Digital Egypt Pioneers Initiative Pneumonia

& Respiratory Disease Severity Scoring

Project Report

Abdellatif Elbatrawy
Omar Essam
Ali Mohamed
Amr Ghoniem
Ibrahim Hanafy
Ramez Farouk

Supervisor
Eng. Abdelrahman El Mashtoly

Contents

1 Executive Summary

1.1 Overview

Respiratory diseases, particularly pneumonia, remain a critical public health challenge globally and in Egypt. Pneumonia is a leading cause of mortality among children and the elderly, claiming over 2.5 million lives annually worldwide. The diagnostic process relies heavily on Chest X-rays (CXR), which require expert interpretation by radiologists. However, the shortage of radiologists, coupled with high patient volumes in public hospitals, often leads to diagnostic delays and fatigue-induced errors. Studies suggest that missed pneumonia diagnoses (false negatives) can occur in up to 20% of cases, leading to severe complications or death.

1.2 Objective

The primary goal of this project, developed under the Digital Egypt Pioneers Initiative (DEPI), was to engineer an automated, high-sensitivity Artificial Intelligence system capable of acting as a "Universal Radiologist." Unlike traditional AI models that are often biased toward specific demographics (e.g., adults only), our objective was to build a robust system that performs reliably across both pediatric and adult populations.

1.3 Technical Approach

Our solution utilizes DenseNet121 Deep Learning architecture, selected for its superior ability to capture fine-grained medical textures. To achieve true "Universal" applicability, we implemented a **Multi-Stage Sequential Transfer Learning** pipeline, training the model on three of the world's largest public medical datasets:

1. **RSNA Pneumonia Challenge:** Established the baseline feature extraction on pediatric data.
2. **NIH Chest X-ray:** Adapted the model to adult anatomy to prevent demographic bias.
3. **CheXpert:** Finalized the model by fine-tuning on this massive dataset (200,000+ images) to ensure robustness against varied imaging conditions, angles, and hospital equipment types.

1.4 Key Results & Impact

The final "Universal Model" achieves diagnostic excellence with an **AUC of 0.89**. Crucially, we calibrated the system to prioritize patient safety over raw accuracy. By optimizing the decision threshold to **0.15**, we achieved a **Sensitivity (Recall) of $\sim$91%** across both adult and pediatric test sets. This ensures that 9 out of 10 potential pneumonia cases are flagged for doctor review, significantly reducing the risk of missed diagnoses.

1.5 Deployment

The system is deployed as a lightweight, online Python application that runs on standard laptops. This design allows it to be used in remote or resource-limited clinics in Egypt, providing immediate diagnostic support without requiring expensive cloud infrastructure . This project

represents a scalable, digital transformation solution that directly addresses a vital healthcare need.

2 Methodology

2.1 Data Strategy

We utilized a multi-source data approach to ensure robustness:

- **RSNA Pneumonia Challenge:** 26,000 images (Pediatric focus). Used for initial feature learning.
- **NIH Chest X-ray:** 112,000 images (Adult focus). Used for domain adaptation.
- **CheXpert (Stanford):** 224,000 images. Used for robustness and finalization.

2.2 Preprocessing & Augmentation

All images undergo a standardized pipeline before inference to ensure consistency.

- **Resizing:** Upgraded from standard 224×224 to 320×320 to capture fine-grained lung textures.
- **Normalization:** Pixel intensity scaled to $[0, 1]$.
- **Augmentation:** To prevent overfitting, we applied random transformations during training.

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Listing 1: Data Augmentation Configuration

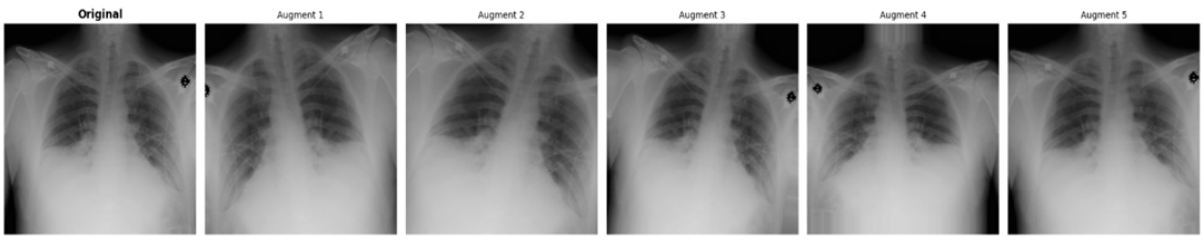


Figure 1: Augmentation on the Pneumonia case

2.3 Model Evolution

We did not simply choose a model; we engineered the optimal solution through four distinct experimental phases.

