Using Adversarial Attacks as Data Augmentation Methods for Deep Learning

澳門理工大學
Universidade Politécnica de Macau
Macao Polytechnic University

Student: Zhongbo Yan, Steve   Supervisor: Xu Yang   Assessor: Liam Lei

Bachelor of Science in Computing (2022-2023)   Macao Polytechnic University

Email: P1908326@mpu.edu.mo | me@aspires.cc

Abstract

Adversarial attacks can be used to deceive deep learning models, producing incorrect predictions. By generating imperceptible perturbations to original images, these attacks can serve as an effective data augmentation method for enlarging training datasets by giving the attacked image a true label, hence improving the performance of image classification models. In this project, an adversarial training method was evaluated using the MRNet dataset for knee MRI images. Using FGSM attack, extra perturbed images were generated for retraining a well-trained CNN model. Results showed improved **accuracy**, **AUC**, **sensitivity**, and **specificity** compared to the baseline model provided by the Stanford Machine Learning Group.

Introduction

Magnetic resonance imaging (MRI) of the knee is the standard care imaging modality to evaluate knee disorders. More musculoskeletal MR examinations are performed on the knee than on any other body region.

The MRNet dataset, provided by the Stanford ML Group, contains knee MRI examinations used to determine abnormal categories. The dataset has been split into the training (1130 exams) dataset and the validation (120 exams) dataset.

In addition, in each exam, there are MR images in three planes, namely **axial**, **coronal**, and **sagittal**, for classifying three types of abnormality (**abnormal**: 80.6%, **ACL tears**: 23.3%, and **meniscal tears**: 37.1%) obtained through manual extraction from clinical reports.

Planes/Tasks	Abnormal	ACL tears	Meniscal
Axial	Model$_{A1}$	Model$_{A2}$	Model$_{A3}$
Coronal	Model$_{C1}$	Model$_{C2}$	Model$_{C3}$
Sagittal	Model$_{S1}$	Model$_{S2}$	Model$_{S3}$

