DEPARTMENT OF COMPUTER SCIENCE

ISLAMIC UNIVERSITY OF SCIENCE AND TECHNOLOGY,
AWANTIPORA PULWAMA

TRANSFER LEARNING FOR RESPIRATORY DISEASE CLASSIFICATION IN CHEST X-RAYS

Presented By:

Shahid-Ul-Islam (MCA-23-07)
Sabreena Jan (MCA-23-20)

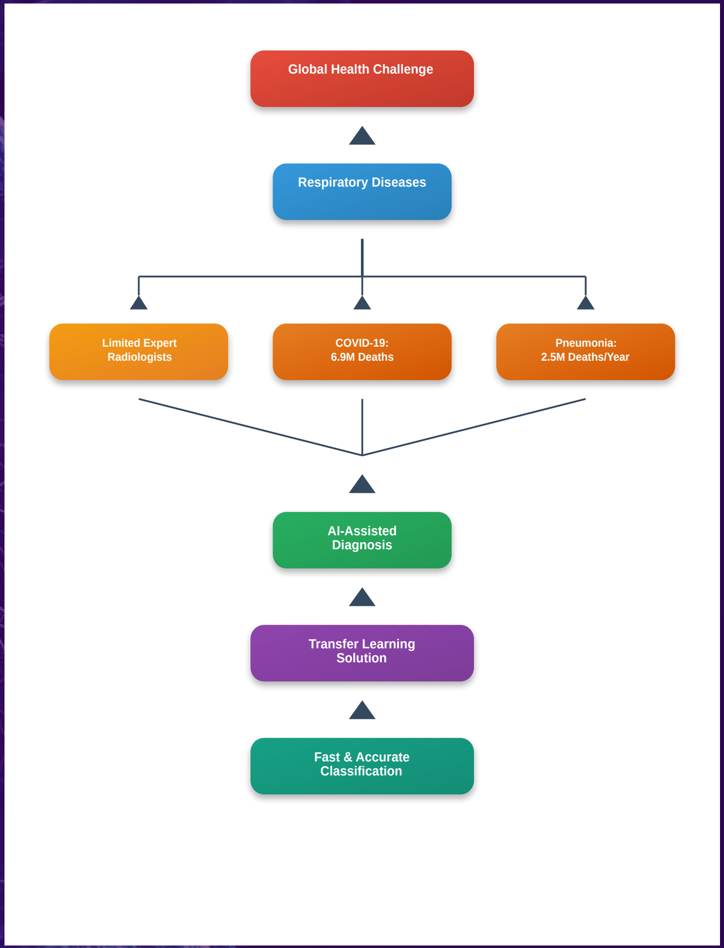
Supervisor:

Dr. Syed Tanzeel Rabani

AGENDA

INTRODUCTION

- Respiratory diseases like pneumonia, and COVID-19 are major global health concerns.
- Chest X-rays are the most widely used imaging technique for diagnosis.
- Manual interpretation is slow, subjective, and limited by the shortage of expert radiologists.
- Deep learning models offer high diagnostic accuracy but need large labeled datasets.
- Transfer learning enables effective training on limited medical data by leveraging pretrained models.
- This project applies transfer learning to classify respiratory diseases from chest X-ray images efficiently.



Key Impact: Early detection can reduce mortality by 40-60% (WHO, 2023; CDC, 2023; Houck et al., The Lancet, 2004).

PROBLEM STATEMENT

A CLOSER LOOK AT THE PROBLEM DRIVING THE RESEARCH

IMPROVE CHEST X-ray QUALITY & STANDARDIZATION

Address data limitations, class imbalance, and visual similarity through preprocessing, augmentation, and class imbalance handling.

01

DEVELOP & COMPARE DEEP LEARNING MODELS

Build robust classifiers using transfer learning (VGG16, ViT B/16) and evaluate against models trained from scratch.

02

ENSURE CLINICAL GRADE PERFORMANCE

Rigorously evaluate models using relevant metrics, while optimizing for real-world constraints like inference time and generalizability.

03

OBJECTIVES

01
Preprocess the Dataset
- Apply image normalization and standardization
- Enhance image quality
- Use data augmentation to improve generalization

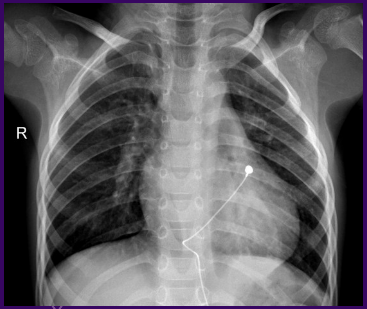
02
Develop Deep Learning Models
- Implement classification models using transfer learning (VGG16 & ViT B/16)
- Fine-tune pretrained architectures for 4-class chest X-ray classification

03
Evaluate Model Performance
- Use classification metrics: accuracy, F1-score, ROC, sensitivity, specificity
- Measure computational efficiency: model size, inference time

