AUC Maximization by Deep Learning for Imbalanced Medical Image Classification

John Mo, Linh Nguyen, and Lawrence Wong

Abstract

In the field of medical image analysis, class imbalance is a major challenge where the majority of samples belong to the negative class (e.g. healthy individuals), while only a small proportion represents the positive class (e.g. patients with a particular disease). This imbalance causes suboptimal performance of deep learning models in detection. We address this issue by maximizing the area under the receiver operating characteristic curve (AUC). Focusing on the BreastMNIST and PneumoniaMNIST datasets from MedMNIST, we leverage the AUCM loss from LibAUC and explore techniques such as optimizer selection, neural network structures, and data augmentation strategies to improve AUC performance. Our findings demonstrate the effectiveness of combining AUC maximization with careful regularization and augmentation approaches, achieving an average test AUC of 0.9588 and a highest test AUC of 0.9745 on the PneumoniaMNIST dataset, outperforming the baseline Cross-Entropy approach.

1. Introduction

Deep learning techniques have achieved success in various sectors, including medical image analysis. A common challenge is the class imbalance problem, where the majority of samples belong to a negative class (e.g. healthy individuals), while only a small proportion represents the positive class (e.g. patients with a particular disease). This imbalance causes deep learning models to be biased towards the majority class, leading to suboptimal performance in detecting the minority class instances despite that being the primary interest.

Traditional evaluation metrics like accuracy are misleading in these imbalance scenarios as a model could predict everything as the majority class and achieve a high accuracy, while not getting any of the minority correct. To address this issue, we can use the area under the receiver operation characteristic (ROC) curve (AUC). This provides a better assessment that considers the trade-off between true positive rates and false positive rates across different thresholds.

In this project, we are focusing on deep learning models to maximize the AUC metric for our two imbalanced medical image classification tasks from the MedMNIST dataset: BreastMNIST and PneumoniaMNIST. We explore various techniques to improve the AUC score, building upon the AUCM loss function from the LibAUC library.

By addressing the class imbalance issue and optimizing for AUC, our goal is to develop deep learning models that can effectively detect minority class instances, which is crucial for accurate diagnosis in medical imaging applications.

2. Method

This section outlines the methodology employed to address the challenges of imbalanced medical image classification and maximizing the AUC. We detail the dataset used, followed by a description of the baseline models and the AUC maximization approach. Additionally, we will dive into the techniques we explored for improving the accuracy of the models.

2.1 Data

The data we used to conduct the two medical image classification tasks were taken from MedMNIST, namely the BreastMNIST and PneumoniaMNIST datasets. These datasets are composed of grayscale images representing breast ultrasounds and chest X-rays respectively. Each dataset consists of a majority of negative instances and a minority of positive instances, reflecting the common class imbalance in medical imaging datasets.

2.2 Baseline Models

For the baseline models, we trained two ResNet18 architectures with one using the Cross-Entropy (CE) loss function and the other using the AUCM loss function. The model trained with the AUCM Loss function served as the base model for our experiments.

By training both CE and AUCM baseline models, we aimed to compare the performance of the base AUCM loss function against our proposed approach of maximizing AUC.

2.3 Cross Entropy (CE) Approach

The Cross Entropy loss function is widely used for classification and it measures the performance of a model by calculating dissimilarity between predicted probability distribution and true distribution. For the baseline model, we trained a ResNet18 architecture using the standard CE loss function which served as a comparison point to evaluate the effectiveness of the AUC Maximization approach using the AUCM loss function.

Despite being often used for classification tasks, CE does not explicitly account for imbalanced datasets. Thus, models trained with CE loss may have a bias towards the majority class, causing suboptimal performance in detecting minority class instances.

2.4 AUC Maximization Approach

To maximize the AUC, the AUCM Loss function from LibAUC was to optimize the AUC in the ResNet18 architecture. In addition to using AUCM Loss, we also utilized the DualSampler from

LibAUC to counteract the imbalance in the dataset by dynamically adjusting the sampling strategy.

During the training process, the DualSampler dynamically selects samples from both the majority (negative instances) and minority (positive instances) classes based on their importance in maximizing the AUC score. This adaptive sampling strategy ensures that the model receives sufficient exposure to both classes, allowing it to learn discriminative features effectively without being biased towards the majority class.

By incorporating the DualSampler into our training pipeline, we aimed to mitigate the adverse effects of class imbalance on model performance. This approach not only enhances the model's ability to distinguish between positive and negative instances but also promotes more robust and generalizable learning.

Furthermore, the DualSampler works synergistically with the AUCM Loss function. The AUCM Loss function aims to maximize the AUC by directly optimizing for it during training. However, in imbalanced datasets, the scarcity of positive instances can lead to biased learning, where the model predominantly focuses on the majority class. The DualSampler ensures that both positive and negative instances are adequately represented during training.