Table 1. MRNet Binary Classification Models

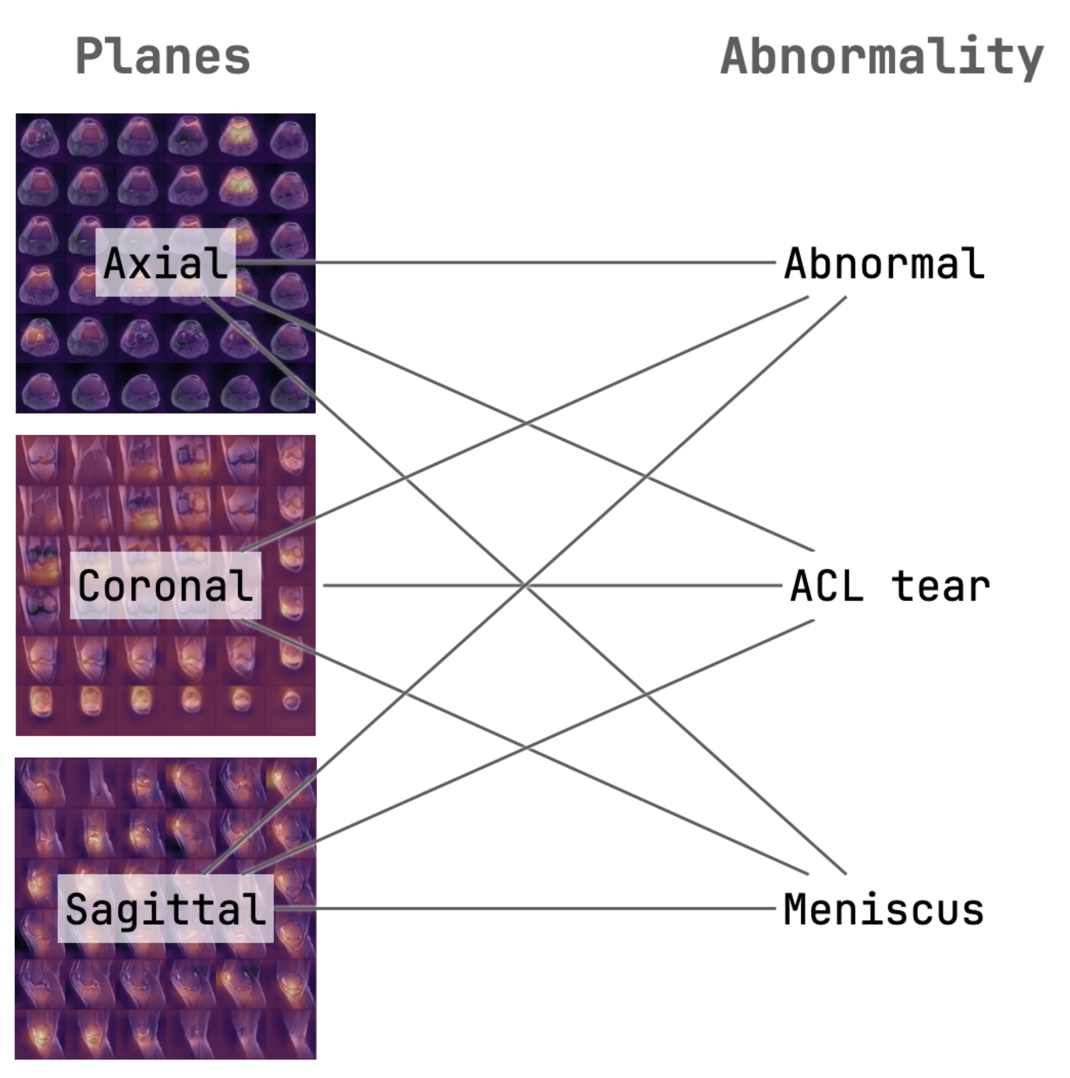

Figure 1. MRNet Dataset

Baseline Model of this project is the best model [mrnet-baseline (single model) Stanford University, AUC: 0.917] released on the ranking board of MRNet. The code of the baseline model can be found on GitHub: `matteo-dunnhofer/mrnet-baseline`|