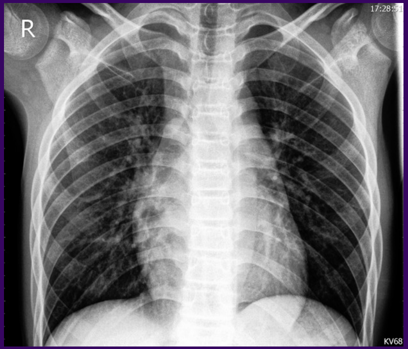
04
Compare Transfer Learning vs. Training from Scratch
- Train equivalent models without pretrained weights
- Assess performance under different dataset conditions and architectures

DATASET DETAILS

VIRAL PNEUMONIA



BACTERIAL PNEUMONIA



NORMAL

COVID-19

DATASET OVERVIEW

- Source: Kaggle Pneumonia & COVID-19 Chest X-ray Dataset
- Total Images: 6,600 chest X-ray images
- Image Format: JPEG, PNG (256×256 to 1024×1024)
  Color: Grayscale medical imaging

DATA QUALITY CHALLANGES

- Varied image quality from different hospitals
- Inconsistent positioning and exposure levels
- Label noise in some COVID-19 samples
- Equipment differences across institutions

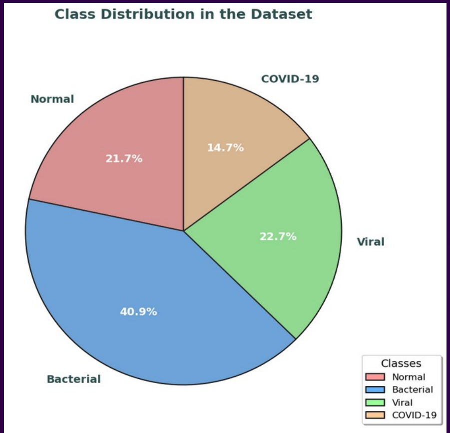
CLASS DISTRIBUTION

- COVID19 : 980
- BACTERIAL : 2727
- VIRAL : 1512
- NORMAL : 1443

Normal: 21.66% | Bacterial: 40.93%
Viral: 22.69% | COVID-19: 14.71%

CLASS IMBALANCE HANDLING

IMBALANCE STATISTICS




```
class_counts = {
        'Normal': 1443,
        'Bacterial': 2727,
        'Viral': 1512,
        'COVID-19': 980
        }
imbalance_severity =
max(class_counts.values()) /
min(class_counts.values())
# Result: 8.28 (High imbalance)
```

WHY WEIGHTED CROSS-ENTROPY LOSS

- Penalizes misclassification of minority classes more heavily
- Mathematically sound approach for imbalanced datasets
- Preserves original data distribution
- Computationally efficient compared to sampling methods

WEIGHT CALCULATION METHOD

Inverse Frequency Weighting

- Calculate Total Samples and Classes
- Compute Class Weights
- Assign Higher Weights to Minority Classes

LOSS FUNCTION IMPLEMENTATION

```
criterion =
nn.CrossEntropyLoss(
    weight=torch.tensor([1.68,
0.61, 0.98, 5.02])
)
```

ALTERNATE METHODS CONSIDERED

- SMOTE: Risk of overfitting in medical images
- Oversampling: Increases training time significantly
- Focal Loss: Complex hyperparameter tuning
- ***Weighted CE: Simple, effective, interpretable***

DATA PREPROCESSING & AUGMENTATION

PREPROCESSING PIPELINE

Smart Preprocessing for Deep Vision: Optimized for CNNs and Transformers.

01

Common Steps

- Load image in grayscale
- Apply CLAHE (Contrast Limited Adaptive Histogram Equalization) to enhance local contrast
- Normalize pixel values to range [0, 1] by dividing by 255
- Convert grayscale to RGB
- Convert NumPy array to PIL Image for compatibility with PyTorch transform

02

VGG16-Specific Transforms

- Resize: 224×224 (ImageNet standard)
- Normalization: ImageNet mean/std
- Data type: Float32

03

ViT B/16 -Specific Transforms

- Resize: 384×384 (ViT-Base optimal)
- Center crop: 224×224 patches
- Patch tokenization: 16×16 patches

DATA PREPROCESSING & AUGMENTATION

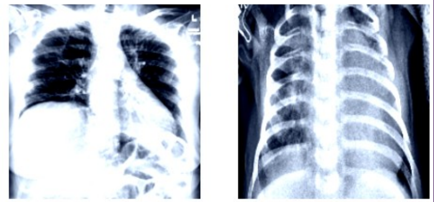
AUGMENTATION STRATEGIES

Optimizing Training Inputs: Augmentations Aligned with Architecture.

01

VGG16 AUGMENTATION

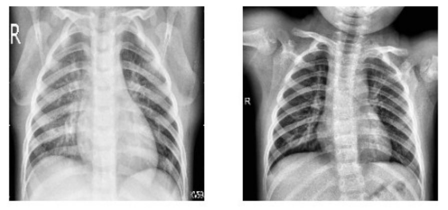
- Random Rotation
- Random Horizontal Flip
- Random Resized Crop
- Color Jitter (Brightness & Contrast)
- Random Affine (Rotation + Translation)



02

ViT B/16 AUGMENTATION

- Resize
- Center Crop
- Color Jitter (Hue & Saturation)
- Random Erasing



REASON FOR DIFFERENT PIPELINES

VGG16: Uses aggressive augmentation (rotation, flips, color jitter, affine transforms) to encourage robust feature learning and improve generalization on limited data.

