## 1. Abstract

I worked on replicating the results of the article, “An Adversarial Approach for the Robust Classification of Pneumonia from Chest Radiographs” by Joseph D. Janizek, Gabriel Erion, Alex J. DeGrave, and Su-In Lee. My GitHub link is https://github.com/pratimmoulik-home/Pneumonia-Detection and the link to the video is https://www.youtube.com/watch?v=XeIepO_satI.

## 2. Introduction

### A. Original Paper

Pneumonia is an inflammatory condition in the lung that affects the alveoli (air sacs where oxygen and carbon dioxide is exchanged in breathing). It is often caused by infection with viruses or bacteria. Pneumonia affects 450 million people globally, and there are 4 million deaths each year. It is a major cause of death amongst very young people and very old people. These days, there are many treatments for pneumonia including antibiotics.

There are many different symptoms of pneumonia which are coughs, shaking chills, fever, shortness of breath, etc. These days, pneumonia is often diagnosed using Chest X-Rays. From a Chest X-Ray, it can take a few hours to a couple of days for doctors to diagnose pneumonia. Chest X-Rays can reveal several symptoms of pneumonia such as lung opacity. Lung opacity is when lungs appear white, which means there is fluid and other inflammation in the air sacs or the air sacs appear dense and firm, which means they are not functioning. An increase in lung opacity can potentially mean that there is a higher chance of pneumonia, whereas for a healthy lung, the lung opacity is low and the X-Ray image comes out as dark. However, potential disadvantages to Chest X-Ray pneumonia detection include the fact that pneumonia symptoms can be subtle. Advances in machine learning as well as artificial intelligence are making pneumonia detection from Chest X-Ray images more accurate. Most of the models for pneumonia detection are convolutional neural networks.

Despite the promises of using machine learning to detect pneumonia from patient X-ray images, there are several limitations. There are many different ways and different angles and views of taking the X-ray image, such as anterior posterior(X-ray taken from the front of the patient), as well as posterior anterior (X-ray taken from the back of the patient). Different kinds of X-ray images have to be taken based on the patient (Janizek et al. 2020). For example, bedridden patients can only have X-rays taken from the front of the body (Anterior-Posterior) rather than from the back (Posterior-Interior). Each hospital also has a different way of taking X-ray images (Janizek et al. 2020). Most of the pneumonia detection models only work for specific image views or angles and do not work accurately in different angles or views. The whole goal of the paper is to come up with an algorithm that can accurately detect pneumonia without regards to the way the X-ray image was taken.

To address this problem, the researchers propose an alternative to the traditional convolutional neural network. Their approach involves two neural networks working in opposition. The first is a convolutional neural network (CNN) that predicts whether a patient has pneumonia from the

X-ray image. The second, called the adversary, attempts to predict the view of the X-ray (anterior-posterior or posterior-anterior) based on the pneumonia prediction for a specific patient (Janizek et al. 2020). In this article, anterior-posterior is labelled as AP and posterior-anterior is labelled as PA. The CNN is trained to minimize its own classification loss while simultaneously penalizing the adversary's ability to correctly predict the view. As a result, the model learns to detect pneumonia independently of the imaging view, leading to better generalization across different X-ray views.

### B. Reproducibility

While I was unable to exactly reproduce the full results of the original study, I was able to arrive at similar conclusions in my own work. The original paper utilized the CheXpert dataset for training — a large-scale collection of chest X-ray images and associated diagnoses, including pneumonia, released by Stanford (Janizek et al., 2020). For evaluation, the authors used the MIMIC-CXR dataset, another extensive repository of chest radiographs and clinical labels, released by MIT. Both datasets exceed 400 GB in size due to their high-resolution imaging, making them resource-intensive to download and process.

Due to time and storage constraints, I used a publicly available, lower-resolution version of the CheXpert dataset hosted on Kaggle, which is approximately 11 GB in size. I was unable to locate a similar open-source alternative for the MIMIC-CXR dataset. Consequently, I divided the Kaggle CheXpert dataset into separate training and testing subsets for my experiments.

Despite these adjustments, I successfully reproduced the main model architectures presented in the original paper. To ensure consistency and reproducibility in my results, I set a fixed random seed (SEED = 42) prior to model initialization in each training run.