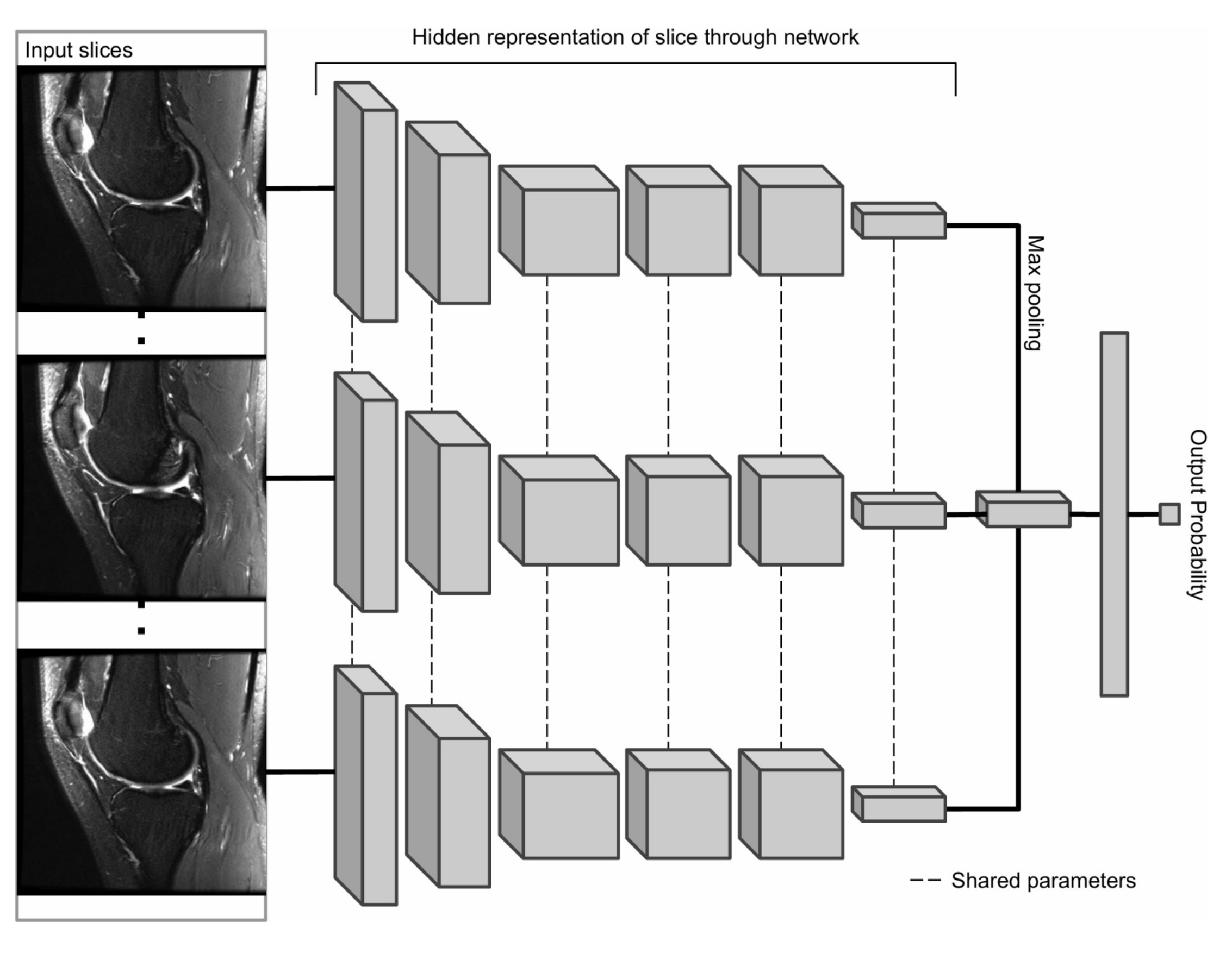

Figure 2. Baseline Network Visualization

This project's experiment involves training a baseline model using its conventional data augmentation techniques. Afterwards, the models are retrained with a particular portion of data, known as the retrain percentage, which is perturbed with a specific Fast Gradient Sign Method (FGSM) attack denoted as epsilon (ϵ). The objective of this experiment is to assess the effect of re-training using adversarial examples on the MRNet model's performance.

Problem

In 2019, Tasnim et al. proposed InvFGSM technique for medical image segmentation. In 2020, Xie et al. introduced AdvProp to improve recognition accuracy using separate batches for original data and adversarial examples generated by PGD attacks. Gong et al. introduced MaxUp, which updated neural network parameters using augmented data with perturbations by maximum loss of each batch.

In the literature, the current published works only evaluate the performance improvement of accuracy and AUC. the impact on other metrics such as sensitivity, and specificity have not been studied, to the best knowledge of the author. Additionally, there has been no previous research conducted on medical Magnetic Resonance Imaging (MRI) classification tasks using adversarial data augmentation.

Algorithm & Implementation

The Fast Gradient Sign Method (FGSM), introduced by Ian Goodfellow et al. is one of the most popular adversarial attack methods, which aims to generate malicious examples by perturbing the input data to maximum gradient of loss. The carefully crafted perturbation is designed to be small enough to be visually imperceptible to humans, while still being large enough to cause the model to misclassify the input.

$$X_{adv} = X + \epsilon \times sign(\nabla_X J(\theta, X, y)) \quad (1)$$

(1) Equation of generating adversarial examples using FGSM

▪ X_{adv}	adversarial example generated using the FGSM
▪ X	original input example
▪ ϵ	the magnitude of the perturbation
▪ $sign()$	function that returns the +/- sign of its argument
▪ $\nabla_x J(, X, y)$	gradient of the model's loss function J
▪ θ	model parameters
▪ X, y	input data, true labels