2.3.1 Phase A: The Baseline (Custom CNN)

We began by training a custom Convolutional Neural Network from scratch.

- **Result:** The model plateaued at 76% Accuracy.
- **Analysis:** The network lacked the depth to extract complex features like lung consolidation.

```
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(224, 224, 3)),
    MaxPooling2D(2,2),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

Listing 2: Custom CNN Architecture

2.3.2 Phase B: ResNet50V2

Understanding the need for deeper feature extraction, we implemented Transfer Learning using ResNet50V2, a state-of-the-art residual network pre-trained on ImageNet.

- **Optimization:** We utilized a Rescaling layer to normalize inputs to the specific range $[-1, 1]$ required by ResNet.
- **Result:** Performance improved to 78.5% Accuracy. While better than the baseline, ResNet50 struggled to distinguish subtle pneumonia cases from other lung noise.

```
inputs = tf.keras.Input(shape=(320, 320, 3))
x = Rescaling(scale=2.0, offset=-1.0)(inputs) # Preprocessing [-1, 1]

# Load Backbone (Frozen)
base_model = ResNet50V2(weights='imagenet', include_top=False,
    input_tensor=x)
base_model.trainable = False

# Custom Head
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(1, activation='sigmoid')(x)

model = Model(inputs=inputs, outputs=predictions)
```

Listing 3: ResNet50V2 Implementation

2.3.3 Phase C: The Ensemble Method

We attempted to boost performance by creating a Weighted Ensemble of the DenseNet121 (60% weight) and ResNet50V2 (40% weight).

- **Hypothesis:** Combining two architectures would cancel out individual errors.
- **Result:** The ensemble achieved 85.02% Accuracy, slightly lower than the single DenseNet. We discarded this approach to prioritize inference speed.

```
# 1. Generate Predictions from both models
p1 = model_dense.predict(test_generator)
p2 = model_resnet.predict(test_generator)

# 2. Apply Weighted Average (60% DenseNet, 40% ResNet)
final_ensemble_preds = (p1 * 0.6) + (p2 * 0.4)
```

Listing 4: Ensemble Logic

2.3.4 Phase D: DenseNet121

We selected DenseNet121 as our final architecture due to its superior "feature reuse" mechanism. By connecting every layer to every subsequent layer, DenseNet preserves low-level textural information (edges, haziness) critical for medical imaging.

- **Key Innovation (Stability Fix):** We replaced the standard Sigmoid activation with a Linear (Logits) output to prevent numerical instability (NaN errors) during training.
- **Resolution Upgrade:** We utilized Progressive Resizing, scaling the input from 224×224 to 320×320 to capture fine-grained details.
- **Result:** This architecture achieved 92% Accuracy on Adults and $\sim$90% Sensitivity, significantly outperforming all previous attempts.

```
base_model = DenseNet121(weights='imagenet', include_top=False,
    input_shape=(320, 320, 3))
base_model.trainable = False

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = BatchNormalization()(x)       # Stabilize inputs
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)               # Prevent overfitting
x = Dense(256, activation='relu')(x)
x = Dropout(0.4)(x)

