AI-POWERED LUNG DISEASE CLASSIFICATION

An End-to-End System with PyTorch and Streamlit

Eshan Puri (8025340013)
M.E. Artificial Intelligence
Thapar Institute of Engineering & Technology
Year: 2025

THE CLINICAL CHALLENGE & OUR MISSION

The Core Problem

- The immense workload on radiologists creates a need for tools that enhance diagnostic efficiency and consistency.

The Opportunity

- AI can serve as a powerful 'second opinion,' assisting clinicians with speed and reliability in analyzing Chest X-rays (CXRs).

Project Goal

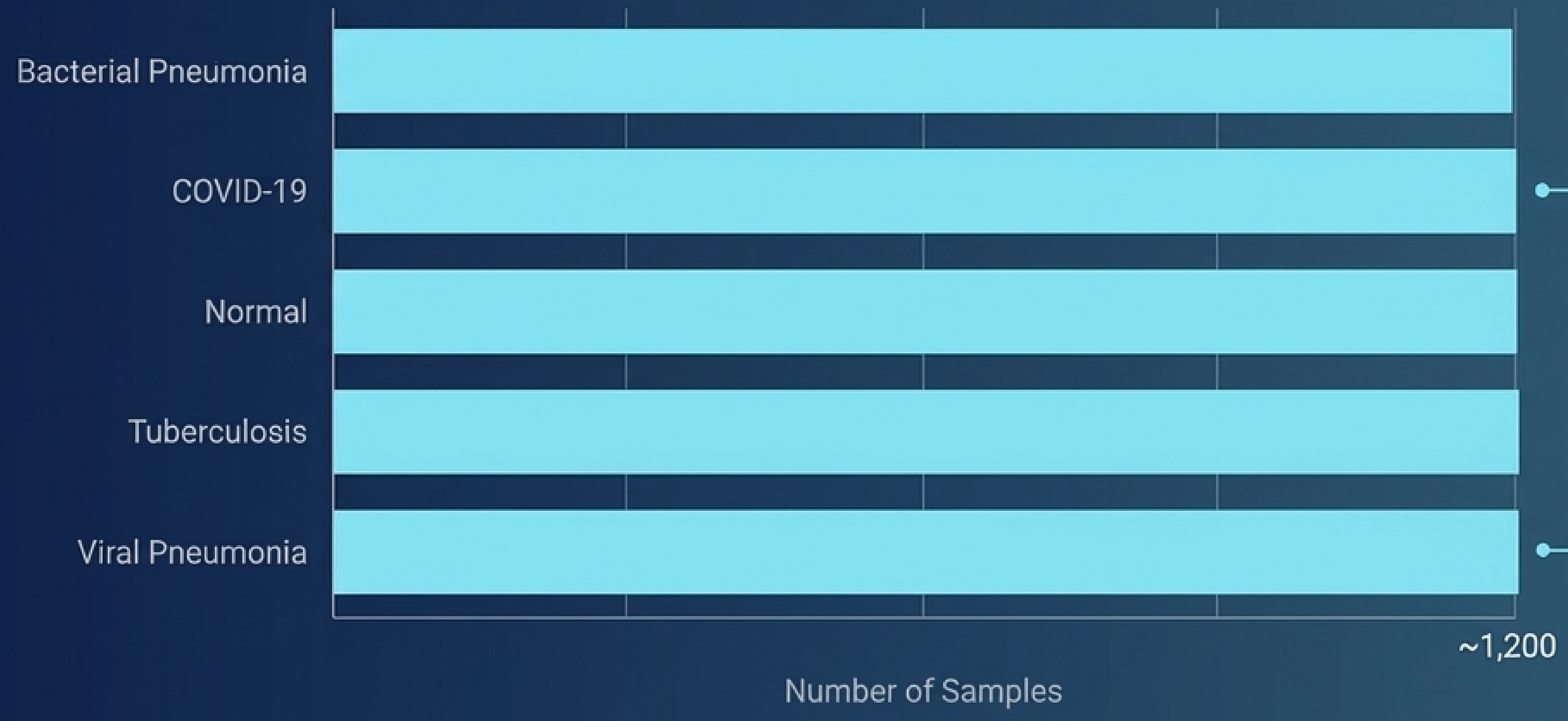
- To build a robust, automated system for classifying five key lung conditions:
  - Bacterial Pneumonia
  - COVID-19
  - Normal
  - Tuberculosis
  - Viral Pneumonia

