NIH Pneumothorax classification with label noise

Karl Ulbæk s183931

Introduction and setup:

Purpose/goal of this rapport:
Investigate the extent of label noise in the NIH “Pneumothorax” x-ray image classification task, in order to determine if this may be the cause of gender biased predictions.

This investigation will be carried out by implementing a number of different techniques which enables or improves deep learning from data with label noise.

If said implemented techniques improves performance and reduces the gender bias gap this should be a strong indication that the data contains label errors and that this is the cause of the gender performance gap.

Disclaimer:
The report will primarily serve to document my whole process of trying to achieve the above stated goal.
This report will be coarse, informal and stick to the essentials. The theory will be kept to a minimum, equations will be screenshots and the plots and figures will be inconsistent in style and labeling as some of the plots are from early stages of the project/code.

Data scope:
All experiments and all techniques have only been employed and tested on the below specified data.

- NIH
- Pneumothorax (4.7% of samples are of the positive class)
- 50/50 male/female
- 1 sample per patient
- 8458 train samples, 1409 validation samples, 8459 test samples

Base hyperparameters and model:

- pretrained resnet50,
- 80 batch size (just as large as possible)
- AdamW optimizer
- lr of 3e-5 with some decay per epoch using a scheduler

A note on class imbalance:
The dataset contains a very strong class imbalance 1/21 ~ 4.7% positive samples/labels. If this class balance is not taken into account during training the model will either take a long time to converge or will get stuck in a state of all ways predicting the negative class, as this will yield 95.3% accuracy.

In order to combat this I initially used the torch weighted binary cross entropy implementation with the weight for positive labels being 21. However some of the implementations used

down the line in this report involve modifying the loss function and directly weighting the positive samples become less straightforward. I therefore adapted the below technique: which I denote **adaptive class weighting schedule.**
I introduced a scaling parameter “s”, which emphasizes positive samples in the loss function. After each epoch the goal is to have the average predicted class of the model (on the train data) being equal to the class imbalance of the training set. Therefore after each epoch if the average predicted class is too low we increase “s” and if it's too high “s” is decreased.

The baseline:

The baseline uses weighted binary cross entropy and will be referred to as weighted BCE or weighted CE or just CE. The baseline also serves to reproduce Nina's results.
Reproducing Ninas Pneumothorax results corresponds to reproducing what is indicated by the red box in the below plot from her paper.

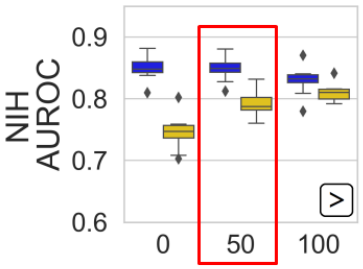

Figure 1: average male AUROC 0.84 and female 0.78 (read of the plot).

Learning curve of my baseline:

