Bias Beyond Accuracy: A Deep Learning Study on Race and Chest X-ray Diagnosis

Prerna Joshi CS 4620 Senior Project Report
Under Supervision of Prof Amar Raheja
California State Polytechnic University, Pomona, California

I. ABSTRACT

This study investigates the presence of racial bias in deep learning models used for chest X-ray (CXR) diagnosis, with a focus on how model performance varies across demographic groups. Using the CheXpert dataset, two convolutional neural network (CNN) models based on the ResNet-18 architecture were trained: one on a racially homogeneous dataset (White patients only), and another on a diverse, mixed-race dataset. The models were then evaluated using AUC scores across 14 disease categories to examine their diagnostic accuracy on both same-race and different-race test sets. The results reveal that models trained on homogeneous data tend to underperform when diagnosing patients from different racial backgrounds, raising concerns about fairness, generalizability, and patient safety in medical AI applications. This work underscores the critical importance of incorporating demographic diversity into training datasets to reduce bias and improve equity in healthcare AI systems.

II. INTRODUCTION

According to a National Library of Medicine review, machine learning tools developed for interpreting chest X-rays (CXR) are highly effective, enhance clicinicans' performances and boost the efficiency of radiology workflows (cite source). During Covid 19 pandemic, the world saw how the integration of machine learning into chest X-ray analysis for COVID-19 diagnosis illustrated the dual nature of AI in healthcare—its benefits and its drawbacks (cite source). This duality becomes even more obvious when examining how these technologies perform across diverse patient populations, raising critical questions about bias, fairness, and equity—questions central to the concerns addressed in this paper.

With the use of AI growing in every field today, it's important to understand how these systems can overlook or mishandle factors they weren't explicitly trained to recognize—like biases. In chest X-ray imaging, such biases mean that demographic and racial differences in the training data can directly affect how well the model performs when diagnosing patients from different backgrounds. This raises serious concerns about patient safety and highlights the need to reduce false diagnoses across all groups. The key focus of this paper is to investigate how the performance of a machine learning model varies across different racial and age groups by comparing results from convolutional neural network models trained on homogeneous versus demographically diverse datasets, in order to evaluate whether racial and age bias emerges in lung disease prediction.

The two models being investigated for racial bias share the exact same architecture but differ in the composition of their training data. Model A is trained exclusively on chest X-rays from patients of a single racial group (White, in this case), while Model B is trained on a more racially diverse dataset that includes chest X-rays from patients identified as White, African American, Hispanic, Asian, and other racial categories included in the dataset (insert full list here). This comparison allows for an evaluation of how training data diversity impacts the model's performance across racial groups. ***(***** add all labels here after).*** To evaluate the performance of both models on the same dataset, Area Under the Curve (AUC) scores are used for various disease labels across both same-race and different-race test sets. This evaluation assesses potential racial bias in medical image classification models by comparing their performance across these subsets. Each model is tested on both same-race and different-race data, with the results highlighting whether a model's ability to detect conditions varies when applied to individuals of a different race than those it was trained on

III. DATASET DESCRIPTION

The CheXpert dataset was used for these experiments. CheXpert is a large, labeled dataset comprising 224,316 chest radiographs from 65,240 patients. It includes both frontal and lateral views and provides labels for 14 common chest conditions, including uncertain findings extracted from radiology reports using a rule-based labeler. The dataset is notable for capturing the inherent uncertainty in radiologic interpretation, which is leveraged during model training. CheXpert also includes validation and test sets annotated by board-certified radiologists, making it a strong benchmark for evaluating model performance against expert-level interpretation.

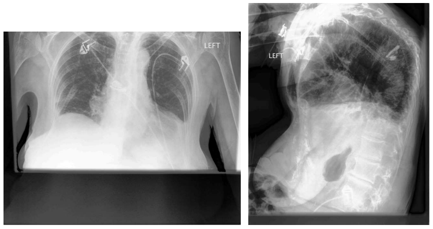


Figure 1.