THE FOUNDATION: A CURATED CXR DATASET

6,054
TRAINING IMAGES

2,016
VALIDATION IMAGES

2,027
TEST IMAGES



Radiographic Similarity

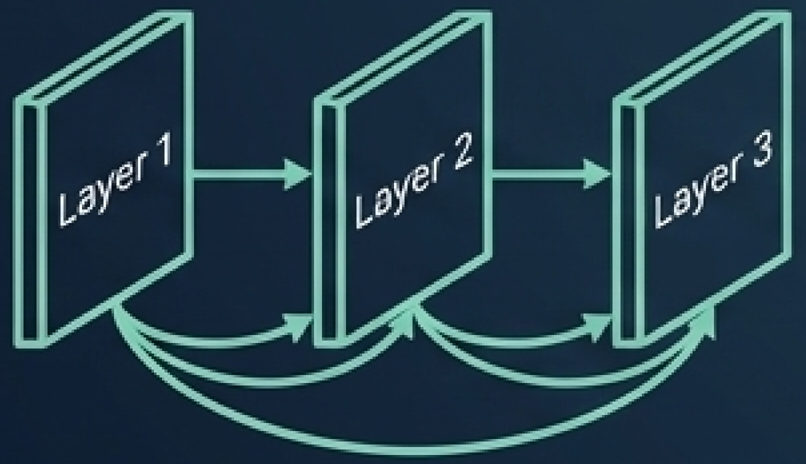
The subtle visual overlap between conditions like Bacterial and Viral Pneumonia presents a significant classification challenge.

Data Consistency

All images are standardized via resizing (224x224) and ImageNet normalization.

ARCHITECTURAL BLUEPRINT: A DIVERSE CNN PORTFOLIO

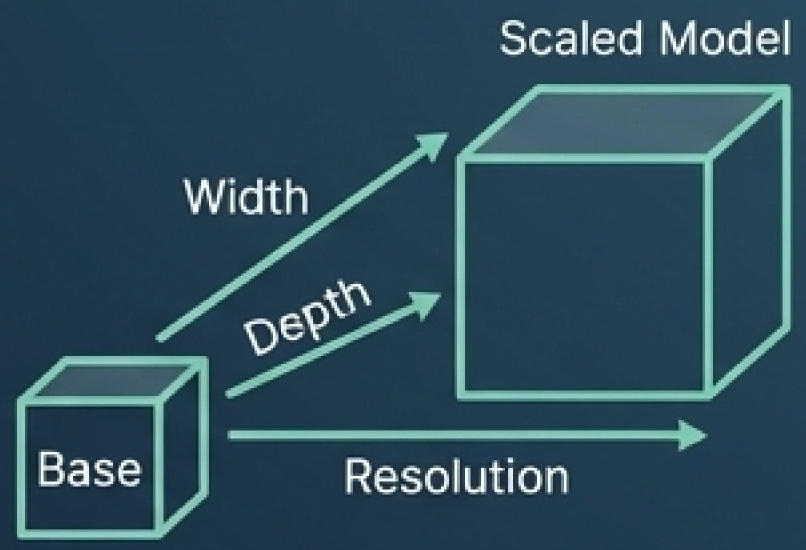
DENSENET121



Concept: Feature Reuse & Efficient Gradient Flow

Why Chosen: Its dense connectivity pattern encourages feature propagation and mitigates vanishing gradients.