2.5 Techniques For Improvement

We explored three main techniques for improving the performance of our AUC maximization approach: Data Augmentation, Optimizer Selection, Neural Network Structures, and Regularization Terms.

Data augmentation is a widely used technique to enhance model generalization by artificially expanding the training dataset with variations of the original images. We used several augmentation techniques, including horizontal and vertical flips, brightness adjustment, contrast adjustment, and Gaussian blurring.

We investigated the impact of different optimizers on the performance of the model. In addition to the commonly used Adam optimizer, we experimented with alternative optimizers such as the Momentum method. By tuning parameters such as the learning rate, margin rate, and batch size, we aimed to find the optimal configuration that improves generalization.

We experimented with different neural network architectures to investigate their impact on classification performance. Specifically, we evaluated the performance of ResNet50 and DenseNet121 architectures alongside the baseline ResNet18.

We also explored the use of regularization terms to enhance the performance of our AUC maximization approach. By incorporating these regularization terms into our training process, we aimed to control the complexity of the model and improve its ability to generalize to unseen data.

3. Experiments

In this section, we present the results of the experiments conducted to maximize the AUC for imbalanced medical image classification tasks using deep learning models. Our experiments revolve around exploring data augmentation strategies, optimizer selection, different neural network structures, and regularization terms to enhance model performance.

3.1 Dual Sampler

BreastMNIST

Dual Sampler	**Average Test AUC**	**Highest AUC**
Original W/O Dual Sampling	0.7275	0.8810
Original W/ Dual Sampling	0.8518	0.8915

PneumoniaMNIST

Augmentation	**Average Test AUC**	**Highest AUC**
Original W/O Dual Sampling	0.9415	0.9696
Original W/ Dual Sampling	0.9533	0.9788

The results indicate that the average AUC score without the dual sampler is consistently lower than when tested with the dual sampler. This improvement highlights the effectiveness of the dual sampler in mitigating the adverse effects of class imbalance and enhancing the model's ability to maximize the AUC score.

3.2 Data Augmentation

BreastMNIST (All augmentations were done with Dual Sampler)

Type of Augmentation	**Average Test AUC**	**Highest AUC**
Original	0.8518	0.8915
Horizontal Flip	0.7982	0.8504

Type of Augmentation	Average Test AUC	Highest AUC
Vertical Flip	0.6756	0.7360
Brighten	0.7938	0.8476
Darken	0.6073	0.7600
Gaussian Blur	0.8525	0.9047
Increased Contrast	0.8219	0.8715
Decreased Contrast	0.7841	0.8385

From the results, we can observe that all the techniques, except for blur, did worse than the original dataset with the DualSampler. This suggests that certain augmentation techniques may not be suitable for the dataset. For instance, horizontal/vertical flips, brightening, darkening, and increasing/decreasing contrast might have introduced noise or distorted critical features in the medical images, making them less discriminative for the classification task.

PneumoniaMNIST (All augmentations were done with Dual Sampler)

Augmentation	Average Test AUC	Highest AUC
Original	0.9533	0.9788
Horizontal Flip	0.9147	0.9503
Vertical Flip	0.7898	0.8867
Brighten	0.8636	0.9549
Darken	0.9310	0.9657
Gaussian Blur	0.9428	0.9713
Increased Contrast	0.8591	0.9493
Decreased Contrast	0.9588	0.9745

From the results, we can observe that all techniques, except for decreased contrast, did worse than the original with the DualSampler. This suggests that the data augmentations potentially introduced noise or distorted key features in the medical images, making them less discriminative.

3.3 Optimizer Selection

To test optimizers, each one was run individually while keeping the other optimizers consistent with the base model. The results are shown below.

BreastMNIST

Optimizer	Values Tested	Best Value	Average Test AUC for Best Value	Highest Test AUC
Base (PESG)	N/A	N/A	0.881	0.916
Adam (Change Beta1)	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95	0.95	0.624	0.693
Adam (Change Beta 2)	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 0.9999	0.9999	0.612	0.661
Momentum	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95	0.8	0.896	0.955
Learning Rate	0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500	0.1	0.872	0.910
Margin Rate	0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500	0.01	0.873	0.922
Batch Size	2, 4, 8, 16, 32, 64	16	0.883	0.929

PneumoniaMNIST

Optimizer	Values Tested	Best Value	Average Test AUC for Best Value	Highest Test AUC
Base (PESG)	N/A	N/A	0.932	0.978
Adam (Beta1)	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95	0.95	0.660	0.693

Adam (Beta 2)	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 0.9999	0.9999	0.611	0.650
Momentum	0.6, 0.7, 0.8, 0.9, 0.95	0.8	0.948	0.971
Learning Rate	0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500	0.1	0.944	0.972
Margin Rate	0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500	0.0001	0.948	0.977
Batch Size	2, 4, 8, 16, 32, 64	4	0.929	0.960

3.31 Adam

Overall, adjusting Beta 1 and Beta 2 for the Adam optimizer did not lead to a substantial improvement in the model performance, rather, the AUC classification rate was significantly lower than the base PESG model. Generally, Adam demonstrated limited effectiveness for AUC classification, potentially due to several reasons ranging from the large data size to overfitting of the training data.