## 3. METHODOLOGY

### A. Environment

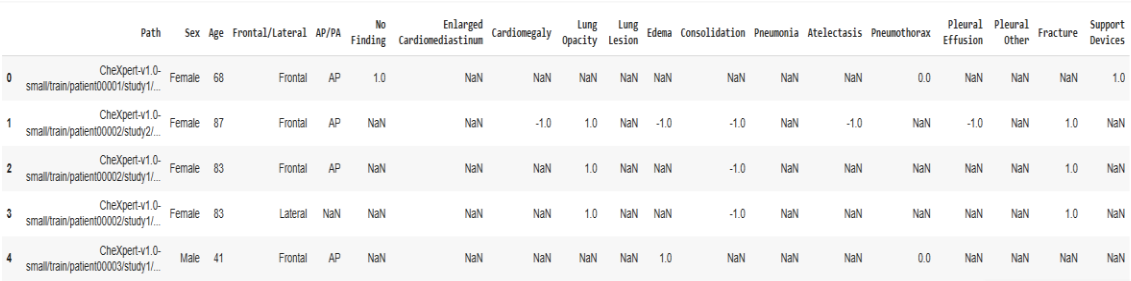
For my code reproduction, I had to take into consideration the Python environments as well as the libraries. I am running it on Python 3.11.2 and its on Google Colab. Kagglehub had to be imported to download the file repository from Kaggle. Pandas and matplotlib were also imported. The code used torch and torchvision modules too. The PIL library was imported in order to process images and sklearn was imported for metrics.

### B. Data

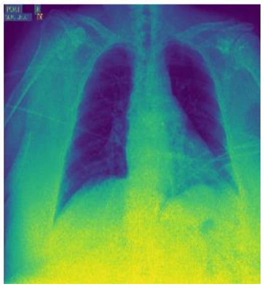
The dataset used in this project is sourced from the CheXpert database, as previously described. The original CheXpert dataset includes high-resolution chest X-ray images and requires approximately 400 GB of storage, making it impractical to use without access to large-scale infrastructure such as an Azure Virtual Machine. To overcome this limitation, I utilized a publicly available, lower-resolution version of the CheXpert dataset hosted on Kaggle, which is under 20 GB in size. The dataset was downloaded using the following command: "kagglehub.dataset_download("ashley/chexpert").

All images were preprocessed by resizing them to a resolution of 224×224 pixels to ensure compatibility with standard deep learning models.

The dataset includes a CSV file containing metadata and diagnostic labels for each patient's X-ray image. Each row in the CSV includes the image path, patient sex and age, image type (frontal or lateral), view position (AP or PA), and diagnostic labels indicating the presence or absence of various conditions such as pneumonia. Below is an example row from the CSV along with its corresponding image.



| | Path | Sex | Age | Frontal/Lateral | AP/PA | No Finding | Enlarged Cardiomediastinum | Cardiomegaly | Lung Opacity | Lung Lesion | Edema | Consolidation | Pneumonia | Atelectasis | Pneumothorax | Pleural Effusion | Pleural Other | Fracture | Support Devices |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | CheXpert-v1.0-small/train/patient00001/study1/... | Female | 68 | Frontal | AP | 1.0 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 0.0 | NaN | NaN | NaN | 1.0 |
| 1 | CheXpert-v1.0-small/train/patient00002/study2/... | Female | 87 | Frontal | AP | NaN | NaN | -1.0 | 1.0 | NaN | -1.0 | -1.0 | NaN | -1.0 | NaN | -1.0 | NaN | 1.0 | NaN |
| 2 | CheXpert-v1.0-small/train/patient00002/study1/... | Female | 83 | Frontal | AP | NaN | NaN | NaN | 1.0 | NaN | NaN | -1.0 | NaN | NaN | NaN | NaN | NaN | 1.0 | NaN |
| 3 | CheXpert-v1.0-small/train/patient00002/study1/... | Female | 83 | Lateral | NaN | NaN | NaN | NaN | 1.0 | NaN | NaN | -1.0 | NaN | NaN | NaN | NaN | NaN | 1.0 | NaN |
| 4 | CheXpert-v1.0-small/train/patient00003/study1/... | Male | 41 | Frontal | AP | NaN | NaN | NaN | NaN | NaN | 1.0 | NaN | NaN | NaN | 0.0 | NaN | NaN | NaN | NaN |



For each patient in the dataset, the corresponding chest X-ray image was linked to its file path, resulting in a final dataset where each entry contains both the image and its associated metadata. To maintain consistency, all diagnostic labels were standardized to binary values (0s and 1s). Each row in the processed dataset includes the X-ray image path, diagnostic labels (including pneumonia), and view type information — indicating whether the image was taken in an anterior-posterior (AP) or posterior-anterior (PA) orientation.