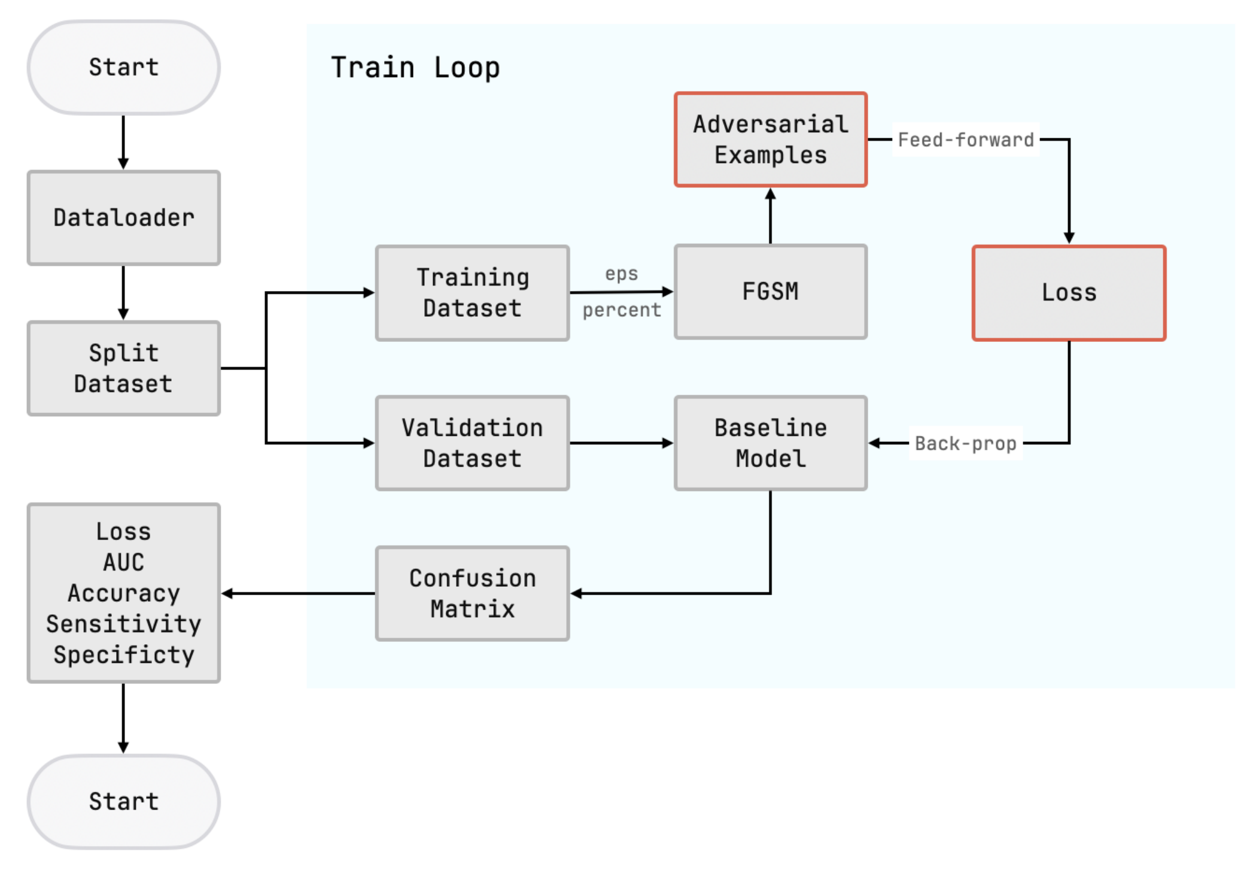

Figure 3. Experiment Flow Chart

Adversarial training is a technique that aims to make models more robust to these attacks by training them on adversarial examples (perturbed input). The diagram [Figure 3] illustrates this procedure.

The adversarial training process is performed for each epoch, following these steps, where μ is the learning rate, θ is the model parameter, and (X_i, y_i) is a batch of training data determined by the hyperparameter for the retrain percentage.

$$\delta_i^{FGSM} := \epsilon \times sign(\nabla_{X_i} J(X_i, y_i, \theta)) \quad (2)$$
$$X_i^{adv} := X_i + \delta_m^{FGSM} \quad (3)$$
$$\theta := \theta - \mu \nabla_\theta J(X_i^{adv}, y_i, \theta) \quad (4)$$

(2) Generating adversarial perturbations δ_i^{FGSM} for training data X_i

(3) Generating adversarial examples X_i^{adv}

(4) Updating (Gradient Descent) Model's parameters θ

Model/% of Data Retrain	1%	2%	6%	10%
Baseline(mean)	89.30	89.29	89.30	89.27
Baseline(mean) + Advtrain	**89.35**	**89.41**	**89.41**	**89.41**
Baseline(Model$_{S3}$)	76.50	76.41	76.53	76.44
Baseline(Model$_{S3}$)+ Advtrain	76.50	**76.53**	**76.64**	**76.87**
Baseline(Model$_{S1}$)	94.78	94.82	94.82	94.95
Baseline(Model$_{S1}$)+ Advtrain	**95.12**	**95.45**	**95.79**	**95.71**

Table 2. Retraining on Multiple Hyperparameters

The experiment found that the best results were obtained with epsilon values of 0 to 0.003, and retrain percentages of 0.01 to 0.06 and 0.1. Overall, the experiment shows that the choice of hyperparameters can have a significant impact on the performance of adversarial training and that careful tuning of these hyper-parameters is necessary to achieve optimal results.

Results

Training the model on perturbed images forces it to make more confident and accurate predictions on perturbed images. This pushes the decision boundary away from the training data distribution and towards the true decision boundary, resulting in improved performance on test images, whether they are perturbed or not.