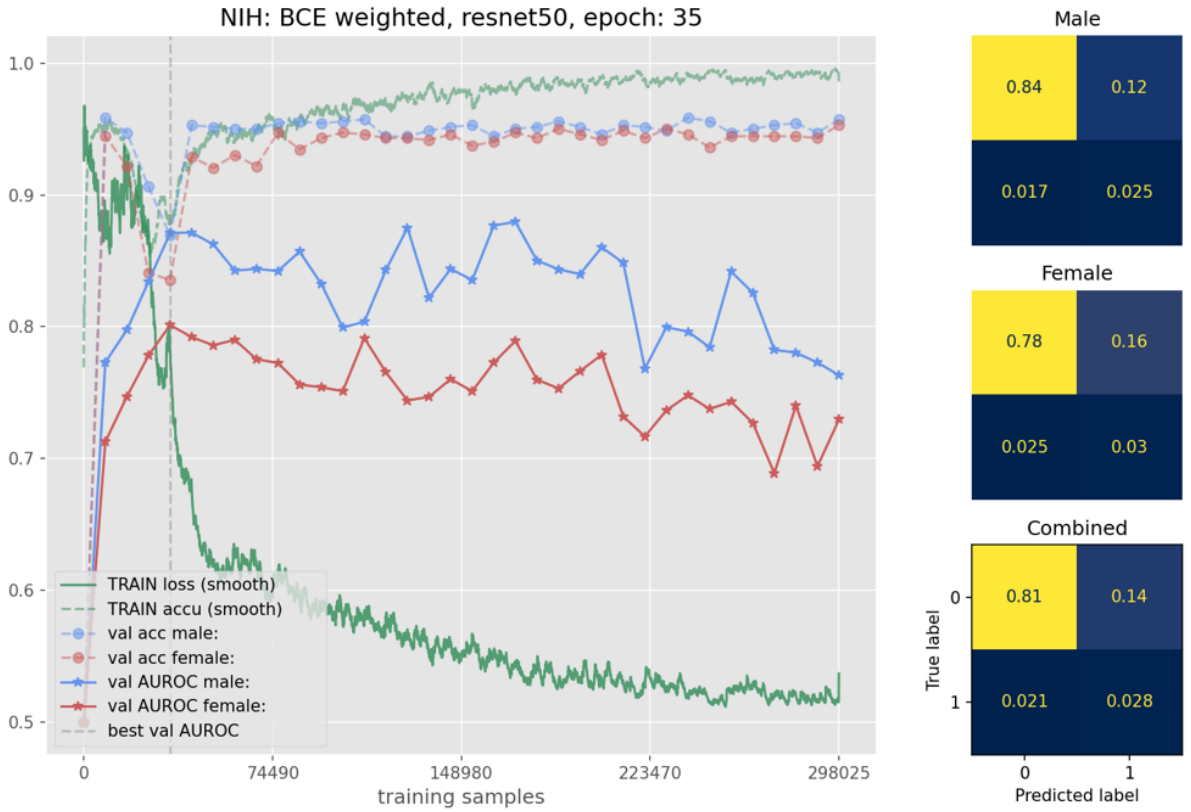

Figure 2: The confusion matrix is in terms of test data using the best model. i.e. the model corresponding to the vertical dashed gray line (best val AUROC). 0.824Male / 0.769Female AUROC on the test set using the best model.

A few notes on the plot in general, which will also be relevant in similar subsequent plots.

- It should be noted that the spikey/unstable loss in the first couple of epochs is caused by the adaptive class weighting schedule having to converge to a good value. This will be a recurring trend in all my learning curve plots.
- Additionally it should be noted that the train loss and accuracies have been smoothed slightly with exponential smoothing. The necessity for the smoothing is due to the fact that the training metrics are evaluated at each batch rather than at each epoch and thus become more spikey.
- The confusion matrix will always be in terms of the test dataset whereas the learning curve graphs will always be in terms of training data and validation data.

The learning curve plot displays clear overfitting for the baseline, which is further emphasized by the fact that the best model (in terms of validation AUROC) is found already after 4 epochs.
It appears that the baseline is able to reproduce Ninas results in terms of performance and relative gender gap.
From the “combined confusion matrix” it can be seen that the model predicts on the positive class 5% of the time which is aligned with the prevalence in the training data, however the model performs essentially random (50/50) given a sample of the positive class.

Mixup: Beyond Empirical Risk Minimization:

The first method I implement is called mixup and is from (https://arxiv.org/abs/1710.09412, from 2018 with 9000 citations today). The method can be described using the following 2 equations:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j,$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j,$$

where x and y are training data and labels and lambda is some number close to 0.5 and is usually drawn from a gamma distribution. The method has the model train on data and labels which are linear combinations of 2 samples and 2 labels. It results in the model having smooth and continuous prediction probability transitions between different classes. This reduces overconfidence and overfitting but also reduces the effect of label noise.

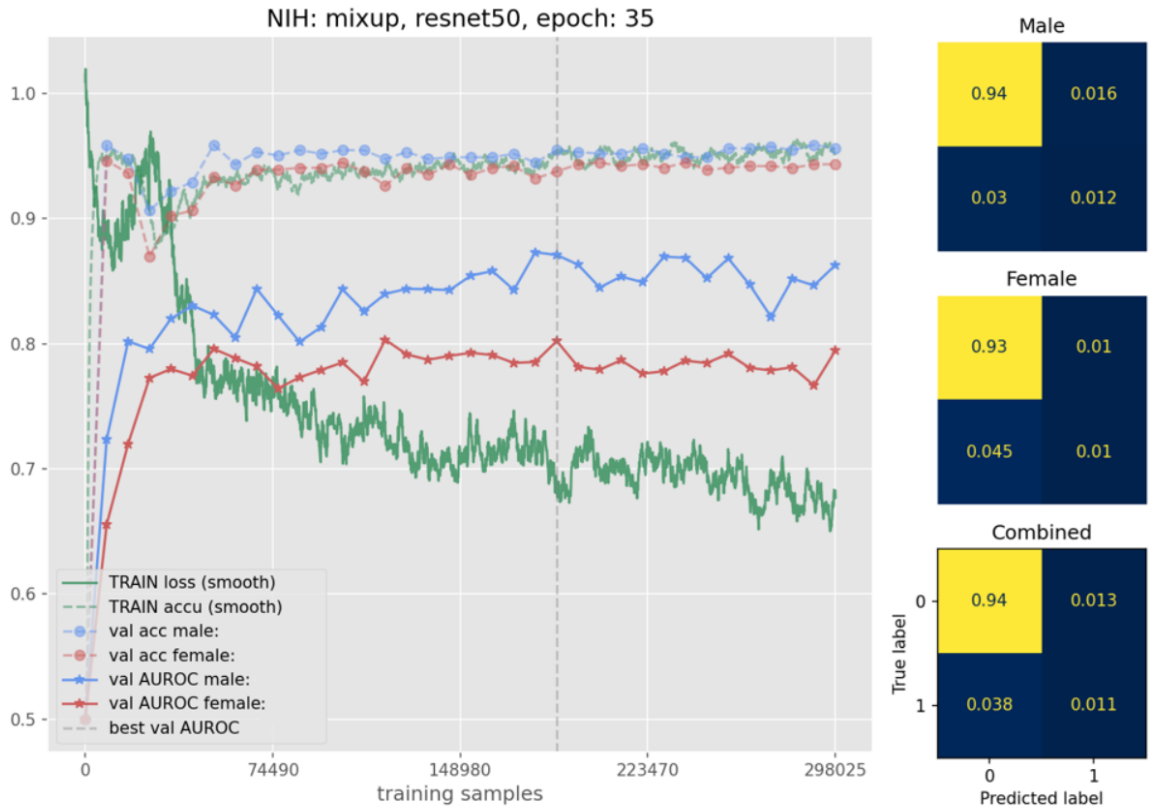

Figure 3: Learning curve of mixup. The confusion matrix is in terms of test data using the best model. i.e. the model corresponding to the vertical dashed gray line (best val AUROC). 0.849Male / 0.806Female AUROC on the test set using the best model.

