Racial Bias in Computer-Aided Diagnosis

Ketan Joshi, Department of Computer Science
Research Partners: Michael Tang, Chris Le, Trevor Smith
Mentor: Dr. Amar Raheja
CPP Student RSCA Conference 2023

Abstract: Racial bias being prevalent in computer-aided diagnostic systems can have drastic consequences resulting in potential misdiagnosis. Image-detecting machine learning models are starting to be integrated into these systems, but they must be cognizant of the training process and dataset used. Because models rely heavily on the dataset they are trained on, analyzing and engineering the data needs to consider making sure the data is not one-sided or biased towards one particular race due to minute and biological differences in X-Ray image appearance. Many machine learning models exist depending on the type of problem domain or task being done. In the case of image detection, the DenseNet model proves to be best fitting for the scenario of chest x-rays. The objective of this experiment is to observe any differences in model accuracy based on the training scenario being applied, such as training models with datasets primarily consisting of one particular race as well as a dataset being of equal proportion and seeing how it performs on a diverse dataset.

	Scenario Description	Accuracy Results
Scenario 1	Trained with equal amount of data from Whites, Asians, and African Americans and tested with X number of samples of each race and a N number of samples of each disease.	Accuracy Rate ~ 45-53%
Scenario 2	Trained with white patients, as it yields the most data out of all the races. We can test the model with data from each race, with X samples of each disease.	Accuracy Rate ~43%
Scenario 3	Trained with an X (equal) amount of data from all races. We can test the model with N number of samples.	Experiment 1: Accuracy = 66.14% Experiment 2: Accuracy = 57.95%

Diseases Evaluated:

Atelectasis
Cardiomegaly
Pneumonia
Pleural Effusion
Lung Opacity

Races Evaluated:

White
African American
Hispanic
Asian

Model Parameters:

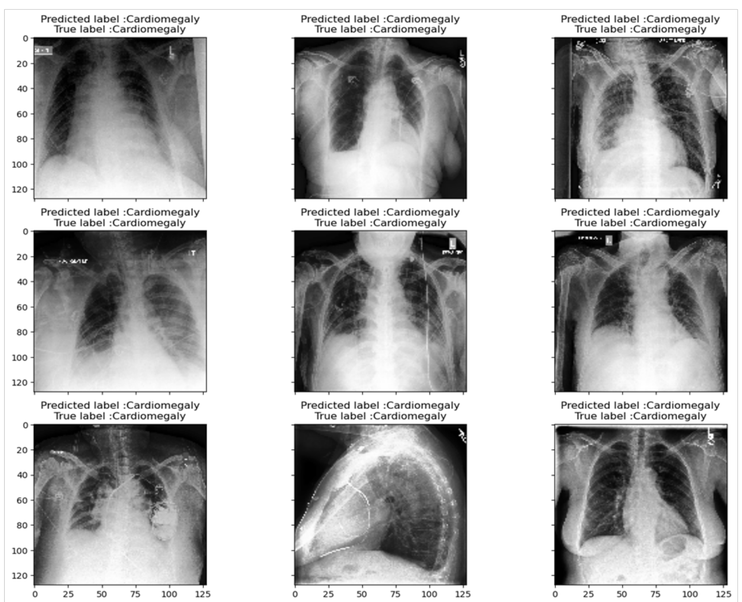
Batch Size = 128
Epochs:
-Scenario 3, Experiment 1 = 50
-Scenario 3, Experiment 2 = 10
Loss Function: Categorical Cross Entropy
Learning Rate: 1e-3

Tech Stack:

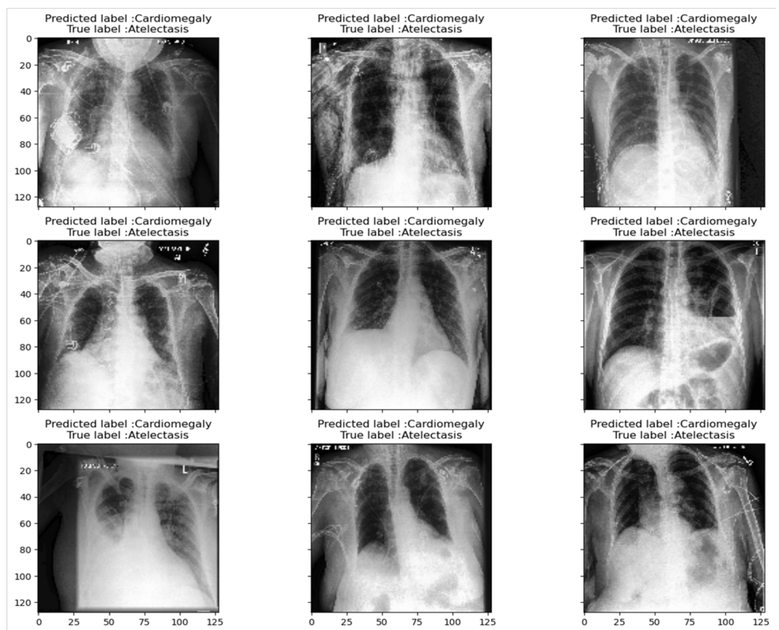
Programming Language: Python
Machine Learning Library: Tensorflow
Image Detection Library: OpenCV/cv2
Neural Network Model: DenseNet121
Data Analysis/Engineering Libraries: Pandas, NumPy

Image Detection Results (Scenario 3):

Correct Results



Incorrect/Ambiguous Results



Implementation:

1. Preprocess the data into readable data using a Pandas Dataframe, with columns for each disease, patient ID, and image path.
2. Turn images into Tensor objects in order to load images in the model using batching
3. Split image dataset according to train and test ratio
4. Configure model and necessary parameters (will have to be revisited after each experiment to fine-tune)
5. Read all the image detection data using the cv2 library
6. Throw in image data into model and evaluate results of image detection testing and accuracy metrics