ViT B/16: Applies minimal augmentation (resize, center crop, slight color jitter, random erasing) to preserve spatial and structural relationships critical for Vision Transformers, which rely on patch-wise image understanding.

Medical Imaging (Chest X-rays): Employs conservative preprocessing and augmentation to maintain diagnostic features without distorting subtle medical details essential for accurate classification.

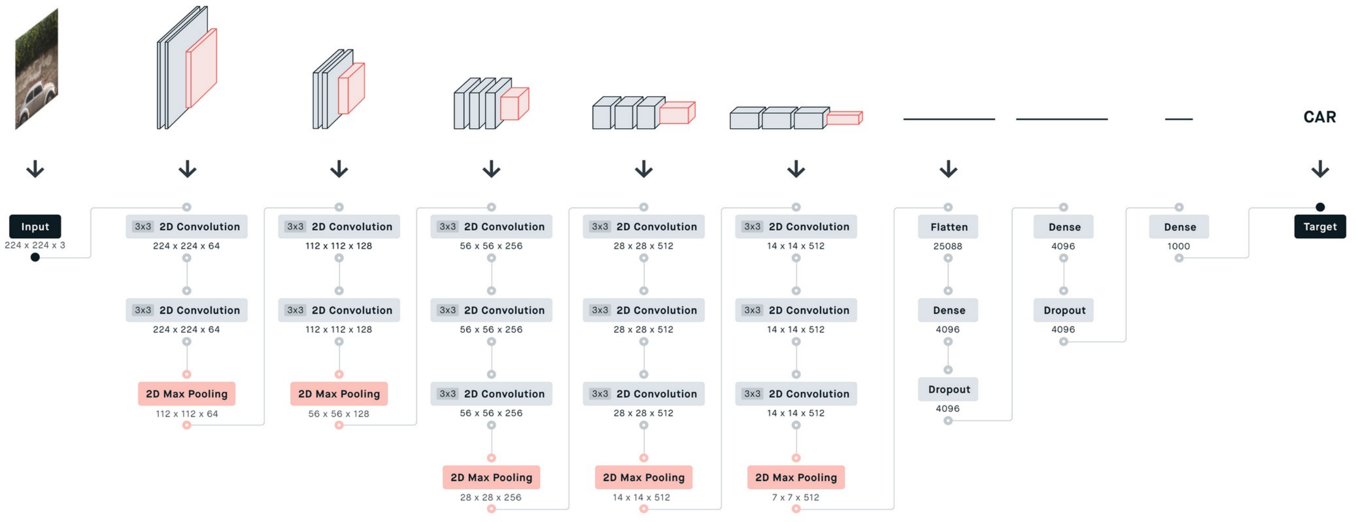
MODEL ARCHITECTURE - VGG16

Architecture Overview

- 16 layers deep convolutional network
- Pretrained on ImageNet (1.2M images)
- Transfer learning via classifier replacement

Implementation Details

- Uses pretrained VGG16 backbone (ImageNet weights)
- Freezes convolutional layers to avoid retraining feature extractor
- Replaces original classifier with custom 3-layer fully connected head
- Includes ReLU activations and Dropout for regularization
- Outputs logits for 4 classes (COVID-19, viral pneumonia, bacterial pneumonia, normal)



Transfer Learning Strategy

- Phase 1: Freeze backbone, train classifier head
- Phase 2: Unfreeze last conv block.
- Phase 3: Fine-tune entire network.

MODEL ARCHITECTURE – ViT B/16

Architecture Overview

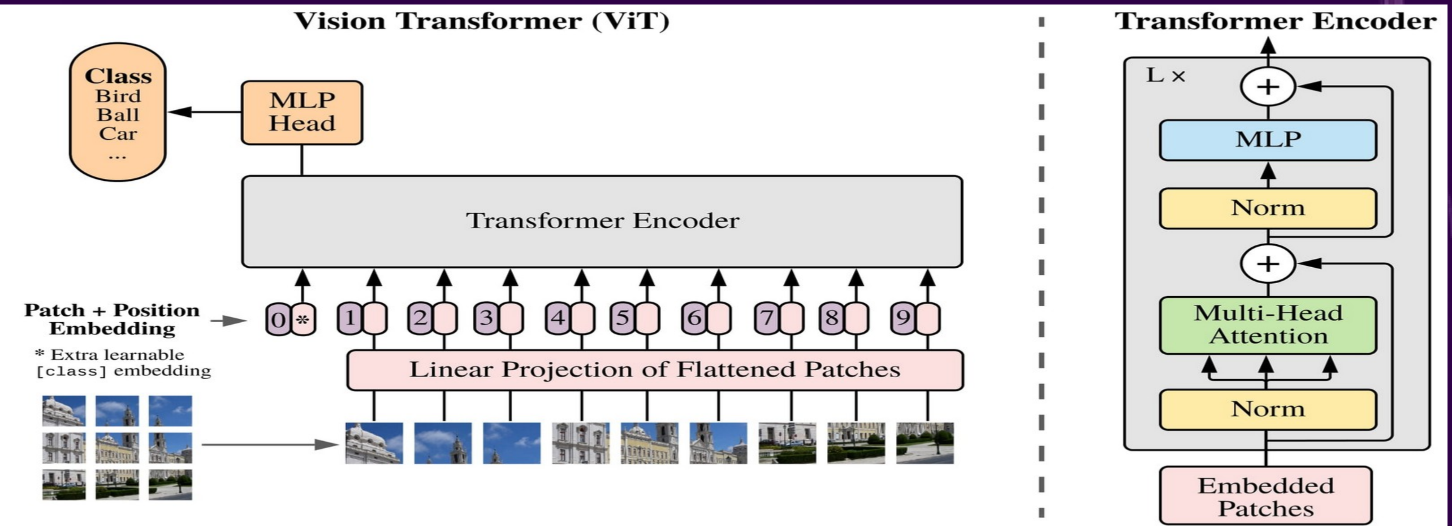
- Attention-based image processing
- Pretrained on ImageNet-21k (14M images)
- 12 transformer blocks, 768 hidden dimensions