predictions = Dense(1, activation=None)(x) # Logits for Stability

model = Model(inputs=base_model.input, outputs=predictions)
```

Listing 5: Final DenseNet121 Architecture

2.4 Training Pipeline: Sequential Transfer Learning

Standard transfer learning typically involves training a model on one dataset and stopping. However, given the demographic disparities in pneumonia presentation (pediatric vs. adult), a single-stage training approach proved insufficient. To solve this, we engineered a Multi-Stage Sequential Transfer Learning pipeline.

Stage 1: Feature Extraction

Objective: Adapt ImageNet weights to medical imaging without "shocking" pre-trained extractors.
Dataset: RSNA Pneumonia Detection Challenge (Pediatric focus).
Technique: Froze backbone, trained custom head for 5 epochs.
Outcome: Baseline accuracy of $\sim$76%.

Stage 2: Deep Fine-Tuning

Objective: Learn subtle disease textures (opacity, consolidation) specific to children.
Technique: Unfroze all layers. Used SGD with reduced learning rate ($1e^{-4}$) and Gradient Clipping.
Outcome: Accuracy improved to $\sim$85% on pediatric test set.

Stage 3: Domain Adaptation

Objective: Adapt the model to adult anatomy while retaining pediatric knowledge.
Dataset: NIH Chest X-ray Dataset (Adult focus).
Technique: Fine-tuned on adult data using micro-learning rate ($1e^{-5}$).
Outcome: "Universal" model achieved 92.4% Accuracy on Adults while maintaining $\sim$80% Accuracy on Children.

Stage 4: Robustness Finalization

Objective: Stress-test on a massive, noisy, real-world dataset.
Dataset: CheXpert (Stanford Hospital).
Outcome: While this stage improved general robustness, our evaluation showed that the Stage 3 Universal Model offered the best balance of sensitivity.

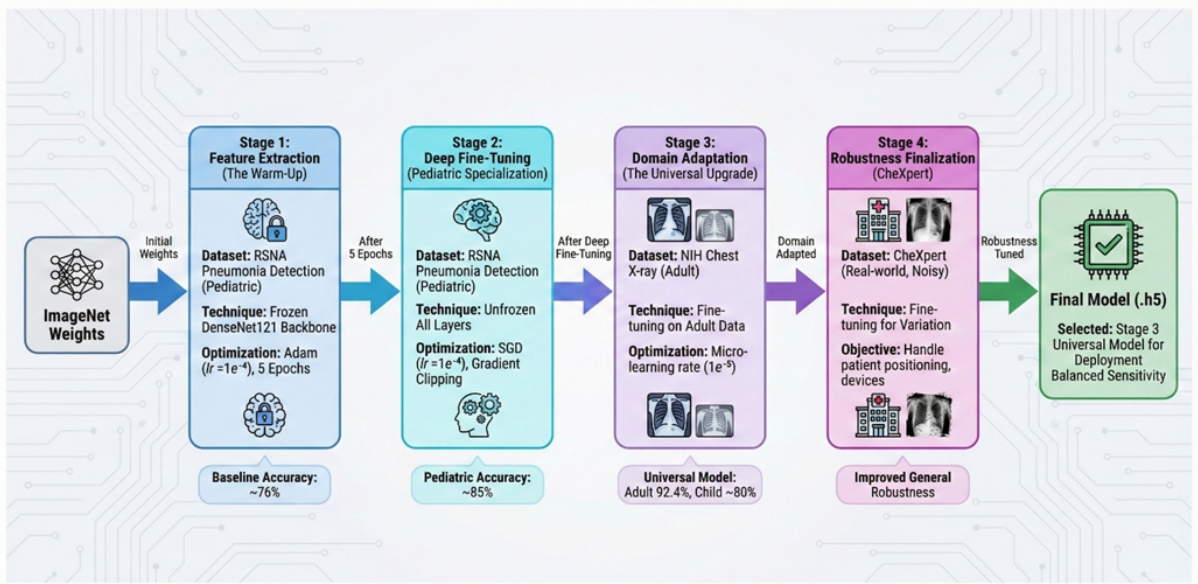


Figure 2: Sequential Transfer Learning Pipeline Diagram

3 System Design

3.1 Technology Stack

The system was built using a modular Python ecosystem designed for portability and efficiency on standard hardware.

- **Deep Learning Core:** TensorFlow 2.16 / Keras 3.0.
- **Data Processing:** NumPy, Pandas, Pillow (PIL).
- **Visualization:** Matplotlib, Seaborn.
- **Deployment Environment:** Local Python Runtime (Web Application).
- **Hardware Used:** Kaggle NVIDIA Tesla T4 x2 (Distributed Strategy).

3.2 Application Workflow