The data splitting for the dataset was based on the "Race" column and used "White" as the target race for the **Single-Race** setup. For the **Single-Race** setup, I randomly sampled 50,000 White patients for training and used another 10,000 White patients for testing — so the model is trained and tested on the same race. For the **Diff-Race** setup, I created a mixed-race training set by pulling 50,000 random samples regardless of race, then used 10,000 non-White patients as the test set — this gives us a model trained on diversity and tested on a specific group the model wasn't entirely focused on. To compare fairly, I also tested each model on both test sets — so the Single-Race model gets evaluated not just on White patients (same race), but also on non-White patients (different race), and vice versa. This helps show how each model handles generalization across racial groups.In this study, we conduct a comprehensive evaluation of model performance across all 14 conditions labeled in the CheXpert dataset and how they are affected based on race. These include Enlarged Cardiomediastinum, Cardiomegaly, Lung Opacity, Lung Lesion, Edema, Consolidation, Pneumonia, Atelectasis, Pneumothorax, Pleural Effusion, Pleural Other, Fracture, Support Devices, and the absence of pathology ("No Finding").

However no explicit balancing of patient characteristics such as age, sex, or race was performed during dataset preparation. Consequently, the demographic distributions within the training and testing subsets reflect the natural occurrence of these attributes in the original dataset. While this maintains the real-world characteristics of the data, it may also introduce demographic biases that could influence model performance. The potential impact of these imbalances should be considered when interpreting results. Future iterations of this work may benefit from stratified sampling or reweighting methods to better control for confounding demographic variables.

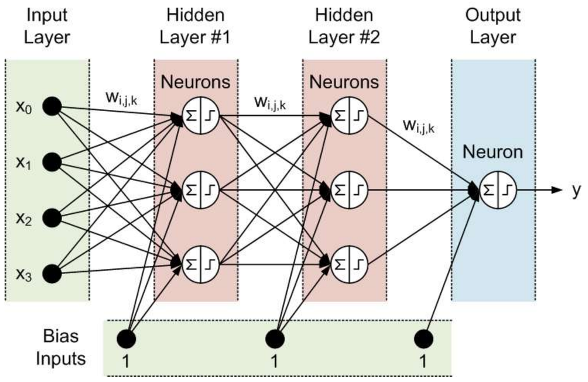
IV. PREPROCESSING AND DATA AUGMENTATION

For preprocessing purposes, all images were resized to a uniform size of 224x224 pixels to match the input dimensions which are expected by the pretrained ResNet-18 model. Images were then converted to PyTorch tensors using transforms.ToTensor(), which scales pixel values to the [0,1] range. transforms.ToTensor() function converts a PIL image (or a NumPy array) into a 3D PyTorch tensor with shape [C, H, W] (Channels, Height, Width), which is the expected input format for most CNNs like ResNet scales them from the original pixel values of the range [0, 255] to the range [0.0, 1.0], which helps prevent issues with large gradients and makes training more stable thus helping the model to converge faster. The same resizing preprocessing steps (resizing to 224×224 and conversion to tensor) were also applied to test images as well. No data augmentation techniques such as random flips, rotations, or brightness adjustments were applied in this version of the training pipeline. This was done to preserve the medical integrity of chest X-ray images, where even small changes in orientation can affect interpretability. No augmentation or distortion was introduced during testing to ensure consistent and unbiased evaluation.

V. CNNs IN MEDICAL IMAGING

Medical images are typically high-resolution and contain dense pixel information. CNNs, and especially architectures like ResNet, are ideal for handling this due to their ability to learn both local and global patterns across many layers. ResNet's depth allows it to capture complex features critical for diagnosing diseases from X-ray scans or similar imaging data. Convolutional Neural Networks (CNNs) in medical imaging are essentially like giving a computer a pair of intelligent eyes—allowing it to analyze and understand X-ray scans much like how a human would. CNNs are inspired by the structure of the human brain, particularly how biological neurons connect and transmit signals. In this case, the machine mimics human visual processing by passing images through layered networks that extract and refine features at each stage to make sense of what it sees.

The image below shows a simplified view of a fully connected neural network - The **Input Layer** takes in multiple values (in the case of imaging, these could be pixel intensities from the X-ray). **Hidden Layers** (shown as Layer #1 and #2) contain **neurons** that compute weighted sums of their inputs and pass them through activation functions to introduce non-linearity. Each neuron is connected to every neuron in the next layer, and bias inputs (shown at the bottom) help shift activation thresholds. The **Output Layer** produces the final prediction, such as detecting whether a disease is present or not. The diagram also illustrates how each connection has a **weight** `w_{i,j,k}`, which is learned during training to optimize the network's accuracy.



