Tuberculosis Detection from Chest X-Ray Images using Deep Learning

Overview

Tuberculosis (TB) continues to be a major public health concern worldwide, particularly in regions with limited medical resources. This project presents a deep learning-based approach to automatically classify chest X-ray images as either *Normal* or *Tuberculosis (TB)*. By applying transfer learning techniques using pre-trained convolutional neural networks (CNNs), the project delivers a reliable classification system that can assist healthcare professionals. Models such as **VGG16**, **ResNet50**, and **EfficientNetB0** were trained and evaluated using medical imaging data, with performance assessed through metrics like accuracy, precision, recall, F1-score, and ROC-AUC.The project also sets the foundation for a real-time interactive application that can be deployed using Streamlit.

Business Use Cases

Early Detection of TB
The system is designed to act as a diagnostic aid for radiologists and clinicians, improving speed and accuracy in identifying potential TB cases from chest X-rays.

Screening in Remote Areas
In regions where medical specialists are not readily available, this automated solution can support large-scale screening efforts and reduce the burden on existing healthcare infrastructure.

Reducing Diagnostic Errors
By providing consistent model predictions based on learned patterns, the system helps reduce variability and potential errors in diagnosis, serving as a second opinion.

Medical Research & Analysis
The platform can be used to analyze the prevalence of TB across datasets, study patterns in diagnosis, and evaluate the effectiveness of deep learning methods in radiographic analysis.

Project Workflow

1. Dataset

The dataset used is publicly available and consists of **3,008** chest X-ray images divided into two categories:

- **TB-Positive Images**: 2,494
- **Normal Images**: 514

The images are structured into separate directories for **training**, **validation**, and **testing** to ensure robust evaluation and model generalization.

2. Data Preprocessing & Augmentation

To improve model performance and generalization, several preprocessing steps were applied:

- Image rescaling to normalize pixel values between 0 and 1
- Resizing all images to 224x224 pixels
- Augmentation techniques like rotation, zoom, shear, and horizontal flipping
- Dataset class balancing through controlled splitting

These steps were implemented using TensorFlow's ImageDataGenerator.

3. Model Development

Three popular CNN architectures were used via transfer learning:

- **VGG16**
- **ResNet50**
- **EfficientNetB0**

Each model follows a similar architecture:

- Base layers from pre-trained ImageNet weights
- A GlobalAveragePooling2D layer
- A fully connected Dense layer with ReLU activation

- Dropout for regularization
- A final output layer with sigmoid activation for binary classification

Training was conducted using the Adam optimizer and binary cross-entropy loss for 10 epochs on the training dataset.

4. Evaluation Metrics

The performance of each model was evaluated using:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**
- **ROC-AUC Score**
- Confusion matrices and ROC curves were plotted for visualization and comparison.

Results and Model Comparison

The three models were evaluated on the same test dataset. Their comparative performance is summarized below:

Model	Accuracy	Precision (TB)	Recall (TB)	F1-Score (TB)	AUC Score
VGG16	86%	0.85	1.00	0.92	0.9778
ResNet50	83%	0.83	1.00	0.91	0.9511
EfficientNetB0	83%	0.83	1.00	0.91	0.8197

ROC-AUC Visualization: The ROC-AUC scores highlight that VGG16 outperformed the other two models in terms of generalization and separation capability:

Insights and Observations:

- VGG16 achieved the highest ROC-AUC score (0.9778) and overall balanced performance.
- ResNet50 and EfficientNetB0 achieved similar performance on TB detection.
- All models suffered from class imbalance, with TB images far outnumbering Normal images.

Conclusion

This project demonstrates the power of transfer learning in solving medical imaging problems. With consistent performance across all three models, the deep learning pipeline developed here can be used as a clinical decision-support system or a screening tool in resource-constrained environments. It also lays the groundwork for real-world applications through future web deployment and visualization tools.