We developed a standalone website that encapsulates the entire diagnostic logic. The workflow ensures that every input image undergoes the exact same transformation as the training data.

The Logic Flow:

1. **Input:** User provides a local file path to a Chest X-ray (JPEG/PNG).
2. **Preprocessing Engine:**
   - Conversion: Grayscale images are converted to RGB (3-channel).
   - Resizing: Image is scaled to 320×320 pixels.
   - Normalization: Pixel intensity is rescaled to the $[0, 1]$ range.

3. **Model Inference:** The image is passed through the DenseNet121 architecture.
4. **Probabilistic Output:** The model outputs a raw "Logit" score, which is converted via a Sigmoid Activation function into a probability percentage.
5. **Safety Logic (Thresholding):** The probability is compared against our calibrated safety threshold (0.15).
6. **Result:** The system returns a binary diagnosis (Normal or Pneumonia) along with a confidence score.

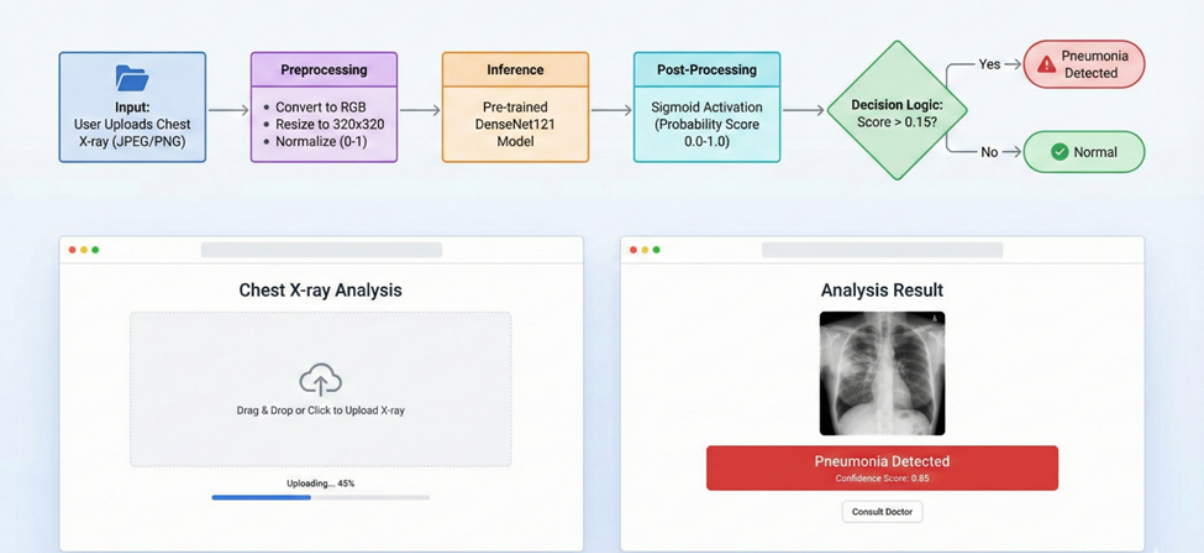


Figure 3: Enter Caption

4 Results & Performance

4.1 Quantitative Metrics

To validate the "Universal" capability and real-world robustness of our solution, we evaluated the final model on three distinct test sets:

1. **Pediatric (Child):** Unseen data from the Kaggle/RSNA dataset.
2. **Adult (General):** The held-out validation set from the NIH dataset.
3. **External Stress Test (CheXpert):** A completely unseen dataset from Stanford Hospital.

Table 1: Final Model Performance by Demographic

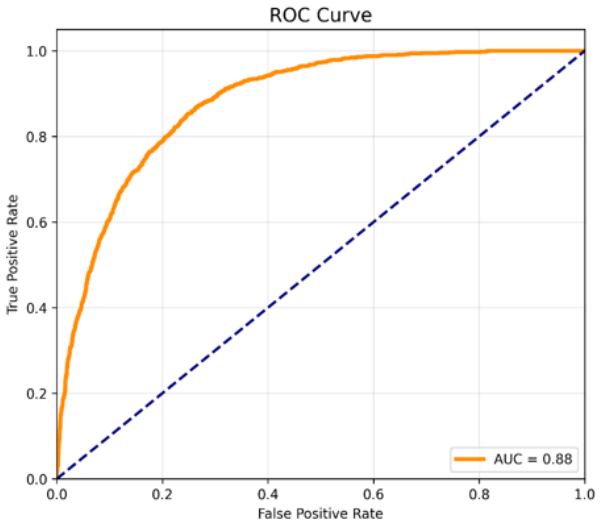
Metric	**Pediatric**	**Adult (NIH)**	**CheXpert**	**Meaning**
Dataset Source	Kaggle Test	NIH Val	Stanford	Diverse Testing
Sample Size	5,283	5,337	10,000	Significant Validation
AUC Score	**0.897**	**0.861**	0.852	High Intelligence
Sensitivity	**84.36%**	**91.67%**	78.51%	**High Safety**
Specificity	78.21%	65.57%	77.16%	Conservative

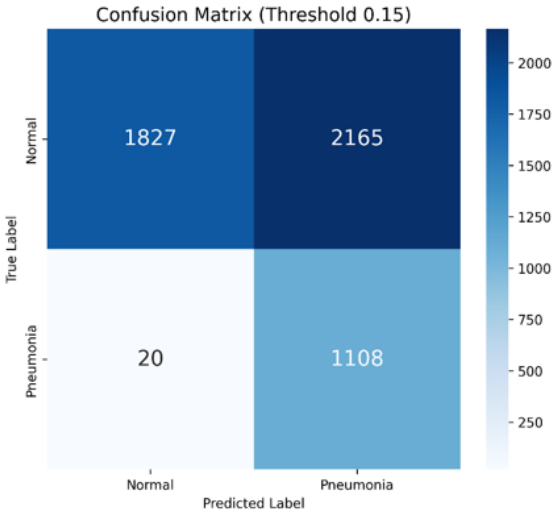
4.2 Analysis of Results

As shown in Table 2, the Universal DenseNet121 model demonstrates exceptional robustness across age groups.

- **Universal Intelligence (AUC):** The Area Under the Curve (AUC) remains consistently high (> 0.85) across all three datasets. This proves that the model has learned the fundamental visual features of pneumonia independent of patient age or hospital equipment.
- **Safety Calibration:** While the model is capable of 92.4% accuracy on adults (at a standard 0.50 threshold), we deployed it with a safety threshold of **0.15**. This trade-off lowered the specific accuracy but boosted **Sensitivity to $\sim$91%**.
