pjr.2023.131732.
- Le TM, Nguyen-Tat BT, Ngo VM. Automated evaluation of tuberculosis using deep neural networks. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 2022; 9: 1-14. DOI: 10.4108/eetinis.v8i30.478.