**My First Mistake – Choosing the Wrong Dataset**

**"Bigger is Better" – A Misconception That Cost Me Time**

At first, I assumed that using **the largest dataset available** would automatically result in the best model. So, I chose the **NIH ChestX-ray14 dataset**, which contains over **112,000 chest X-ray images**.

But after spending **a lot of time preparing and analyzing it**, I realized it **wasn't the right choice** for my problem. Here’s why:

**🚨 Problems with the NIH Dataset**

1. **Extreme Class Imbalance:**
   * The dataset had **far more normal cases than pneumonia cases**, making it difficult for a model to learn pneumonia-specific patterns.
2. **Multi-Label Confusion:**
   * Many images were labeled with **multiple diseases at once** (e.g., pneumonia + tuberculosis), making it harder to train a model that focuses on just pneumonia.
3. **Data Leakage Risk:**
   * The same patient's images appeared in **both training and validation sets**, leading to overfitting and **false confidence in model accuracy**.

After struggling with these issues for **weeks**, I finally accepted that **a large dataset with poor labels is worse than a smaller, well-labeled dataset**.

**💡 The Solution – Switching to the RSNA Pneumonia Dataset**

After wasting time on the **NIH dataset**, I found the **RSNA Pneumonia Detection Challenge dataset**, which was **better suited** for my task.

✅ **Why RSNA Was the Right Choice:**

* **Labeled bounding boxes for pneumonia regions** (making it easier for the model to learn).
* **More balanced class distribution** (~8,851 pneumonia cases out of 25,684 images).
* **Higher-quality images** specifically focused on pneumonia detection.

This dataset change was a turning point—it allowed me to **build a model with meaningful learning instead of just handling dataset biases.**

**📌 3. Data Preparation – Converting Raw X-Rays for AI Training**

Once I had the right dataset, I had to **preprocess the images** before training a deep learning model.

**🔹 Preprocessing Steps**

1. **DICOM to PNG Conversion**
   * The RSNA dataset contained images in **DICOM format** (used in medical imaging).
   * I converted them to **PNG format**, which is easier to process with deep learning models.
2. **Resizing Images**
   * I resized all images to **224×224 pixels**, ensuring they matched the input size required by my model (**EfficientNetB0**).
3. **Normalizing Pixel Values**
   * X-ray images have different brightness levels, so I normalized all pixel values to **[0,1]** to stabilize training.

**🚨 Challenge: Image Format Mismatch During Testing**

After training my model, I noticed **errors when passing test images into the model**. The problem?

* **The test images had a different format from the training images**, causing mismatches in pixel intensity.

**💡 The Fix:**

I **standardized the preprocessing pipeline** so that **both training and test images** followed the same **conversion, resizing, and normalization steps**.

**📌 4. Model Selection – Finding the Best Deep Learning Architecture**

Initially, I considered **ResNet**, **VGG16**, and **MobileNet**, but after some experiments, I realized that **EfficientNetB0** was the best fit.

**🔹 Why EfficientNetB0?**

✅ **Higher accuracy with fewer parameters** than traditional CNNs.  
✅ **Computationally efficient**, making training and inference faster.  
✅ **Scales depth, width, and resolution intelligently**, improving performance.

I trained **EfficientNetB0** with transfer learning, starting with **frozen layers** to use its pre-trained features.

**📌 5. Training My Model – The Turning Point**

At first, I trained the model with **frozen layers** (meaning only the final classification layer was trainable).

* This **limited its learning**, and I **got poor results**:
  + **AUC = 0.579** (close to random guessing).
  + **Recall = 0.336 (too many missed pneumonia cases).**

**💡 The Fix – Fine-Tuning the Model**

To make the model **learn meaningful pneumonia features**, I **unfroze the layers** and fine-tuned it.

📌 **Final Training Results After Fine-Tuning:**

| **Epoch** | **Training AUC** | **Validation AUC** | **Validation Recall** |
| --- | --- | --- | --- |
| 1 | 0.6483 | 0.5791 | 0.3364 |
| 5 | 0.8312 | 0.8372 | 0.8252 |
| 10 | **0.8720** | **0.8532** | **0.8374** |

🚀 **Final Model Achievements:**  
✅ **AUC = 0.853 (high confidence in classification).**  
✅ **Recall = 0.837 (fewer missed pneumonia cases).**  
✅ **Accuracy = 72.6% (consistent predictions).**

**📌 6. Challenges and How I Overcame Them**

| **Challenge** | **Problem** | **Solution** |
| --- | --- | --- |
| **Dataset Choice** | NIH dataset had poor labels, causing wasted time. | Switched to **RSNA dataset** for better-labeled pneumonia cases. |
| **Class Imbalance** | Pneumonia cases were outnumbered by normal cases. | Used **class weighting and data augmentation** to balance data. |
| **Frozen Layers Limited Learning** | Model couldn’t extract pneumonia-specific features. | **Unfroze layers and fine-tuned** EfficientNetB0. |
| **Image Format Mismatch** | Test images had different formats, causing errors. | Standardized preprocessing pipeline. |

**📌 7. Conclusion & Future Work**

**🚀 Key Takeaways**

✅ **Choosing a high-quality dataset is more important than dataset size.**  
✅ **Fine-tuning the model significantly improved accuracy and recall.**  
✅ **EfficientNetB0 was the best model for pneumonia detection, achieving AUC = 0.853.**

**🔍 Future Work**

1. **Exploring ensemble learning (EfficientNet + ResNet hybrid) for better robustness.**