The model takes longer to converge and arguably has not fully converged after the 35 epochs. The problem of overfitting appears to have vanished, a common and well documented effect of mixup. The overall model performance does not improve and the model still severely struggles with positive samples.

Unsupervised Label Noise Modeling and Loss Correction (GMM)

This approach is from the paper (https://arxiv.org/abs/1904.11238. from 2019 with 500 citations). I will refer to this method as the GMM-approch (gaussian mixture model). The method evolves fitting a 2 mode gaussian mixture model after each epoch to the loss distribution: in an ideal world this would yield the below plot:

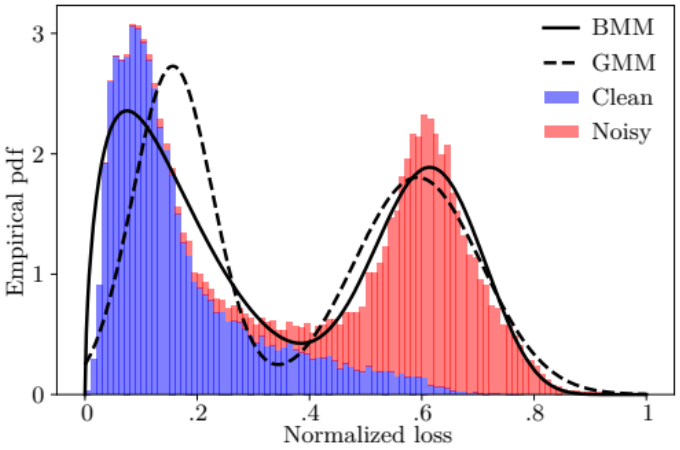

Figure 4: The figure is taken from the original paper and shows an instance of training when the data has 50% label noise.

The notion here being that all losses belonging to the first mode are from data with correct labels and all losses belonging to the 2. mode are from data with incorrect labels. This loss distribution can be modeled with some 2 mode probability distribution. In the paper they experiment with both a beta mixture model and a gaussian mixture model. I only implemented the method using the gaussian mixture model.

The method proceeds as follows.
During all epochs standard unweighted cross entropy is calculated for all samples. Only in the first epoch are these used for the backward pass. After each epoch a GMM is (re)fitted to CE losses of the previous epoch. Then for the subsequent epoch the loss used for the backward pass is calculated as follows:

$$\ell_B = -\sum_{i=1}^{N} \left(\left(1 - w_i\right) y_i + w_i z_i \right)^T \log\left(z_i\right)$$

Where “z” is the predicted label probability and where “w” is the GMMs predicted probability of the standard CE loss of the sample. i.e when w is large the GMM believes the sample belongs to the high loss or noisy mode (red in the figure) and we should thus not trust the given label “y” but rather the “z” which is the model predicted label. “z” may either be a soft label i.e. probability or a hard label i.e. 0, 1. I used soft labels.