Implementation Details

- Uses pretrained ViT-Base (patch16_224) model from timm
- Removes original classification head (num_classes=0) to extract features
- Adds custom classification head:
- LayerNorm → Dropout → Linear (768 → 256) → GELU → Dropout → Linear (256 → num_classes)
- Forward pass: extracts features via ViT backbone, then classifies with custom head
- Designed for 4-class classification (COVID-19, viral pneumonia, bacterial pneumonia, normal)

Patch Embedding

- 16×16 patches → 768-dim embeddings
- Learnable positional encoding
- Class token for classification

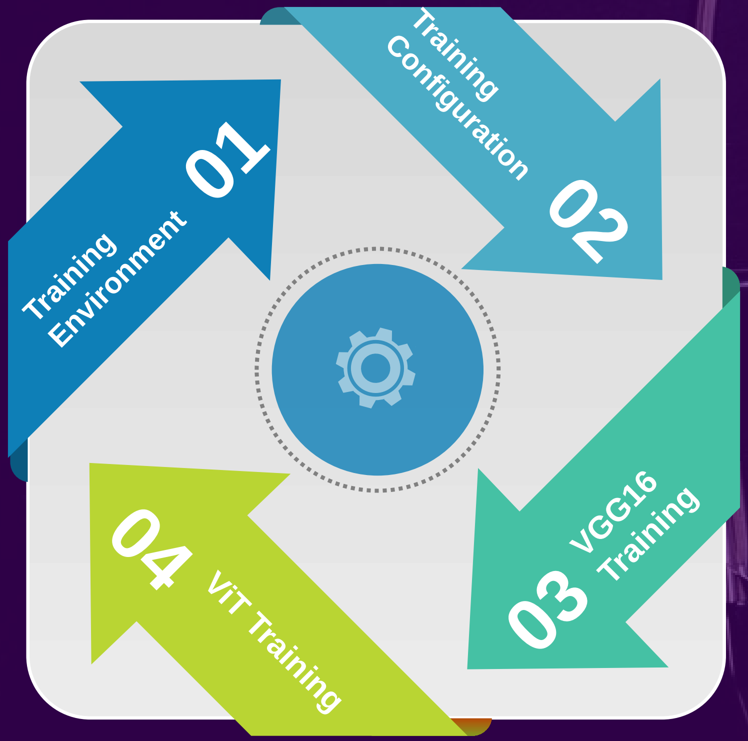


Self Attention Mechanism

- Each token looks at all other tokens in the input to decide what to focus on.
- It learns contextual relationships by weighing the importance of other tokens dynamically.

TRAINING PIPELINE

- Platform: Kaggle
- GPU: NVIDIA Tesla P100 (16GB VRAM)
- RAM: 25GB system memory
- Storage: 100GB persistent disk



Parameter	VGG16	ViT-Base
Batch Size	32	16
Initial LR	1e-3	5e-4
Epochs	15	15
Dropout	0.5	0.1
Weight Decay	1e-4	1e-2

Optimizer:
- Used AdamW (preferred for transformers)
- Learning rate: 5e-4
- Weight decay: 0.01 (to prevent overfitting)

Learning Rate Scheduler:
- Used ReduceLROnPlateau to reduce LR on validation loss plateau
- Reduces LR by a factor of 0.5
- Waits for 3 epochs (patience)
- Minimum LR capped at 1e-6

Optimizer:
- Used Adam optimizer
- Learning rate: 1e-3
- Weight decay: 1e-4 (for regularization)
- β values: (0.9, 0.999) for momentum

Learning Rate Scheduler:
- Used ReduceLROnPlateau to reduce LR on validation loss plateau
- Reduces LR by a factor of 0.5
- Waits for 6 epochs (patience)
- Minimum LR capped at 1e-6

EVALUATION METRICS

Primary Metrics

ACCURACY

- accuracy = (TP + TN) / (TP + TN + FP + FN)
- Overall performance indicator
- Target: ≥90% for clinical acceptance

F1 SCORE (Macro Avg.)