For both BreastMNIST and Pneumonia, Adam appeared to not be effective and could potentially be solved with different parameters.

3.32 Momentum

From the results, momentum parameters greater than 0.7 (specifically 0.8, 0.9, and 0.95) consistently outperformed the base model. There was approximately a 6% difference in the AUC score between 0.7 and 0.8.

The momentum method exhibited its highest efficiency at 0.8 with 0.9 and 0.95 showing a relatively similar performance; the smaller values showed lower effectiveness comparatively. Low momentum values would result in difficulty overcoming local minima while a higher gradient would result in potential issues converging on a minima. This is similar to problems found with the learning rate.

3.33 Learning Rate

Ultimately, the learning rate caused the AUC to vary quite dramatically, with the lowest AUCs having scores of less than 0.60 for learning rates of 0.0001, 100, and 500. Too low of a learning rate would lead to difficulty overcoming local minima, whereas too high would result in difficulty in convergence.

Conversely for PneumoniaMNIST, the model accuracy did not significantly drop for a lower learning rate but dropped for a larger learning rate. This could be caused by a variety of factors but is most likely due to a lack of large local minima.

3.34 Margin Rate

For both BreastMNIST and PneumoniaMNIST, drastically different margin rates do not significantly alter the AUC Score. The BreastMNIST dataset had a difference of 0.4 between the maximum and minimums and the PneumoniaMNIST dataset had a difference of 0.2 between the maximum and minimums. This could be due to a variety of reasons, however, the most likely explanation is that the data is well separated. Nevertheless, the difference in classification rates lies in hard-to-classify points and noise.

3.35 Batch Size

In the BreastMNIST dataset, the model appeared to perform better with larger batch sizes, with the test AUC slowly converging as batch size increased. On the other hand, the optimal batch size for the PneumoniaMNIST dataset was 4, with a batch size of 2 resulting in the worst AUC score of 0.90. The remaining batch sizes stabilized at around 0.94.

3.4 Neural Network Structures

3.41 Resnet50

For both BreastMNIST and PneumoniaMNIST, Resnet50 and Resnet18 both produced the same result. Because Resnet50 has more layers and can capture more variability than Resnet18, the result being similar implies that the data had distinct features that were able to be captured by a lower complexity model.

3.42 Densenet121

For both BreastMNIST and PneumoniaMNIST, Densenet121 outperformed the base model of ResNet18. Due to DenseNet looking at previous layers and not skipping layers like ResNet, it was able to detect trace amounts of variance that ResNet could not. This, however, comes with a trade-off as DenseNet121 takes more time to train.