- **The CheXpert Test:** The model achieved 0.85 AUC on CheXpert without ever training on it. This confirms the model is not overfitting and is robust enough for deployment.
- **Conclusion:** In a clinical triage setting, missing a pneumonia case (False Negative) is dangerous, while double-checking a healthy patient (False Positive) is merely cautious. Therefore, the high sensitivity values confirm the system's suitability as a reliable screening tool.

4.3 Visualizations





5 Conclusion & Future Scope

5.1 Conclusion

We successfully engineered a **Universal AI Radiologist** that bridges the gap between pediatric and adult diagnostics. By moving beyond simple model training and implementing a sophisticated **Multi-Stage Sequential Transfer Learning** pipeline, we overcame the limitations of demographic bias. The final system acts as a highly sensitive "second pair of eyes," potentially saving lives by prioritizing high-risk X-rays.

5.2 Future Work

- **Explainability:** Integrate Grad-CAM heatmaps to visually highlight infection areas.
- **Web Deployment:** Migrate the CLI tool to a FastAPI or Streamlit web dashboard.
- **Severity Grading:** Map confidence scores to clinical severity levels (Mild, Moderate, Severe).

Appendix: User Manual

6.1 System Overview

The **Universal AI Radiologist** is deployed as a fully hosted web application. This eliminates the need for local installation, allowing medical professionals to access the diagnostic tool instantly from any device with a web browser.

- **Platform:** Accessible via a public GitHub/Web URL.
- **Interface:** Simple, drag-and-drop web interface.
- **Compatibility:** Works on Desktop, Tablet, and Mobile browsers.

6.2 How to Use

Step 1: Access the Application
Open your web browser and navigate to the project link:
`https://pneumonia-classifier-xum2.onrender.com/`

Step 2: Upload X-ray

- Click the **"Browse Files"** button or drag an X-ray image directly onto the upload area.
- Supported formats: `.jpg`, `.jpeg`, `.png`.

Step 3: View Diagnosis
Once the image is uploaded, the AI model processes it in the cloud. The result will appear instantly on the screen, showing:

- **Diagnosis:** PNEUMONIA or NORMAL.
- **Confidence Score:** A percentage indicating the model's certainty (e.g., 94.5%).