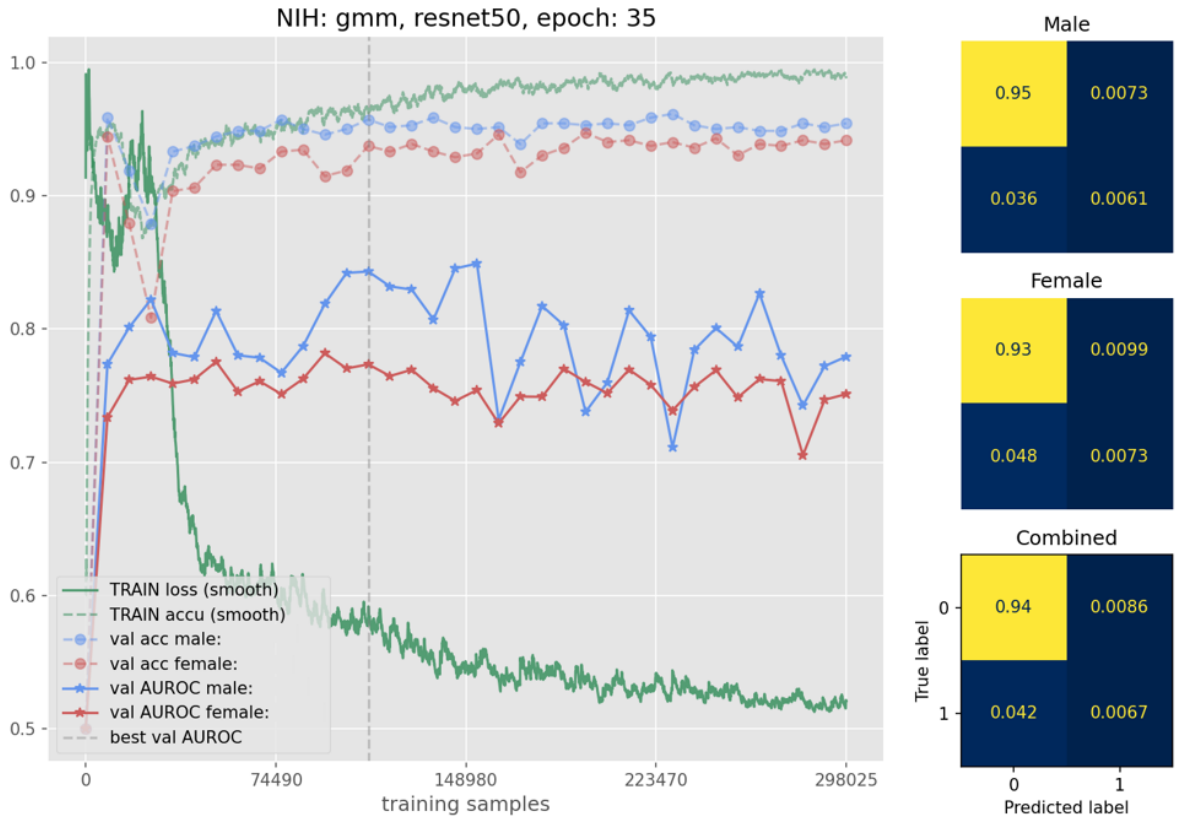

Figure 5: The confusion matrix is in terms of test data using the best model. i.e. the model corresponding to the vertical dashed gray line (best val AUROC). 0.848Male / 0.750Female AUROC on the test set using the best model.

GMM does not appear to improve performance or reduce gender gap. Overfitting emerges again and the model still struggles to perform on positive samples as it barely archives any true positives.

Combining GMM and Mixup:
The GMM paper explains that the above loss correction did as of 2019 when the paper came out not beat current state of the art methods. However when they combined GMM with MIXUP they were able to beat the competition. Whereas the 2 approaches individually don't introduce any sizable computational overhead, combining them however requires one additional forward pass for each batch. One forward pass is performed using the mixed data and one with the unmixed data. The forward pass with the unmixed data is used for fitting the GMM and for predicting the weight “w” used in the loss function.

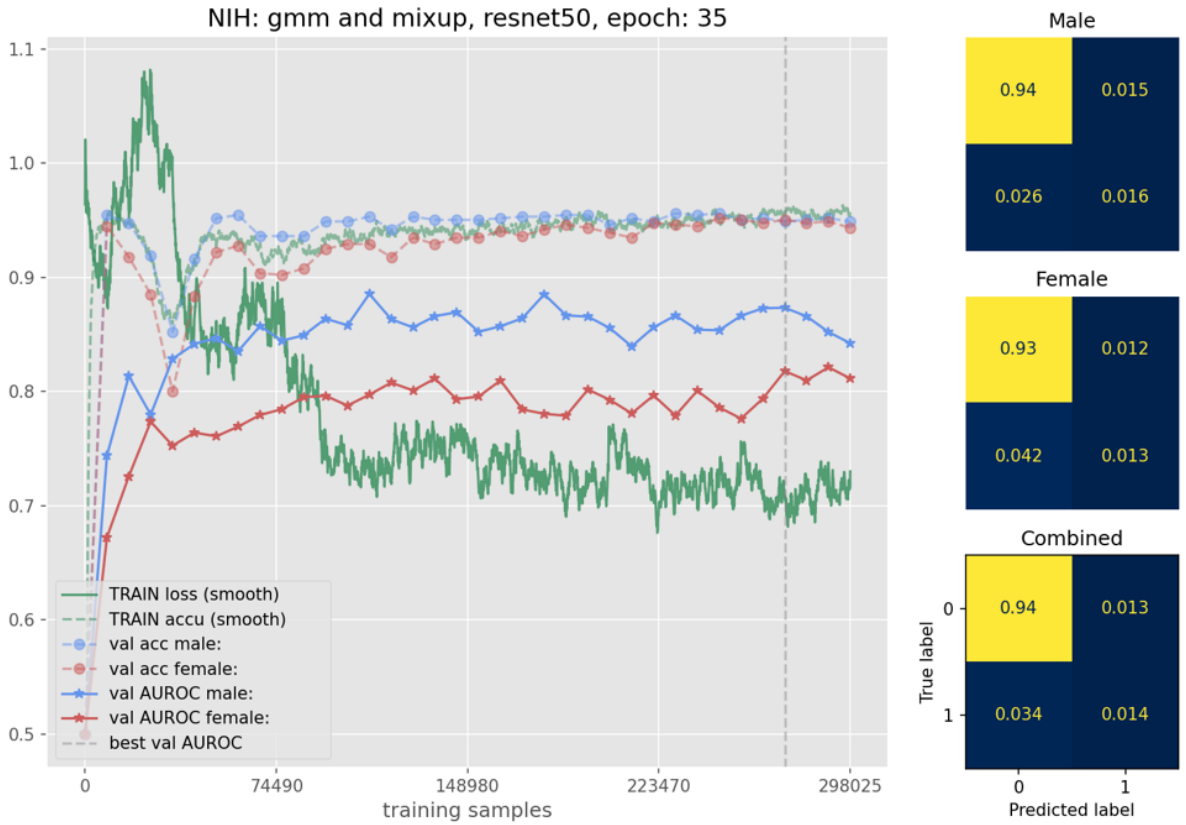

Figure 6: The confusion matrix is in terms of test data using the best model. i.e. the model corresponding to the vertical dashed gray line (best val AUROC). 0.839Male / 0.815Female AUROC on the test set using the best model.