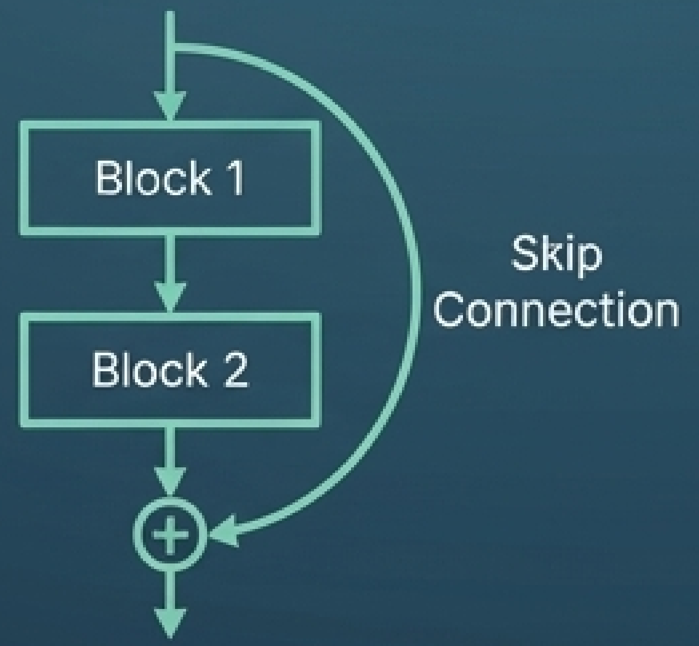
EFFICIENTNET-B0



Concept: Compound Scaling Efficiency

Why Chosen: A modern architecture offering an excellent balance of high accuracy and low computational cost (4.01M parameters).

RESNET50

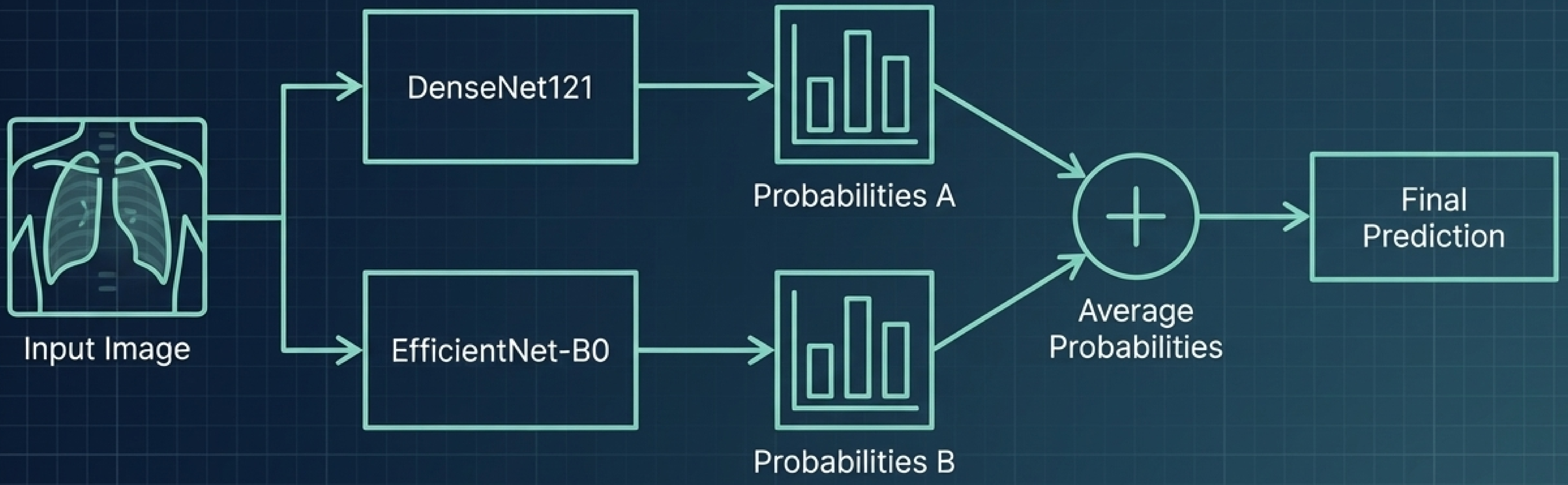


Concept: Deep Residual Learning

Why Chosen: A foundational architecture whose skip connections enable stable training of very deep networks.