- f1_macro = mean([f1_class_i for all classes])
- precision = TP / (TP + FP)
- recall = TP / (TP + FN)
- f1 = 2 * (precision × recall) / (precision + recall)
- Balanced measure considering precision & recall
- Critical for imbalanced datasets

AREA UNDER CURVE

- Threshold-independent metric
- Multiclass: One-vs-Rest approach
- Clinical significance: Ranking capability

Clinical Metrics

SENSITIVITY (Recall)

- sensitivity = TP / (TP + FN)
- Critical for medical diagnosis
- COVID-19 target: ≥95% (minimize false negatives)

SPECIFICITY

- specificity = TN / (TN + FP)
- Minimize false positives
- Prevents unnecessary treatments

INFERENCE SPEED

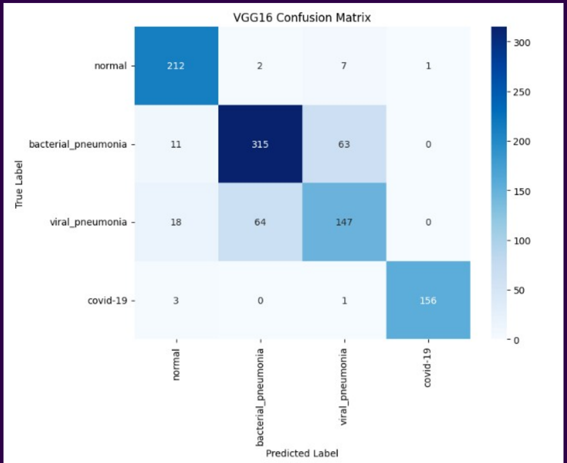
- Target: <100ms per image
- Measured on: CPU (Intel i7) and GPU
- Batch inference: 16, 32 images
- Clinical requirement: Real-time processing

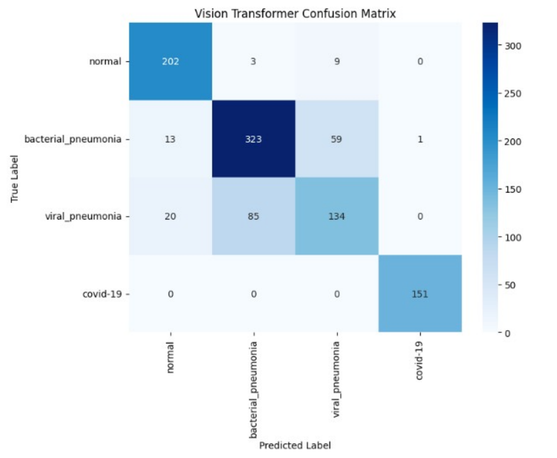
MODEL EFFICIENCY

- Parameters: VGG16 (138M), ViT (86M)
- Memory usage: Peak GPU memory during inference
- Model size: Compressed model for deployment