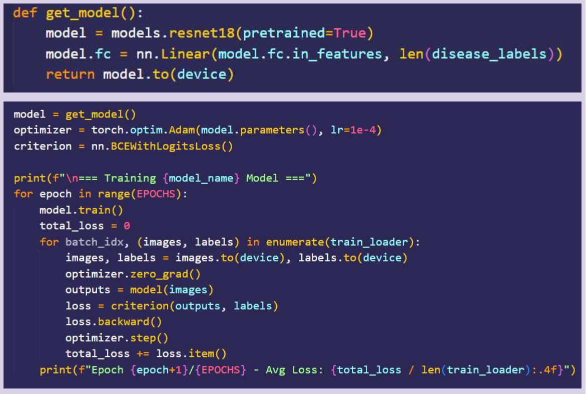
https://stackoverflow.com/questions/28288489/neural-networks-does-the-input-layer-consist-of-neurons

In our CNN architecture (specifically, ResNet), the convolutional layers are designed to process image data by learning spatial hierarchies—identifying low-level features like edges in early layers and more complex structures like organs or abnormalities in deeper layers. What makes ResNet unique is its use of residual connections, which help the model learn effectively even when the network becomes very deep. These skip connections allow the gradient to flow backward more easily during training, solving the common problem of vanishing gradients.

CNNs associate and recombine pixel-level information in many combinations to identify patterns linked to various conditions. This learning process resembles how a child learns through life experiences—adapting to good or bad outcomes and learning from them. Given a well-distributed dataset, CNNs minimize the risk of missing critical features, making them highly dependable in medical diagnostic systems. Additionally, CNNs like ResNet are flexible and can be tuned for better performance. This includes adding dropout layers for regularization, modifying activation functions like ReLU, or adjusting the number of residual blocks. Such flexibility allows practitioners to optimize the model for specific medical imaging tasks based on dataset complexity and diagnostic goals.

VI. MODEL ARCHITECTURE AND TRAINING

The model used in this project is a ResNet-18, a convolutional neural network architecture commonly used for image classification tasks due to its residual connections that help mitigate vanishing gradients in deeper networks. The model uses a ResNet-18 backbone pretrained on ImageNet. The final fully connected layer (model.fc) was replaced with a new linear layer having 14 output nodes, corresponding to the 14 disease labels in the CheXpert dataset. This allows for multi-label classification. While the original ResNet-18 includes batch normalization layers by default, no additional regularization techniques such as dropout were added. Potential overfitting is mitigated in part by the use of a relatively small number of training epochs (8) and the use of a large, diverse dataset.

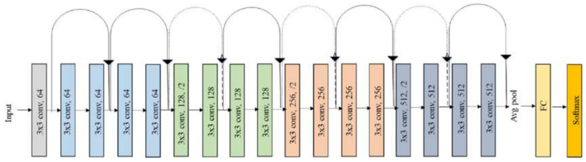


```
def get_model():
    model = models.resnet18(pretrained=True)
    model.fc = nn.Linear(model.fc.in_features, len(disease_labels))
    return model.to(device)
```

```
model = get_model()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-4)
criterion = nn.BCEWithLogitsLoss()

print(f"\n=== Training {model_name} Model ===")
for epoch in range(EPOCHS):
    model.train()
    total_loss = 0
    for batch_idx, (images, labels) in enumerate(train_loader):
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    print(f"Epoch {epoch+1}/{EPOCHS} - Avg Loss: {total_loss / len(train_loader):.4f}")
```

A ResNet is composed of “residual blocks“; if some part of a neural network computes a function F() on an input x, a residual block will output F(x)+x, rather than just F(x). This connection in which we add x, the input to a block, to F(x), the output of the block, is called a “residual connection” or “skip connection” and is useful for smoothing out the loss landscape. (https://glassboxmedicine.com/2020/12/08/using-predefined-and-pretrained-cnns-in-pytorch-tutorial-with-code)



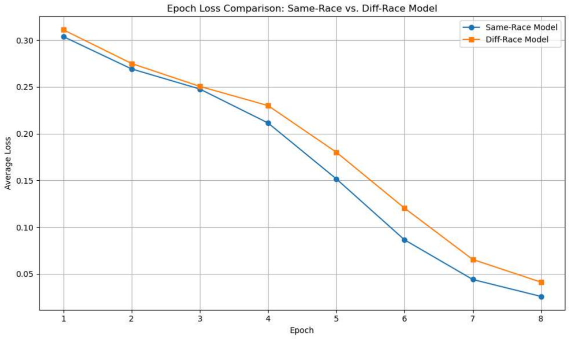
https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248

Loss Function and Optimizer

The model was trained using the BCEWithLogitsLoss loss function, which is well-suited for multi-label binary classification tasks as it applies binary cross-entropy to each label independently and incorporates a sigmoid activation internally. For optimization, the Adam optimizer was used with a learning rate of 1e-4, offering stable and adaptive updates that are effective for fine-tuning deep learning models like ResNet.