THE ENSEMBLE: STRONGER THROUGH COLLABORATION

Method: Top-2 Soft-Average Ensemble



Rationale: By combining the outputs of two diverse, high-performing models, we reduce prediction variance, mitigate the weaknesses of any single architecture, and produce a more stable and accurate final classifier.

A DISCIPLINED & OPTIMIZED TRAINING REGIMEN

TWO-PHASE STRATEGY

1. Phase 1: Frozen Training (10 Epochs)

Action: Train only the final classification layer.

Rationale: Adapt the model to the new task without corrupting valuable pre-trained features.

2. Phase 2: Fine-Tuning (10 Epochs)

Action: Unfreeze and train the last two backbone blocks.

Rationale: Subtly adjust deeper features to the specific nuances of CXR data.

KEY TRAINING COMPONENTS

Optimizer: AdamW

Scheduler: Cosine Annealing with warm-up

Safeguard: Early Stopping (patience=5)

Acceleration: Automatic Mixed Precision (AMP)

FROM THEORY TO CODE: IMPLEMENTATION DETAILS

Modular Structure

Clean project organization with separated concerns (UI, models, utils).

Interactive UI

Streamlit's session state and callbacks for a responsive user experience.

PyTorch Fundamentals

Efficient data loading with `DataLoader`, tensor operations, and `autograd` for backpropagation.

Automated Reporting

The `FPDF` library for on-the-fly generation of downloadable PDF reports.

Model Agility

The `timm` library used as a model factory for rapid experimentation with state-of-the-art architectures.

A MULTI-FACETED VIEW OF PERFORMANCE

Accuracy

What percentage did we get right?

Precision

Of all 'COVID-19' predictions, how many were correct?

Recall

Of all actual COVID-19 cases, how many did we find?

F1-Score

The balanced harmonic mean of Precision and Recall. Crucial for managing the trade-off between the two.

DIAGNOSING THE ENSEMBLE: MISCLASSIFICATION PATTERNS

True Class \ Predicted Class	Bacterial Pneumonia	COVID-19	Normal	Tuberculosis	Viral Pneumonia
Bacterial Pneumonia	**85**	2	3	5	15
COVID-19	1	**92**	2	1	4
Normal	2	1	**95**	0	2
Tuberculosis	3	1	1	**90**	5
Viral Pneumonia	12	3	2	5	**88**

****Strong Performance****
High values along the diagonal, especially for Normal, COVID-19, and Tuberculosis.

****Primary Confusion Point****
Draw attention to the off-diagonal values between **Bacterial Pneumonia** and **Viral Pneumonia**, explaining this reflects a known clinical challenge.