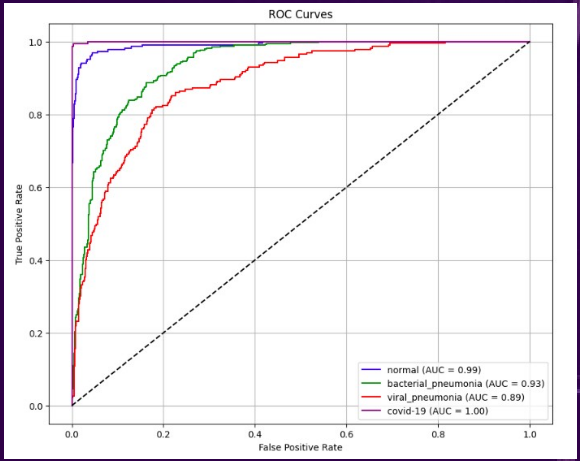
COMPARISON : VGG16 vs ViT

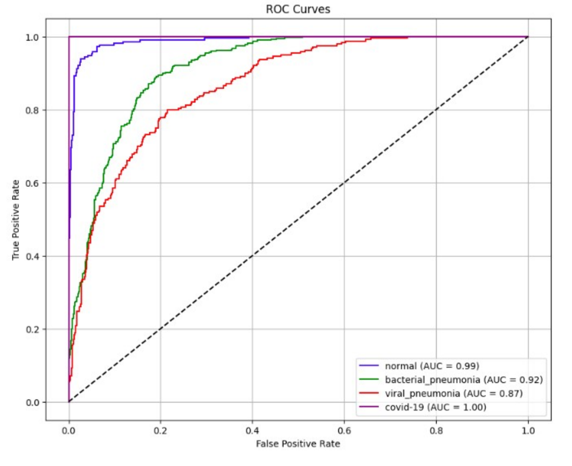
Confusion Matrix



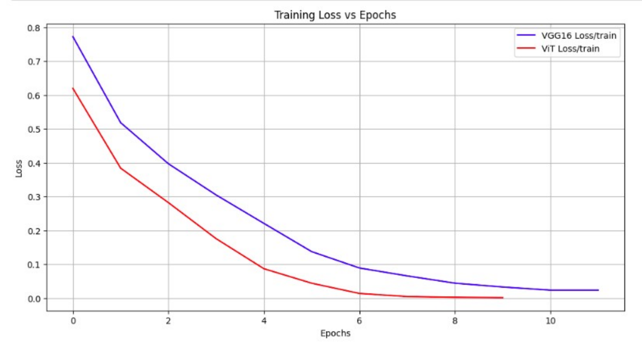


ROC Curves

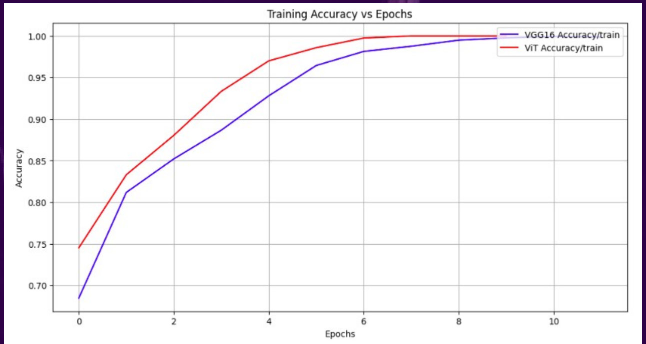




Training Loss / Epoch

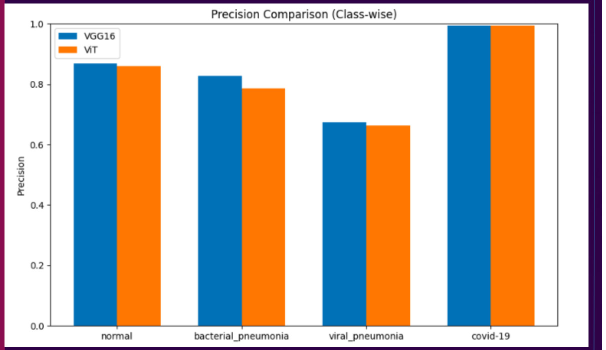


Training Accuracy / Epoch

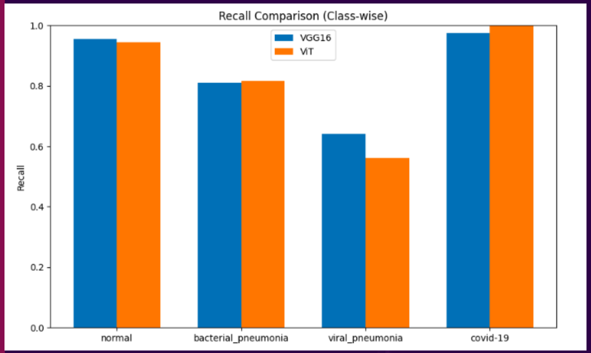


COMPARISON : VGG16 vs ViT

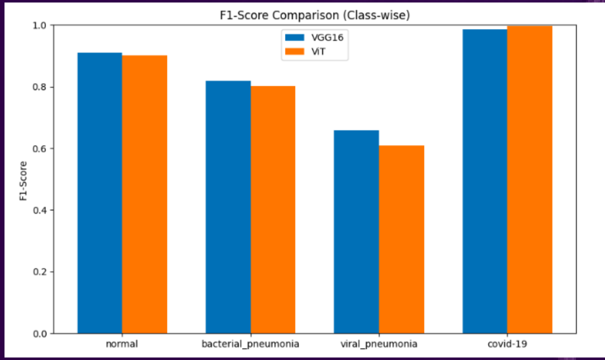
Precision



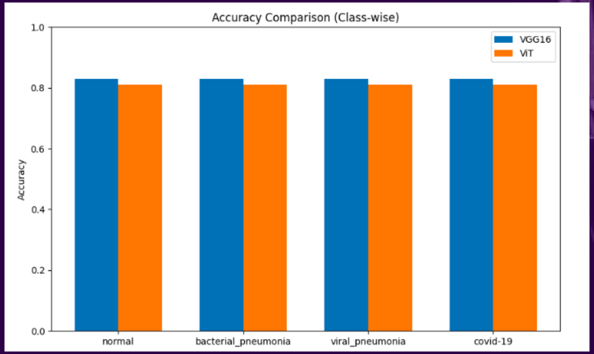
Recall



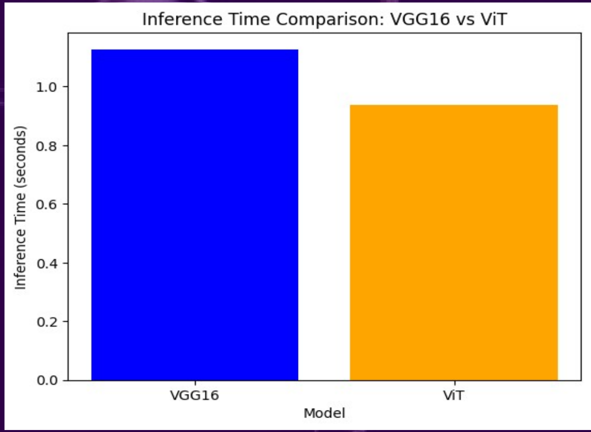
F1 Score



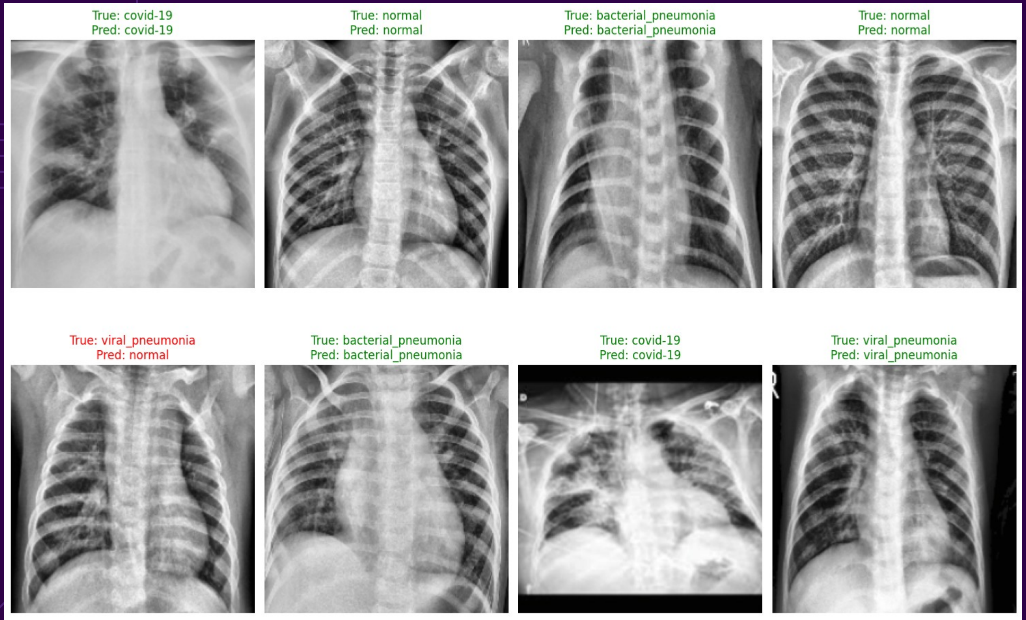
Accuracy



Inference Time

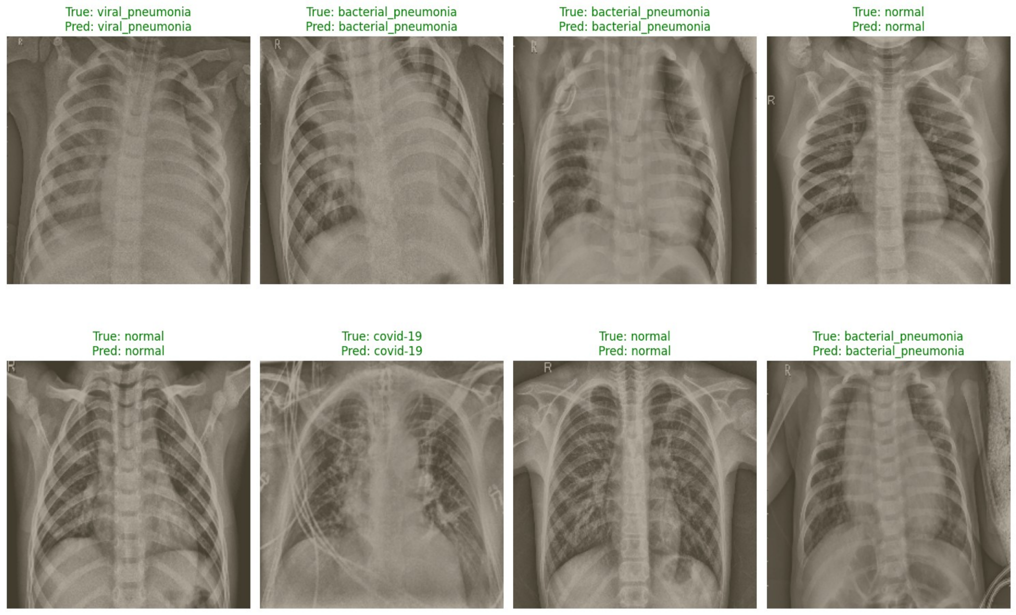


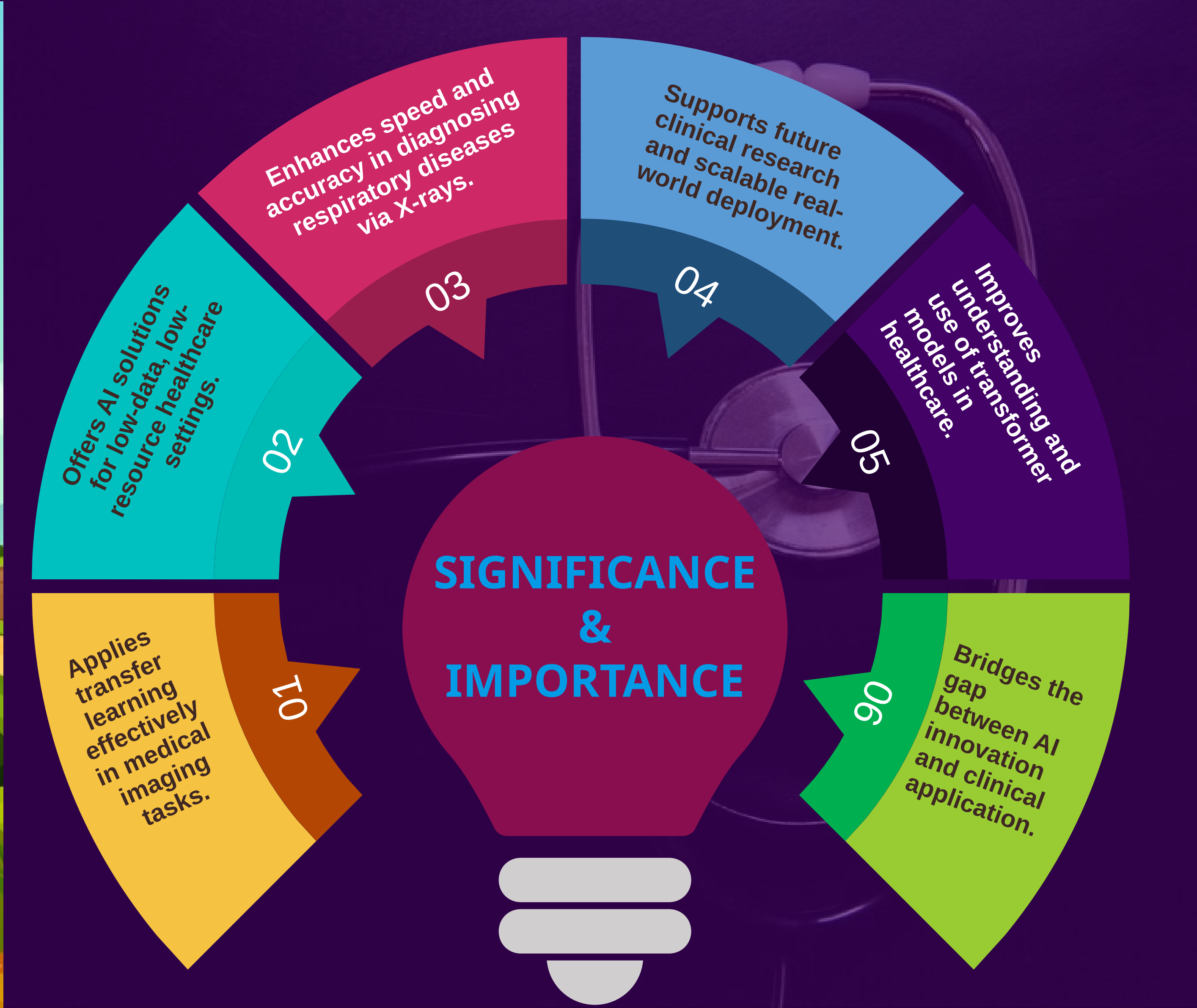
PREDICTIONS



VGG16 Predictions

ViT Predictions




SIGNIFICANCE & IMPORTANCE
01
Applies transfer learning effectively in medical imaging tasks.
02
Offers AI solutions for low-data, low-resource healthcare settings.
03