It appears that Mixup helps regularize GMM but the inherent issues are still present and the model does not improve compared to plain mixup.

To investigate if GMM behaves meaningfully and solves the problem it is intended to, I plotted the loss distribution of the training data of the best model.

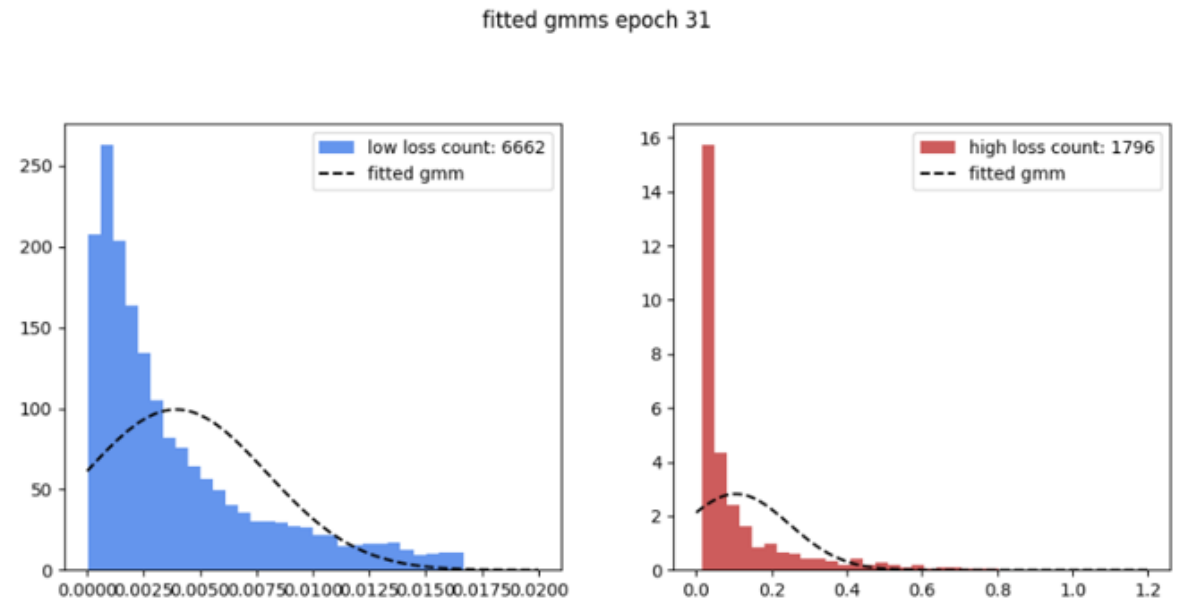

Figure 6: The loss distribution of the training data and the fitted GMM. Due to the scale of the 2 modes of the fitted normal distributions the modes had to be plotted in separate windows. The reader should pay close attention to the scales of the axis of the 2 plots.

The true empirical loss distribution (i.e. the bins of the histogram) emerges not as a 2 mode gaussian but rather as a single ordinary exponential distribution (pay close attention to the axis scales), and the idea of the GMM approach falls short for obvious reasons.

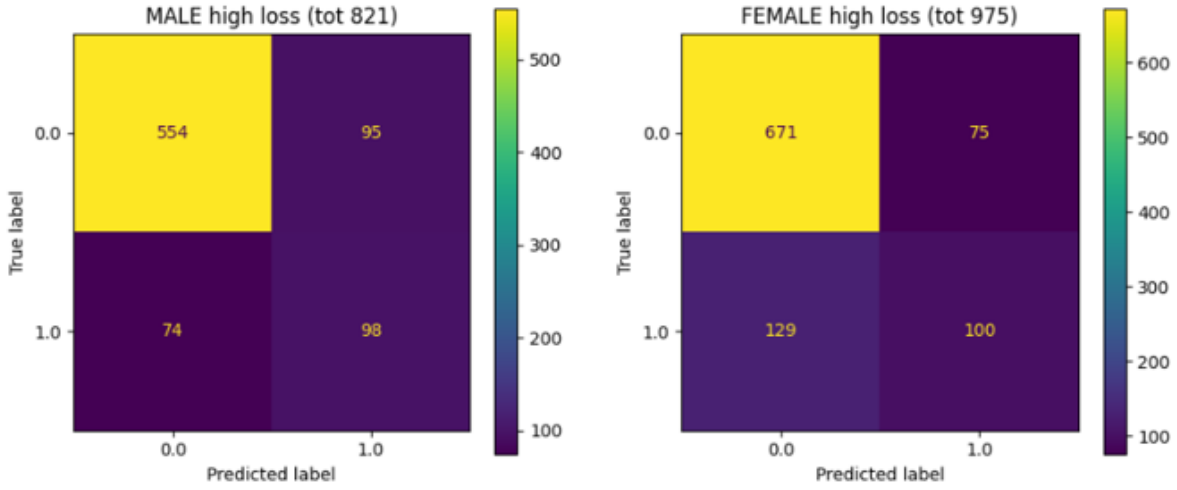

Figure 7: The model performance on the training samples belonging to the “high loss” group.