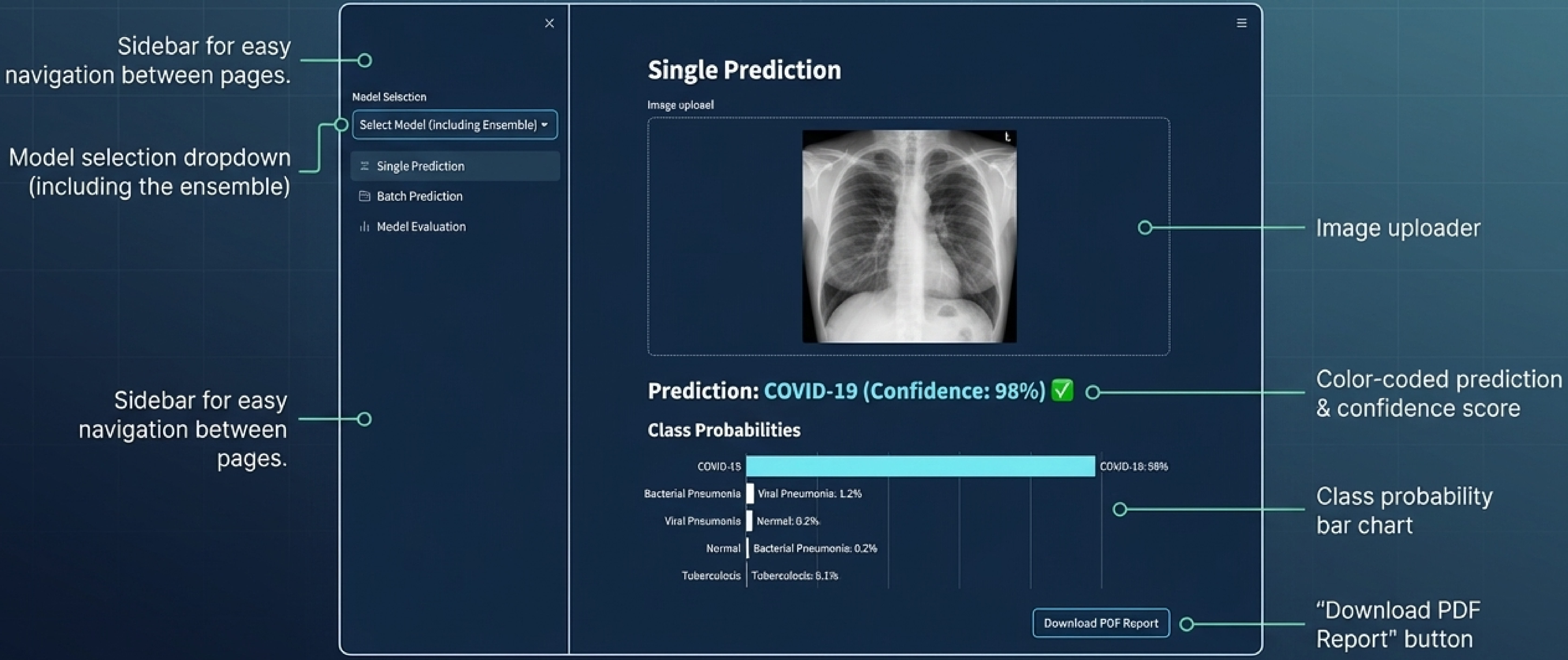
Takeaway: The analysis reveals the model's behavior mirrors real-world diagnostic difficulty, and the ensemble was most effective at minimizing these errors.

THE FINAL VERDICT: QUANTITATIVE RESULTS

Model	Test Accuracy	Parameters (M)
ResNet50	76.81%	23.52
DenseNet121	83.03%	6.96
EfficientNet-B0	84.71%	**4.01**
Ensemble (Top-2)	**85.40%**	**N/A**

- **Ensemble Wins:** The Top-2 ensemble clearly outperforms all individual models, validating the strategy.
- **Efficiency Over Size:** The compact EfficientNet-B0 (4.01M params) significantly outperforms the much larger ResNet50 (23.52M params), proving that superior architecture beats brute force.

FROM MODEL TO TOOL: THE STREAMLIT APPLICATION



Additional Functionality Mentioned: The app also includes pages for **Batch Prediction** (with .csv export) and **Model Evaluation**.