Enhances speed and accuracy in diagnosing respiratory diseases via X-rays.
04
Supports future clinical research and scalable real-world deployment.
05
Improves understanding and use of transformer models in healthcare.
06
Bridges the gap between AI innovation and clinical application.

CONCLUSION

01

Transfer learning shows strong potential for medical image classification, even with limited data.

02

Initial results suggest reduced training time and improved baseline accuracy.

03

Both VGG16 and ViT exhibit promise, with trade-offs in complexity and resource demands.

04

Ongoing work aims to further optimize accuracy and model performance.

05

This project lays groundwork for practical, efficient AI tools in healthcare diagnostics.

REFERENCES

[1] Kim H.E. et al. (2022). *Transfer Learning for Medical Image Classification: A Literature Review*. BMC Medical Imaging.

[2] Salehi M. et al. (2021). *Automated Detection of Pneumonia using Deep Transfer Learning with Pediatric Chest X-rays*. British J. Radiology.

[3] Irvin J. et al. (2019). *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels*. arXiv:1901.07031.

[4] GiBi13. *Pneumonia & COVID-19 Image Dataset*. Kaggle.
🔗 https://www.kaggle.com/datasets/gibi13/pneumonia-covid19-image-dataset

[5] Raghu M. et al. (2019). *Transfusion: Understanding Transfer Learning for Medical Imaging*. NeurIPS.

[6] Tajbakhsh N. et al. (2016). *CNNs for Medical Image Analysis: Full Training or Fine Tuning?* IEEE Transactions on Medical Imaging.

[7] Dosovitskiy A. et al. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. ICLR.

[8] Khan S.H. et al. (2022). *Transformers in Vision: A Survey*. ACM Computing Surveys.

[9] Wang X. et al. (2017). *ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks*. CVPR.

THANKS!

SHHAHID-UL-ISLAM
shahid9664@gmail.com
+917780858125
https://Khanz9664.github.io/portfolio
https://github.com/Khanz9664
https://linkedin.com/in/shahid-ul-islam-13650998