## C. Model

The model architecture comprises two interconnected neural networks: a DenseNet-121 convolutional neural network (referred to as the **CXRClassifier**) and an **Adversary** network (Janziek et al. 2020). The model code is found in the link, https://github.com/suinleelab/cxr_adv/. I could not run the whole code since the dataset would have to be saved locally to run and the model code was for the original CheXpert and MIMIC-CXR database, which were over 400 GB long each, and would take a long time to train. DenseNet-121 is a deep convolutional model tailored for image classification tasks and is available in the PyTorch library. DenseNet-121 is different from other convolutional neural networks in that each layer receives input from all previous layers and passes its own output to all subsequent layers within a block. The CXRClassifier takes chest X-ray images as input and generates predictions across 14 diagnostic categories: *No Finding, Enlarged Cardiomediastinum, Cardiomegaly, Lung Opacity, Lung Lesion, Edema, Consolidation, Pneumonia, Atelectasis, Pneumothorax, Pleural Effusion, Pleural Other, Fracture,* and *Support Devices*. In parallel, the Adversary network receives the pneumonia prediction from the classifier and attempts to infer the X-ray view — either anterior-posterior (AP) or posterior-anterior (PA) (Janziek et al. 2020).

The primary objective was to train the CXRClassifier to accurately detect pneumonia while minimizing its dependence on view-specific features, thereby achieving view-invariant predictions. To accomplish this, training was carried out in two phases. In the first phase, the CXRClassifier was trained and evaluated independently to establish a baseline for performance.

In the second phase, the classifier was jointly trained with an Adversary network. During this stage, the classifier was penalized whenever the adversary successfully predicted the X-ray view (AP or PA) based on the pneumonia output, thereby encouraging the classifier to suppress view-related information in its predictions.

To ensure consistent and reproducible results, a random seed (SEED = 42) was set prior to model initialization. Additionally, to avoid fluctuations associated with using pretrained models—such as DenseNet-121 initialized on natural images from the internet—the CXRClassifier was trained from scratch, using only chest X-ray images of pneumonia patients from the dataset.

Due to the large size of the original CheXpert and MIMIC-CXR datasets (over 400 GB), which typically require high-performance infrastructure to handle, I opted to use a publicly available, lower-resolution version of the CheXpert dataset from Kaggle (approximately 11–20 GB). Since no comparable lightweight version of the MIMIC-CXR dataset was available, I partitioned the Kaggle CheXpert dataset into separate training and testing subsets for this project.

### D. Training

The models were trained in Google Colab using NVIDIA A100 GPUs, which are among the fastest available. Compared to T4 and L4 GPUs, which were significantly slower, the A100 allowed each epoch to complete in approximately 30 seconds. Each model was trained for 10 epochs. The core architecture, DenseNet-121, is a well-established convolutional neural network, and its hyperparameters in the CXRClassifier were kept fixed. A batch size of 64 was used to ensure efficient training with a reasonable number of examples per batch.

The primary hyperparameter that required tuning was the loss weighting factor, **λ (lam)**, used during adversarial training. When jointly training the CXRClassifier with the Adversary network, the Adversary's loss is multiplied by λ and subtracted from the CXRClassifier's loss. This setup encourages the classifier to make predictions that are invariant to the X-ray view, as it is rewarded when the Adversary fails. After experimentation, a λ value of 0.1 was selected; values too high or too low led to decreased validation performance compared to training the classifier alone.

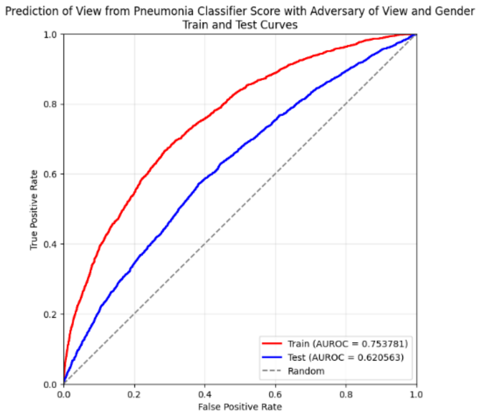
The learning rate was set to a low value of 0.0001 to ensure stable, gradual convergence—a standard practice in training convolutional networks. Binary Cross-Entropy (BCE) loss was used in all scenarios: for the standalone CXRClassifier and for both networks during joint training. In the adversarial setup, the final loss for the classifier was computed as the BCE loss minus λ times the Adversary's BCE loss.