The high loss group contains considerably more female training data, yet again highlighting the underlying problem of the model struggling with female data. Additionally it should be noticed that the high loss group contains a relatively greater concentration of true samples of 20% compared to true concentration of 5% present in the whole training set, highlighting the models inability to fit the positive samples.

Confident Learning (CL)

The method is from the paper “Confident Learning: Estimating Uncertainty in Dataset Labels” (https://arxiv.org/abs/1911.00068 from 2021 with 500 citations as of today).

Whereas the previous two methods alter the training procedure CL aims to remove noisy data and retrain the model on the now (hopefully) clean data. Because of this approach CL is essentially model and method independent as it works for any model which outputs a predictive probability distribution over labels.

In practice CL is performed as follows. For your training dataset perform a k-fold cross validation (I used k=5). Train the model for all splits and predict the holdout part of the split. This way, out-of-trainingset-predictions are archived for your entire train dataset.

These predictions are used to estimate the model self confidence of each class:

$$t_j = \frac{1}{|\boldsymbol{X}_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta})$$

In words, the self confidence “t” of class j is calculated as: For each sample with dataset label “y=j”, what is the model's probability prediction of that sample being in class “j”. Averaged across all samples of class “j”.

We now introduce the counting matrix/table C.

C	y*=j	y*=i
y^=j		
y^=i		

Where “y*” are the true underlying label and “y^” is the noisy label given in the dataset. C is constructed using the following formula.

$$\hat{\boldsymbol{X}}^{(\text{simple})}_{\tilde{y}=i, y^*=j} = \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \ \hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta}) \geq t_j\}$$

For each sample of observed class “y^=i”, if said sample has predicted probability of class “j” greater than the models self confidence of class “j” then we assume that the true label should have been “y*=j” and we consider the sample noisy/incorrect and add a count to the matrix C at the corresponding index. If a sample does not violate any of the class self confidences the count is added to the diagonal of the matrix at the corresponding indices.
If all the data is clean we would expect the matrix C to be a diagonal matrix. Finally, all samples not in the diagonal would then be removed (or have their label changed) for the retraining as these are considered faulty.
The authors of CL state that the diagonal element of each row should be the greatest value in said row else the theory breaks down as the model/data is invalid. Why this is important in my case will be evident momentarily.

My experience performing CL.
Initially, I did the K-fold training using standard weighted cross entropy loss but this yielded very overconfident predictions resulting in more extreme self confidence scores of the model. According to the CL paper the CL approach should handle overconfidence just fine, however I ended up using MIXUP for the k-fold training as the results seemed more trustworthy.

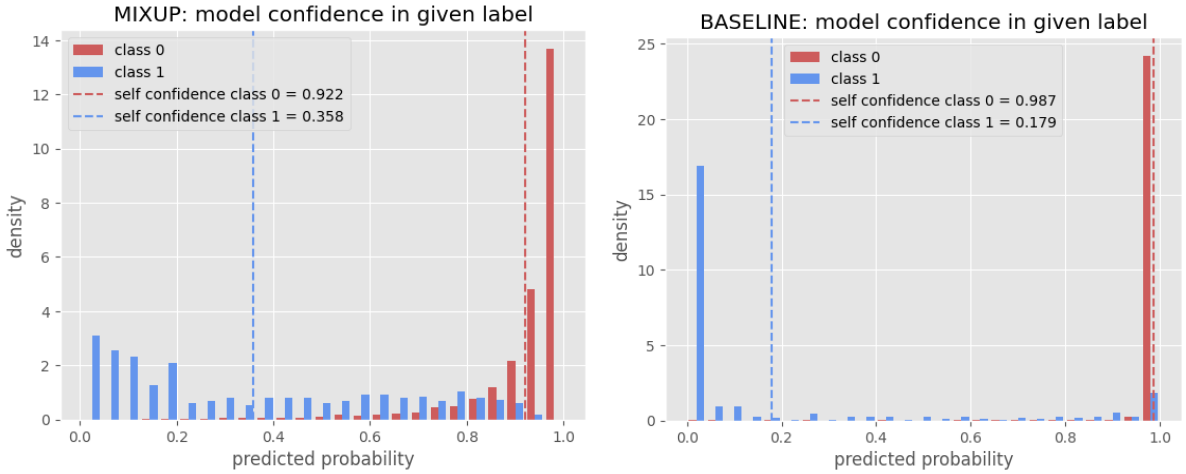

Figure 8: comparison when using MIXUP versus BASELINE. In the case of mixup the predicted probabilities are less extreme/overconfident compared to the CE baseline.

Male and female self confidence can also be calculated individually:
MALE self confidence for class=0: 0.929
MALE self confidence for class=1: 0.399
FEMALE self confidence for class=0: 0.914
FEMALE self confidence for class=1: 0.326

This enforces the notation that the model is more confident in males and more confident in class 0. Gender specific self confidence is not used in any subsequent calculations.

Computing the C matrix using the combined (gender independent) self confidence values yields:
C = [7973 409]
[73 3]

However as stated the diagonal elements should be the greatest element in each row. This is not the case, and if I were to clean my data according to C i would essentially remove all class=1 samples from my data. That the diagonal elements are not dominating my C matrix gives evidence to a fundamental problem with the data and/or model.