The validation AUC learning curves indicate improvement in several models, particularly in classifying abnormalities on the sagittal plane. Additional learning curves on other evaluation metrics (accuracy, sensitivity, specifity) can be found in the Final Report. The following curves are visualized on hyperparameter:

$$[\epsilon = 1e-5, \text{retrain percentage} = 6\%]$$

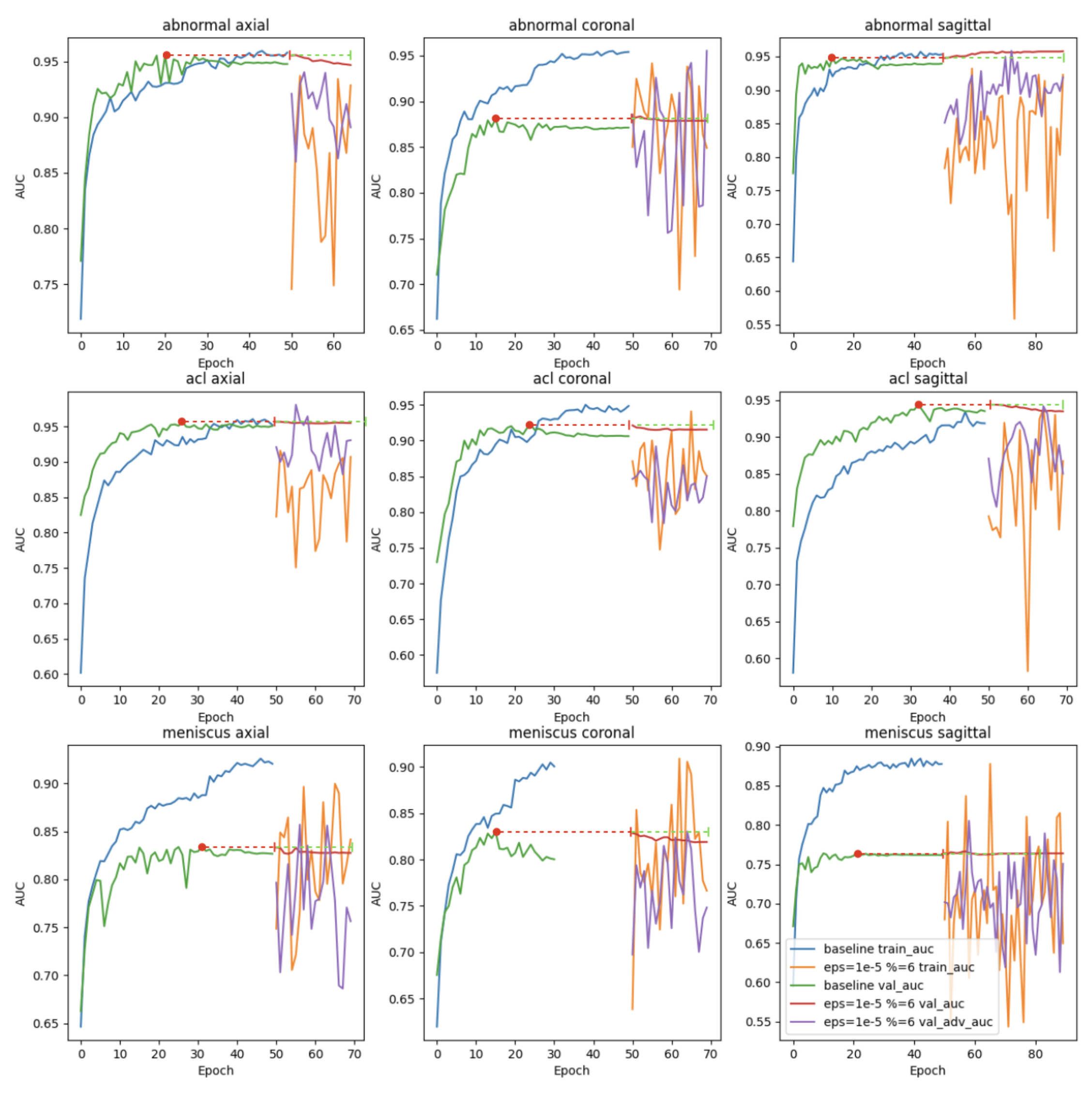

Figure 4. Learning Curve on Validation AUC

The ROC curves for the nine binary models can tell how sensitive (true positive rate) and specific (false positive rate) the classifiers are for each threshold. The closer the curve is to the top-left corner of the plot, the better the classifier is.