NotebookLM

DESIGNING FOR CLARITY, TRUST, AND USABILITY

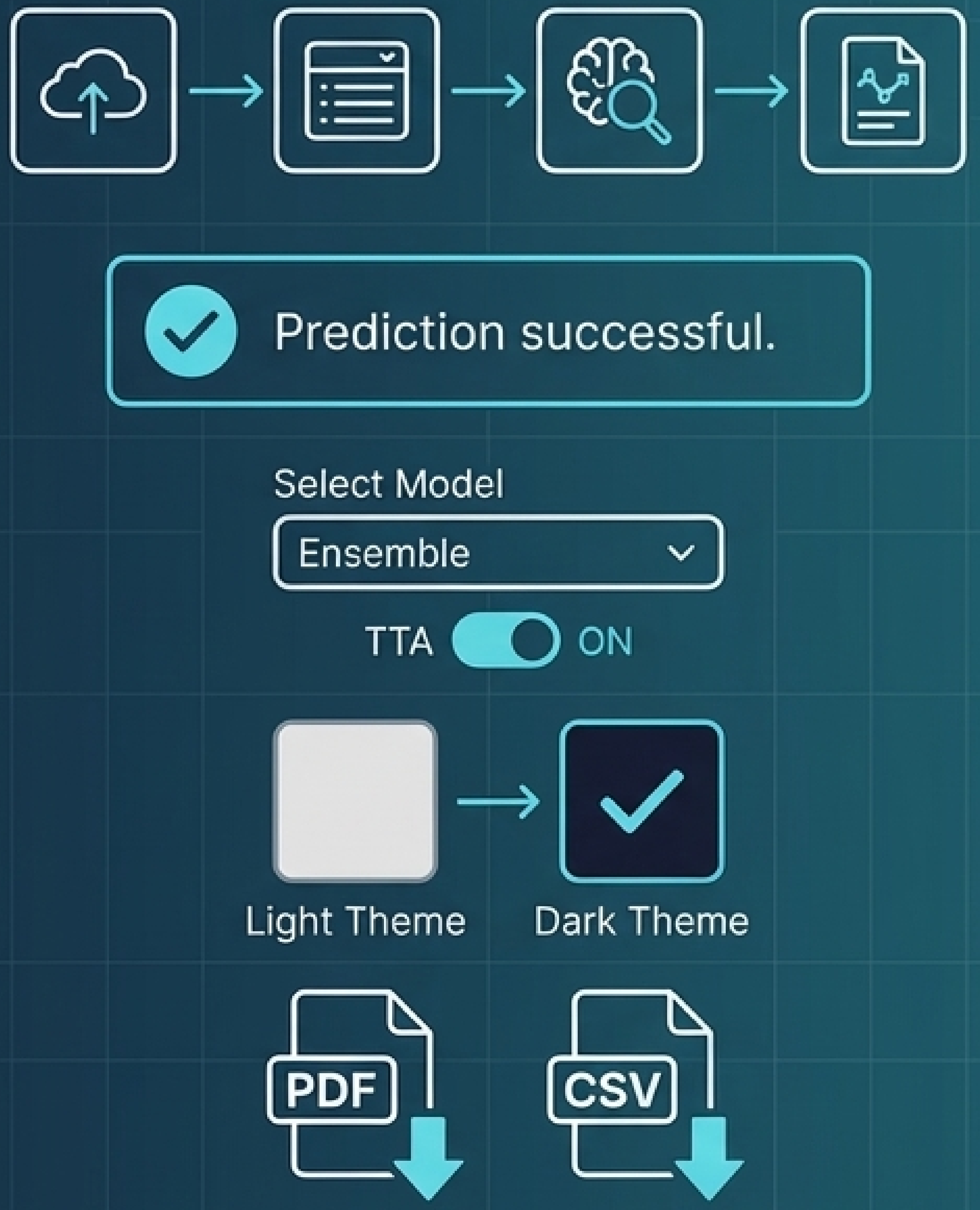
Intuitive Workflow
A simple, guided process: Upload -> Select Model -> Predict -> Report.

Informative Feedback
Use of success/warning messages and visual cues to clearly communicate status and results.

User Control
Empowering the user with options for model selection and Test-Time Augmentation (TTA).