I still wanted to try the method and thus did something not mentioned/proposed in the paper.

I introduced a scaling parameter “s” to reduce self confidence. I manually tweaked “s” untill exactly 1% of the counts appeared in the off diagonals, which would correspond to removing 1% of the data:

s=0.085

combined s-scaled C:
[[8046 85]
[0 327]

We can mask the data according to gender to see the gender specific distribution of labels with noise.

male s-scaled C:
[[4051 28]
[0 150]

female s-scaled C:
[[3995 57]
[0 177]

We see that according to the CL-approch there are more female than male labels which are incorrect/noisy.

Before showing the results for retraining the model on the data when cleaned according to the above found s-scaled C matrix i want to do a quick sidestep.

Garbage in garbage out.
As stated the non s-scaled C matrix points to an inherent problem in the data/model/problem. The C matrix is not the first indicator hereoff and the most obvious evidence was in the confusion matrices of the test predictions of all previous attempted methods.
The problem being that, the resnet50 is not able to classify the class=1. For all test samples of class = 1, the predictions are essentially random. Given the data the model is not able to make meaningful predictions.

This is a strong indicator that the problem of predicting/classifying Pneumothorax based on x-rays is not feasible. To rule out that the poor performance of this classification problem was either caused by too little data or too simple and/or incapable model architecture, I tried 2 things.

Increase the complexity of the model. Rather than using a resnet50 I instead used a ConvNext model (https://arxiv.org/pdf/2201.03545.pdf from 2022 with 2700 citations as of today). ConvNext is a SOTA convolutional based network architecture. The specific model I used has 126M parameters compared to resnet50s 26M. Additionally it was pretrained on

imagenet 22k rather than imagenet 1k as the resnet50 in question.

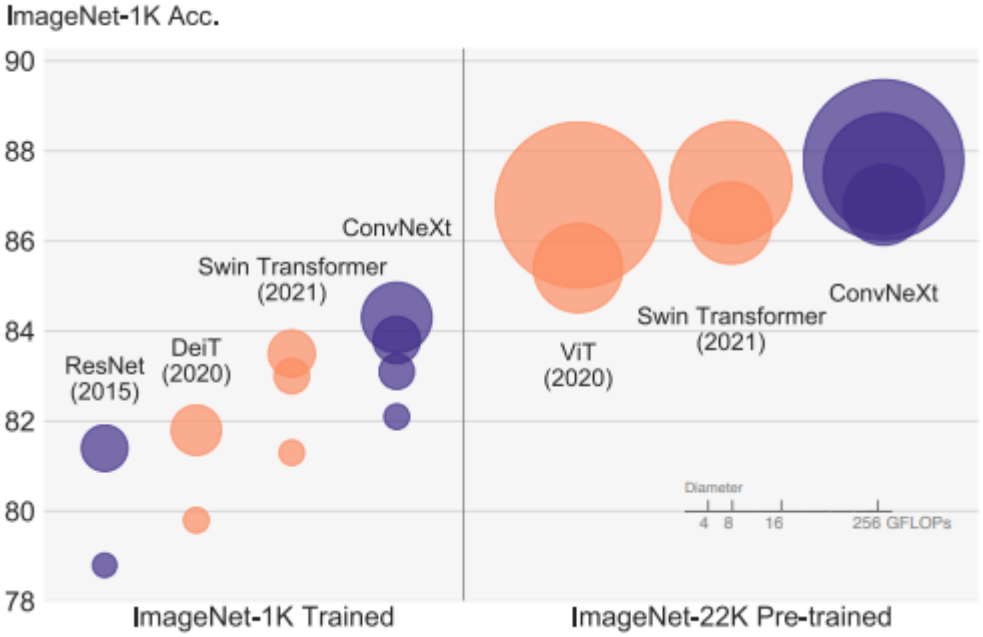

Figure 9. Performance plot of convnext taken from the paper where it is proposed. From another course I have personal experience with ConvNext being a very capable vision model.

In short i employed a model with a SOTA architecture, 5 times the parameters and better pretraining, thus i would expect to see better performance for the problem at hand. The results did not improve.

Increase the amount of training data.
To see if it was a matter of insufficient training data I trained a ConvNext model on the combined test and training data set. i.e. on twice the amount of data than what is used in any of my other training runs. Doubling the amount of training data did not lead to any performance improvements.