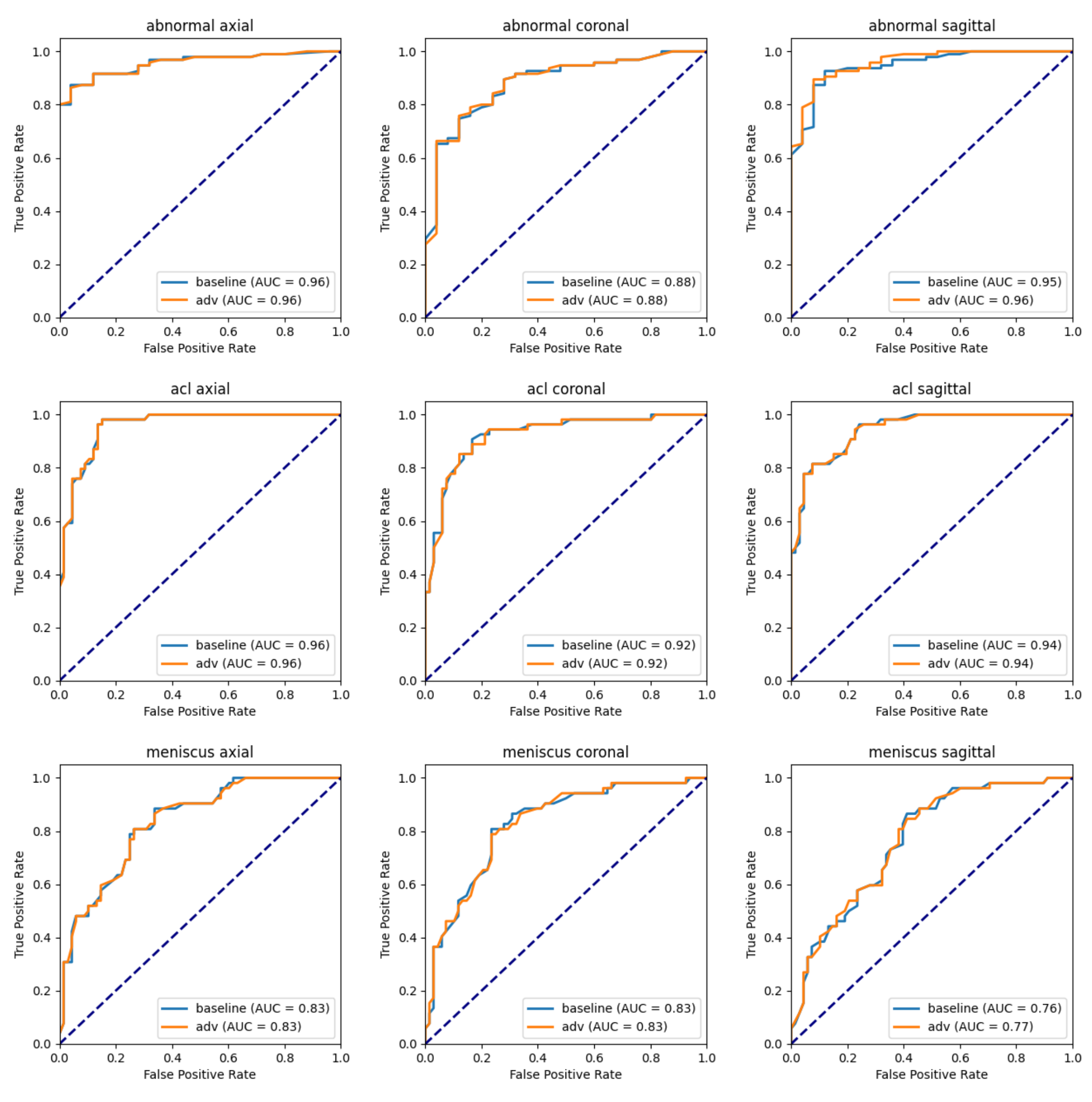

Figure 5. ROC Curve comparison

Conclusion

The adversarial noise δ_i^{FGSM} is different from normal noise and its ability to improve the classification AUC suggests that training with adversarial examples can influence the decision boundary. This occurs by causing the model to learn from misclassified inputs that are very similar to the original data. The perturbations can shift the input to a different region of the input space, which may be closer to a different class. As a result, the decision boundary can shift, or even new decision boundaries not present before can be created .

Adversarial training is a powerful technique for improving deep neural networks' performance on medical image analysis tasks by helping them learn more complex and informative features specific to the task. However, the finding that the ROC curve worsened for one of the classification tasks (meniscus-coronal) highlights the potential limitations and challenges of adversarial training. Careful fine-tuning of the hyperparameters and implementation details is crucial for optimizing the model's performance.