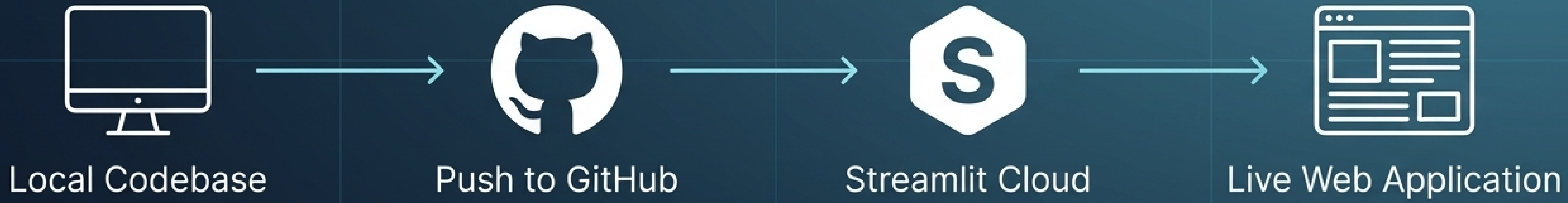
Professional Aesthetic
A dark theme was chosen to reduce eye strain and provide a modern, clinical feel.

Data Portability
Ensuring key outputs like PDF reports and batch CSV files are easily downloadable for offline use.



DEPLOYMENT AND REPRODUCIBILITY

Deployment Pipeline



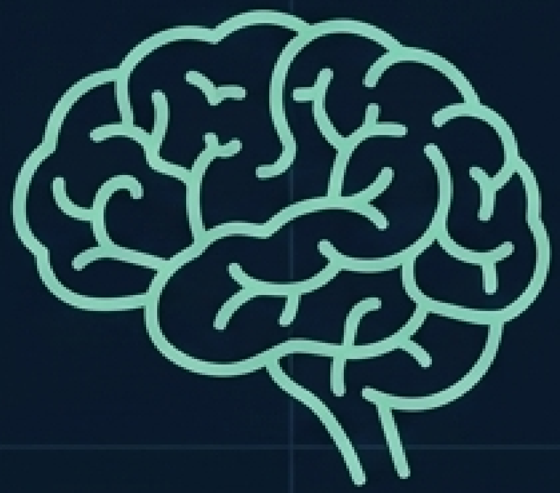
Key Components for a Reproducible Environment

- `requirements.txt` – Locks down all Python dependencies.
- `runtime.txt` – Specifies the exact Python version for the cloud environment.
- `.gitignore` – Manages the repository by excluding unnecessary files.
- `models/ folder – Contains pre-trained `.pth` model weights for fast inference on the deployed app.

AN HONEST ASSESSMENT: PROJECT LIMITATIONS

1. **Dataset Scope:** Performance is tied to the curated dataset; generalization to diverse real-world clinical data is not guaranteed.
2. **Lack of Interpretability:** The system is a ‘black box.’ Advanced tools like Grad-CAM were not implemented, which limits clinical trust by not explaining *why* a prediction was made.
3. **Basic Quality Checks:** The current blur and brightness checks do not handle complex radiographic artifacts or positioning issues.
4. **Absence of Clinical Context:** The model operates on images alone, without crucial patient metadata (e.g., age, symptoms) that a clinician uses.

THE PATH FORWARD: FUTURE WORK



MODEL-CENTRIC

- Integrate Grad-CAM for visual interpretability.
- Explore newer architectures like Vision Transformers (ViT).

DATA-CENTRIC

- Train on larger, multi-center datasets to improve domain generalization.
- Incorporate clinical metadata into the model.

APPLICATION-CENTRIC

- Implement an active learning loop for expert feedback.
- Expand to multi-label classification.

CONCLUSION: AN END-TO-END AI ENGINEERING SHOWCASE

- Successfully engineered a complete pipeline from data processing to an interactive, deployed tool.
- Demonstrated the superior performance of a Top-2 ensemble model, achieving **85.4%** test accuracy.
- Delivered a fully deployable, user-centric application that showcases Python's power in unifying backend AI with frontend UX.

GitHub Repository

`ESPK-Eshan/lung_app`