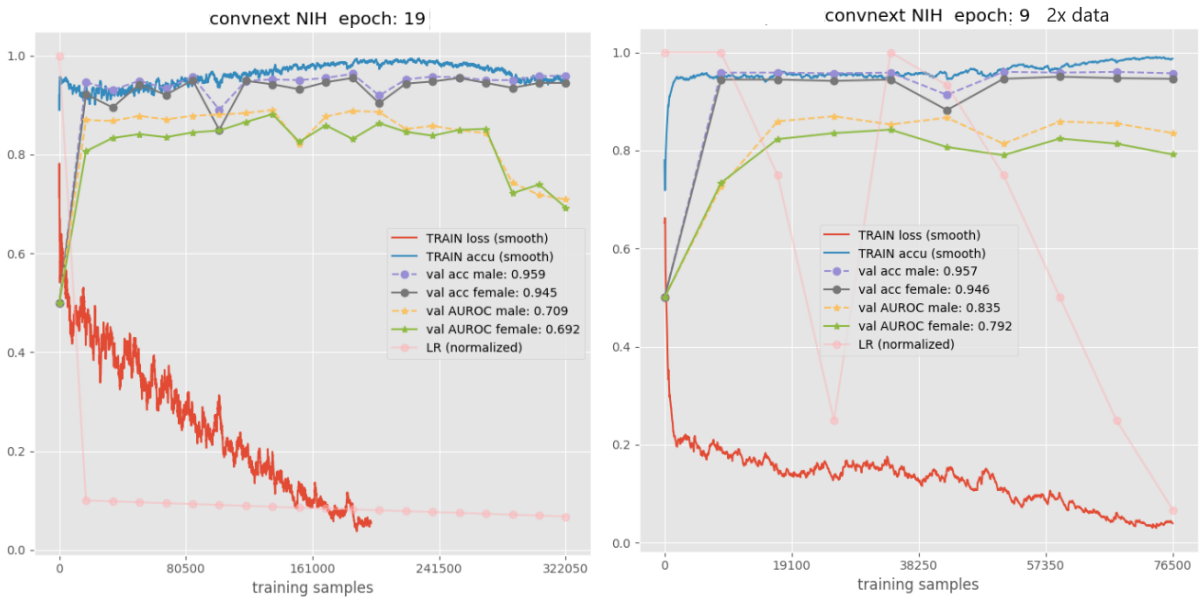

Figure 10: Left: Learning curves of ConvNext. Right: Learning curves of ConvNext using 2x data.

The figures have a lot of shortcomings but convnext is a heavy model to train and I only bothered to run the experiments ones.The important information is still there in the plot, so please bear with me and look past the following:

- ACC and AUROC in the legends are in terms of the last completed epoch, so don't mind these as they have overfitted in the left plot.
- On the left plot the training loss got corrupted with a NAN so matplotlib refused to keep plotting it after epoch 12, don't mind this as the model is obviously still learning which can be seen from the training acc i.e. the blue line.
- The 2 training runs had 2 different learning rate schedules but this did not affect performance.

The only thing I want the reader to pay attention to in the 2 plots are the green and the yellow lines. These are the male and female validation AUROCs. These have converged and more importantly these have converged to the same values as essentially all the previous trained models.

Neither using a vastly superior model nor doubling the amount of data (or both) improves performance and thus I think it's fair to say that the problem was infeasible from the get go. At best the model is able to predict 50/50 on samples of class 1, which is insufficient.

This discovery essentially invalidates this whole project, nonetheless I only found out very late in the process and will still present my results from employing confident learning.

All test set results in a combined table:

data/method	BCE weighted	Mixup	GMM	MixupAndGMM
standard	0.824 / 0.769	0.849 / 0.806	0.848 / 0.750	0.839 / 0.815
CL 1% removed	0.831 / 0.773	0.839 / 0.821	0.830 / 0.788	0.843 / 0.812
CL 2% removed	0.793 / 0.742	0.838 / 0.793	0.804 / 0.778	0.846 / 0.802

Table: Male and female AUROC on the **test set.** The resnet50 model used for the above evaluations where whichever archived peak validation set performance somewhere during the training.“standard” corresponds to the results previously presented.

The above table is by no means conclusive but does suggest a few noteworthy things:

- Mixup appears to increase performance for both genders
- GMM improves male performance but not female performance
- MixupAndGMM performance on par with plain mixup and I choose to credit the performance to the Mixup part of MixupAndGMM rather than the GMM part.
- removing 1% of the data using CL appears to reduce the gender gap for all methods.
- removing 2% of the data appears to negatively impact performance for all methods.

The smallest gender gap is achieved by reducing the training data according to 1%CL and retrain using mixup.

All results in the above table are not averaged across multiple training runs and are most likely not statistically significant. I did not have time nor patience to train that many models. Neither did I have the willingness to do so considering the problem/task of predicting Pneumothorax appears infeasible in the first place.

The reason for the employed noisy label methods not being able to increase the performance may be 2 fold. On one hand the data may not even be noisy in the first place. On the other hand and more likely is it, that severe class imbalance negatively impacts the effect of these methods. In the noisy label literature the data used to develop these methods are rarely imbalanced or at least not to this extent. Thus there is no guarantee that the method works in the extreme class imbalance case present for Pneumothorax in the NIH dataset.

Future investigations:

- Check for shortcut learning:
  - Manual human inspection of the correct/incorrect predicted x-ray images, to check for patterns.
  - Implement some kind of saliency map visualization to better understand what the model looks for when predicting.

- Try the noisy learning methods on other diseases
- Rerun experiments to check the significance of the results.

Conclusion:
There are strong indications that the task of predicting/classifying Pneumothorax from x-ray images using the NIH dataset is ill-posed or infeasible as the trained models performs 50/50, that is random on the data of the positive class. Using either a better model and/or twice the amount of data did not result in a better classifier, which strongly supports the claim of an ill posed task.
If one were to disregard this fact and consider the results of the employed noisy label learning methods, there is some evidence that confident learning combined with mixup is able to reduce the gender gap slightly. This result is by no means conclusive and most likely not even statistically significant. Nonetheless this result suggests that the gender gap may be caused by noisy labels in the training data.