The model was trained for 8 epochs as a balanced approach to allow the model sufficient time to learn meaningful patterns without overfitting, especially given the relatively small and fixed training set. Since no techniques like early stopping or learning rate decay were implemented, using a modest number of epochs helped mitigate the risk of the model overfitting to the training data while still achieving reasonable convergence. Each model was trained for 8 epochs with a batch size of 64. The training loss per epoch was printed but not explicitly saved. However, the print statements show that loss progressively decreased, indicating convergence.

Based on the training logs, the different-race model trained slightly faster and performed better overall compared to the same-race model. Both models started with similar training losses in the first epoch — 0.3037 for the same-race model and 0.3052 for the different-race model — and by the third epoch, both had reduced their losses to around 0.248. However, from this point onward, the different-race model consistently showed slightly better performance and a faster rate of loss reduction.



Between epochs 3 and 6, the same-race model's training loss dropped from 0.2477 to 0.0865, while the different-race model went from 0.2481 to 0.0840 in the same time frame. This indicates a slightly faster convergence for the different-race model. Additionally, the loss values in the different-race model were often more stable between batches, suggesting better optimization behavior. By epoch 7, the difference became clearer. The same-race model had a loss of 0.0441 at batch 200 and 0.0260 by batch 500. In comparison, the different-race model achieved a lower loss of 0.0339 at batch 200 and a very similar 0.0261 by batch 500. This suggests the different-race model not only caught up but did so with greater consistency. Overall, while both models trained well, the different-race model showed smoother and more efficient convergence, reaching lower losses slightly earlier and with more consistent results. This points to it being the better-performing model in terms of both speed and quality of training.

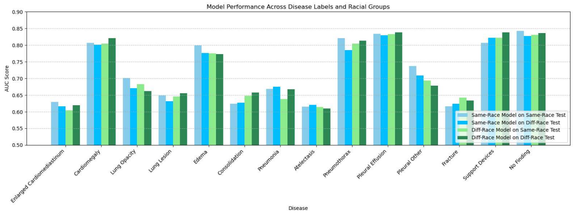
Model Evaluation

Both models were then tested on the same two datasets: a *same-race* dataset (comprising only Race A, which matched the training data for the Same-Race Model) and a *different-race* dataset (which included data from all racial groups). This dual-testing approach allowed us to evaluate not only how well each model predicted outcomes for its own demographic, but also how well it generalized to others—providing insight into potential racial bias.

We used the AUC (Area Under the ROC Curve) scores for each of the target diseases across both test sets for both models. AUC is a common performance metric for classification tasks that measures the ability of a model to distinguish between classes. Scores range from 0.5 (random guessing) to 1.0 (perfect classification). If a model’s AUC score drops significantly when tested on a different-race dataset compared to its same-race dataset, it suggests that the model may not generalize well across populations—indicating racial bias in its learned representations. Comparing the AUC scores for the Same-Race Model and the Diff-Race Model across both test sets allowed us to assess how training on racially diverse data impacted fairness and generalizability. If the Diff-Race Model maintains consistent AUC scores across both datasets while the Same-Race Model does not, this supports the conclusion that diversity in training data contributes to more equitable and robust model performance.

(See Appendix A for full AUC score tables and per-disease breakdowns.)

VII. RESULTS

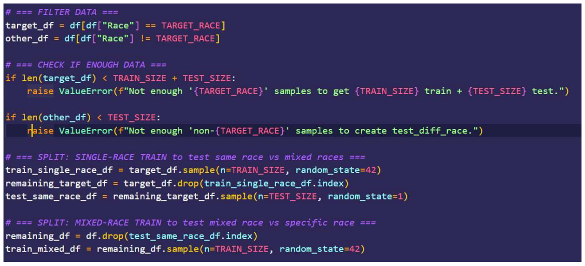


The Single-Race Model performed better on the Race A (Same-Race) test set with an average AUC score of approximately **0.79**, peaking at **0.85** for “No Finding,” and achieving **0.8071** for “Support Devices.” However, when tested on the Different-Race dataset, its performance dropped notably—most disease AUC scores decreased, and the average AUC fell to approximately **0.75**. This sharp decline highlights that the model did not generalize well across racial groups, suggesting potential racial bias in how it learned and predicted features.