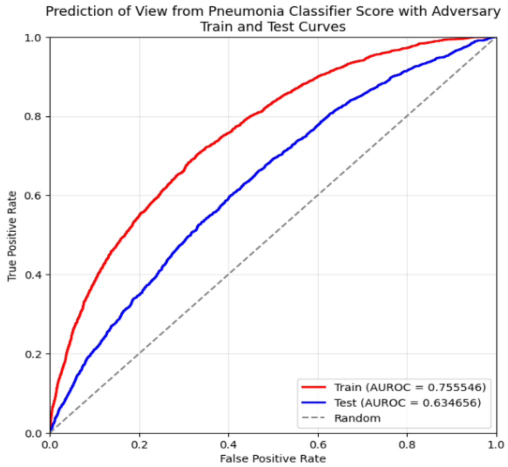
### E. Evaluation

To evaluate performance, the Area Under the Receiver Operating Characteristic Curve (AUROC) was used to compare predicted and actual pneumonia diagnoses(Janziek et al. 2020). AUROC is a robust metric that captures the balance between true positives, false positives, true negatives, and false negatives. In the context of pneumonia detection, it is particularly important to ensure that true cases are correctly identified. The model outputs a probability score for pneumonia, where values closer to 1 indicate higher likelihood of disease.

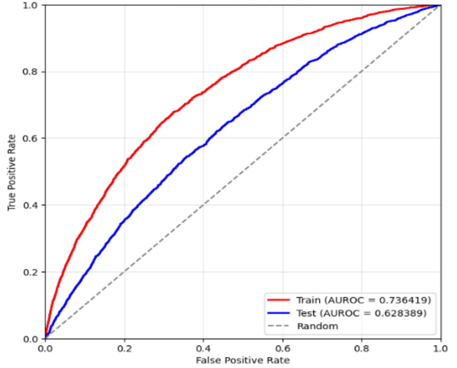
# 4. RESULTS

The results had similar conclusions as the original article, but had its own differences too. For the “Train” and “Test” dataset on my code, the AUROC score was higher both times when the CXR Classifier was jointly trained with the Adversary rather than the CXR Classifier being trained by itself. However, for the article, the AUROC score was lower for the CXR Classifier being trained with the Adversary for the “Train Dataset”. For the article, the AUROC was 0.791 and 0.703 for the train and test for pneumonia classifier without adversary(Janziek et al. 2020). With the adversary, the AUROC was 0.747 and 0.739. The numbers were different because of the different nature of the datasets. The article used very high resolution X-ray images while mine had lower resolution X-ray images. Since the article was able to parse the MIMIC-CXR repository into a proper dataset, the Train and Test results were different from what I have. Here is the AUROC graph for my pneumonia classifier without adversary and the pneumonia classifier with the adversary. It seems from my tests, when training the model with the adversary, the model performs slightly better on pneumonia detection than the model without the adversary which means that the view of the X-ray gets removed as a confounding factor in pneumonia detection.





From the research of this article, there can potentially be several different extensions of the study. This paper mostly focused on the X-ray image view as a confounding variable when detecting pneumonia. However, there are so many other confounding variables such as age, gender, hospital site, etc. One potential thing to do is to train adversaries on those variables too and then jointly train the CXR Classifier with those adversaries so that the variables like age, gender, etc do not become confounding variables when detecting pneumonia. I created a model that added gender as a confounding variable on top of the X-ray view. Training with the two adversaries however caused the pneumonia detection to become less accurate. This could have been a result of the loss function factors or how gender was encoded. Here is a copy of the AUROC results.



# 5. DISCUSSION

The original paper is technically reproducible since the open source code is available on GitHub. The data is accessible, but it is over 400 GB long which means to completely reproduce the paper accurately, the dataset would need to be stored on an Azure Virtual Machine, which can be expensive. It is likely that the authors took much longer to train their datasets since it was being trained on very high resolution images. The code from the GitHub points to the original dataset and as a result, was probably run on the Virtual Machine. My reproduction was simplified due to time constraints. The easy part was that once the paper was read or was interpreted by LLMs, the models themselves are not that complicated. However, the hard part was accessing the datasets. I initially tried to run an Azure Virtual Machine for the 400 GB long CheXpert dataset, but saw that it would not be time efficient, which lead me to find an open source version of the CheXpert dataset from Kaggle that has the same images, but with much lower resolution. For testing, MIMIC-CXR dataset was used by the article and it is also over 400 GB long and I could not find an open source version with lower resolution images of the MIMIC-CXR dataset which is why I had to improvise and divide up the Kaggle dataset into two for training and testing rather than having CheXpert for training and MIMIC-CXR for testing. In order to improve the reproducibility of the article, it might be more beneficial to show on the open source GitHub more of how to run the code. The authors could also have given better instructions on how to download the dataset and store the dataset on a virtual machine since it is very big.

# 6. CONTRIBUTIONS

I am the only team member. I did all of the research, coding, as well as the writeup and video.

**CITATIONS**

Janizek, J.D., Erion, G., DeGrave, A.J., and Lee, S.-I. 2020. An Adversarial Approach for the Robust Classification of Pneumonia from Chest Radiographs. *arXiv preprint arXiv:2001.04051.*