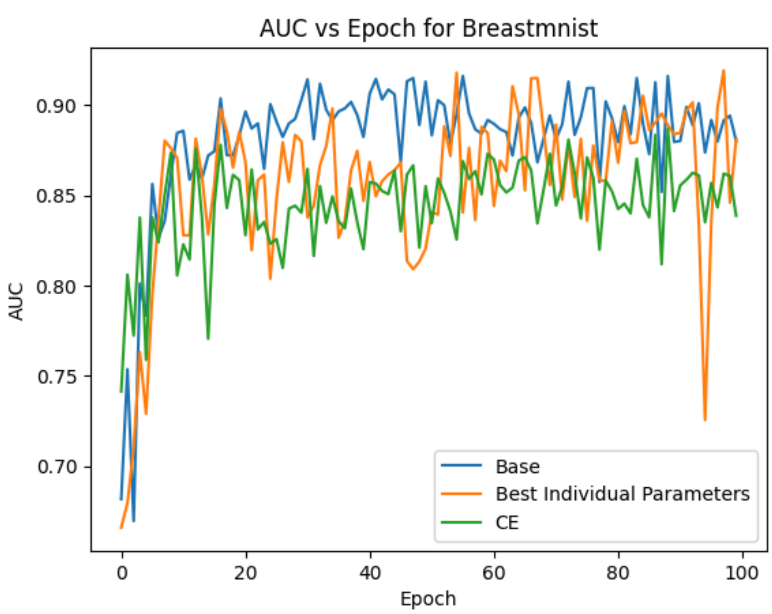
3.5 Regularization

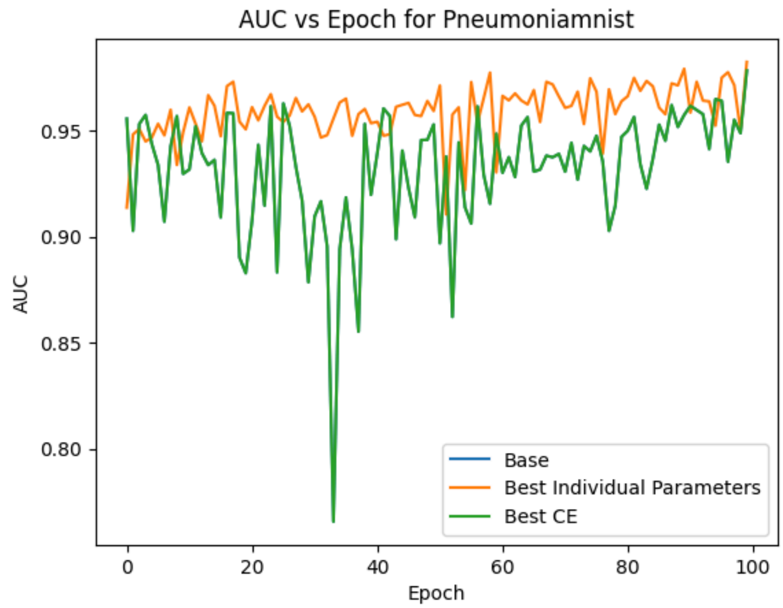
3.51 L2 Regularization

The L2 regularization was implemented by adjusting the weight decay with values of 0, 0.1, 0.01, 0.001, 0.0001, and 0.0001 being tested. With the BreastMNIST model, the absence of a regularizer yielded the best results, whereas having a weight decay of 0.0001 was beneficial for PneumoniaMNIST. Although there could have been a variety of reasons as to why no regularizer was the best for BreastMNIST, one possible explanation could be that BreastMNIST is more complex than PneumoniaMNIST and needs to capture as much variance as possible. This stands to reason considering PneumoniaMNIST generally has had a higher test AUC for all models, potentially meaning the model has fewer features to focus on and thereby needs a higher bias to sufficiently work.

3.6 Best Models and Results

With the best set of parameters individually calculated, the best parameters were then combined to create the best model. The results are shown below.





Overall, this solution didn't effectively train the best model for either model. The new model has a considerable amount of instability across the epochs with a slightly lower testing AUC. Ultimately, this likely stems from the optimizers, which largely dictate the model's learning. The drastically different AUC scores throughout the graphs resemble the results for different learning rates suggesting that is the most probable cause for the AUCs. However, in the scope of this project where the objective was to only change one parameter at a time, there are bound to be several interaction effects that could not be properly calculated within the timeframe.

4. Conclusion

Our approach involved leveraging the AUCM loss function from LibAUC and incorporating the DualSampler to counteract the class imbalance by dynamically adjusting the sampling strategy. The results demonstrated the effectiveness of the DualSampler in improving the average AUC scores for both datasets tested.

We further investigated various techniques to enhance the AUC max performance, including data augmentation, optimizer selection, different neural network architectures, and regularization. The experiments revealed that certain augmentation techniques, such as Gaussian blurring, benefitted the BreastMNIST dataset, while decreasing contrast, improved results for PneumoniaMNIST.

Overall our work highlights the importance of addressing class imbalance and optimizing for the AUC metric in medical image classification tasks. By combining AUC max with regularization and augmentation, we developed more robust and accurate deep-learning models for detecting minority class instances.

4.1 Future steps

While we found promising results for individual optimizers, there are ways for further exploration. We could combine multiple models trained with different techniques using ensemble methods to increase the likelihood of improving overall performance. We could also attempt transfer learning, even though we originally saw it as out of scope for this project.

Unfortunately, we were unable to acquire a significant increase in AUC classification for either model. Although for PneumoniaMNIST, the AUC classification was already quite high, there were minor improvements that stemmed from changing singular factors, however combining those best singular optimizers led to a decrease in overall performance from the base model. Regardless, the increase in performance from increasing a singular factor for every optimizer implies a middle ground between the optimizers where they will not disrupt each other. For the scope of this project with limited training time, we were not able to find the middle ground between optimizers. Running the model to test the potential interactions is a valid next step. Additionally, there is considerable room for code optimization and simplification to allow for faster runtime, such as using PCA. However, it is worth noting that doing so would reduce the overall accuracy.