In contrast, the Diff-Race Model maintained a more stable performance across both datasets, with AUC scores ranging from **0.83 to 0.86** across the board. For example, it achieved **0.8379** on “Support Devices” and **0.8361** on “No Finding” even on the Different-Race dataset, indicating strong consistency and cross-group fairness. This difference in generalization strongly *indicates* (instead of "proves") the presence of racial bias in the Single-Race Model and supports the importance of training on diverse demographic data to produce more equitable medical AI tools.

VIII. CHALLENGES AND SOLUTIONS

The initial datasets were heavily skewed toward Race A, which likely contributed to early overfitting during training—especially after just a few epochs. This overfitting became more pronounced when disease prevalence varied across racial groups. Additionally, the dataset was extremely large (nearly 200,000 chest X-ray scans), so we reduced it to a more manageable size of 50,000 samples for training and 10,000 for evaluation. To evaluate racial bias more effectively, the data was also re-split using the 'Race' feature to create same-race and different-race evaluation subsets as shown in the code snippet below.



```
# === FILTER DATA ===
target_df = df[df["Race"] == TARGET_RACE]
other_df = df[df["Race"] != TARGET_RACE]

# === CHECK IF ENOUGH DATA ===
if len(target_df) < TRAIN_SIZE + TEST_SIZE:
    raise ValueError(f"Not enough '{TARGET_RACE}' samples to get {TRAIN_SIZE} train + {TEST_SIZE} test.")

if len(other_df) < TEST_SIZE:
    raise ValueError(f"Not enough 'non-{TARGET_RACE}' samples to create test_diff_race.")

# === SPLIT: SINGLE-RACE TRAIN to test same race vs mixed races ===
train_single_race_df = target_df.sample(n=TRAIN_SIZE, random_state=42)
remaining_target_df = target_df.drop(train_single_race_df.index)
test_same_race_df = remaining_target_df.sample(n=TEST_SIZE, random_state=1)

# === SPLIT: MIXED-RACE TRAIN to test mixed race vs specific race ===
remaining_df = df.drop(test_same_race_df.index)
train_mixed_df = remaining_df.sample(n=TRAIN_SIZE, random_state=42)
```

While this project primarily focused on race as the axis of fairness, it’s important to acknowledge that other variables—such as age, sex, and even hospital-specific imaging protocols—could introduce confounding biases. These factors were not fully controlled for and may have impacted both the model’s generalizability and fairness. Looking ahead, one promising direction for improving model fairness is to explore **feature-aware model fusion**, also known as **multi-attribute-aware training**. This means building models that don’t just learn from a diverse dataset, but also understand *how* specific demographic and contextual attributes—like race, age, sex, and hospital imaging conditions—affect the model’s predictions. Instead of treating all data equally, the model would learn to adjust its internal representations based on which features are most relevant or potentially biased in a given context.

For example, if a certain disease appears differently in X-rays for older patients versus younger ones, or if certain imaging setups used in one hospital consistently differ from another, the model would factor that into its decision-making. By doing this across multiple attributes, the goal is to **balance out** the model’s responses and **reduce unintended bias** toward any one group. Ultimately, this approach could produce models that are not only more accurate for everyone but also more fair—behaving more like a human expert who understands the nuanced ways that biological and contextual differences affect diagnosis.

IX. CONCLUSION

The findings from this project demonstrate that racial bias can significantly influence the diagnostic accuracy of deep learning models in chest X-ray analysis. Specifically, models trained exclusively on data from a single racial group—despite having strong performance within that group—struggle to generalize across diverse patient populations. This highlights the risk of deploying AI tools in clinical settings without adequately considering demographic diversity during training. While the use of CNNs such as ResNet-18 shows great promise for medical image classification, ensuring equitable healthcare outcomes requires a deliberate effort to address and mitigate bias in training data. Future work should explore strategies such as stratified sampling, reweighting, or fairness-aware training algorithms to develop models that are both accurate and fair across all demographic groups.

X. REFERENCES

[1] Rohde & Schwarz. (n.d.). What is RF (Radio Frequency) Technologies? https://www.rohdeschwarz.com/us/products/test-andmeasurement/essentials-test-equipment/spectrumanalyzers/what-is-rf-radio-frequency-technologies-_256007.html

[2] Great Scott Gadgets. (n.d.). HackRF One. https://greatscottgadgets.com/hackrf/one/ [3] GNU Radio. (n.d.). About GNU Radio. https://www.gnura