[redacted] Assignments

1. Contents

Assignment 1

1. Analysis

- **Visualize the dataset:**

The initial inspection of the visualization result of the dataset appears satisfactory. The annotations corresponding to the glomerulus are consistent with the shape of the glomerulus. However, without annotation documents, it remains uncertain whether any glomeruli have been mistakenly labeled as normal tissue.

- **Unbalanced Data Handling:**

The issue of unbalanced data must be addressed. To handle this, I extracted the class weights during the dataloading phase and computed the loss accordingly. I then assigned a specific weight factor to each pixel in the image, forming a weight map. Applying this weight map to the sample_loss in the loss calculation proved effective in mitigating the effects of unbalanced data. Additionally, incorporating specific loss functions in further experiments, such as Dice loss or focal loss, may further contribute to resolving this issue.

- **Treatment of Unannotated Data:**

The dataset includes a category known as unannotated data, which I believe should not be treated as a separate class. To address this, I made several considerations:

- The images were segmented into only 3 classes, excluding the unannotated data.
- In the computation of weights, I assigned a value of zero to the unannotated data. Consequently, during the training process, the pixels corresponding to this unannotated class had no impact on the loss, ensuring that they did not affect the overall performance.
- While computing the metrics, I also factored in the weight assignment, 0 for unlabeled data and 1 for others. The pixels of the unlabeled data were excluded from metric computation, maintaining the integrity of the evaluation process.

- **Specific color distribution**

The original images are of a unique color distribution. Considering this, I extracted the mean and standard deviation (std) for the input images. By normalizing the images with these statistical parameters before inputting them into the network, I ensured they were processed appropriately. Besides, the color augmentation method shouldn't be use or use it in a low frequency during the training in case it affects the color distribution on the data.

- **Limited data**

The dataset is limited, which raises the possibility of overfitting. To mitigate this risk, I implemented some data augmentation techniques. These included random cropping, rotation, color jittering, and enhancements to brightness and contrast, as well as the addition of noise. Such augmentations may contribute to improving the model's ability to generalize from the limited data available.

2. Data Preparation

I divided the original dataset into ten folds, allocating 8 for training, 1 for validation, and 1 for testing. To further assess the performance of the model, cross-validation techniques may be employed.

3. Model Training

a) Code Structure and corresponding introduction

-- data

    |-- train

        |-- images: Contains the original training images.

        |-- mask: Contains the corresponding masks for segmentation.

    |-- val: Holds validation images and masks.

    |-- test: Stores the images and masks for final testing.

-- saved_models: contains the checkpoints

-- src

|-- *models*

|-- *unet.py:* Implements the U-Net architecture with the pretrained ResNet-50 as the backbone. U-Net is famous for its efficiency in biomedical image segmentation. Using ResNet-50 as the backbone adds the benefit of deep residual learning, enhancing the ability to learn from complex patterns in the data. The combination of U-Net with ResNet-50 not only maintains high-resolution features throughout the network but also promotes faster convergence and potentially better performance on segmentation tasks.

|-- *utils*

|-- *dataaugment.py:* Defines functions for data augmentation, including random cropping, rotation, color jittering, and brightness and contrast enhancement. These techniques aim to prevent overfitting and improve model generalization.

|-- *dataloader.py:* Handles data loading, preprocessing into tensors, class weights computation, image normalization using previously calculated mean and standard deviation, and the transformation of masks into one-hot format. Additionally, it sets the buffer size, batch size, and shuffle parameters.

|-- *help_functions.py:* Provides visualization tools for the original images, predicted results, and ground truth. These visualizations are essential for monitoring the training process and debugging.

|-- *losses.py:* Specifies the loss functions, employing cross-entropy loss with a weight factor and supervision of the intermediate layer of the U-Net. It also details the combination of losses with specific weight factors to ensure model convergence.

|-- *metrics.py:* Describes the metrics used for evaluation, during training process, primarily focusing on mean Intersection over Union (mIoU). A function for class-wise IoU is also included, if needed. Besides, during the evaluation phrase, precision, recall and F1 score for glomerulus is also calculated.

|-- inference.py: Contains the inference procedure to observe the model's performance on unseen data.

|-- *train.py:* Manages the training process using the Adam optimizer with exponential decay learning rate. It includes visualization of the loss, training mean IoU, and validation mean IoU, as well as model saving functionality.

|-- *evaluate.py:* Defines the evaluation procedures for the trained model.

4. Evaluation

Class-wise IOU	[0.7726709594530521, 0.9889613793494276, 0.9977415732451711]
Mean IOU	0.9197913040158836
Precision for glomerulus	0.8020767178966086
Recall for glomerulus	0.9547009354625342
F1 for glomerulus	0.8717590315705915

The precision of glomerulus is not meeting the expected standards. This deficiency might be attributed to the weight factor, which seems to cause more pixels to be predicted as glomerulus. To enhance the accuracy, a series of further experiments should be initiated, aimed at a comprehensive understanding of this behavior, and to consequently develop more precise prediction methods.

5. Bad Case Analysis

After completing the initial version of the model, it's essential to analyze the bad cases and evaluate relevant metrics. This evaluation process may involve identifying and addressing specific issues that arise from the results. In the context of a segmentation task, this could include assessing whether the network adequately captures both contextual and spatial information, and then making appropriate adjustments to the network structure accordingly.

In my own experiments, I attempted to resolve certain problems. For instance, when I found that the context information was insufficient, I introduced additional losses to supervise the shallow layers during the up-sampling process. However, it turns out it is a bug in the data augmentation process. And I also discovered that color augmentation led to a decrease in the model's performance. To mitigate this, I carefully controlled the randomness of color augmentations. Despite these efforts, the validation set exhibited a high variance, and with such a limited number of samples, drawing a credible conclusion is challenging.

6. Further Improvements Could Be Made:

- **Data Collection**

The existing quantity of the dataset is limited, leading to potential inadequacies in training the model. Collecting more diverse and comprehensive data could substantially improve model performance.

- Exploration of Different Neural Network Architectures

Investigating alternative network architectures or backbones may yield improvements. For instance, the utilization of atrous convolutions may enhance the receptive field of the model, allowing it to capture more complex patterns. Experiments should be conducted.

- **Experimentation with Various Loss Functions**

The model's performance might benefit from testing different loss functions. Some loss functions may contribute to address the unbalanced data.

- **experiments of Data Augmentation Techniques:**

Implementing additional data augmentation methods, beyond the standard practices, could further enhance the model's robustness and ability to generalize from the available dataset.

- **Form a reliable validation set**

Since the validation set is relatively small: conducting experiments on such a high-variance dataset may lead to results that lack credibility and reliability.

Assignment 2

1. Understanding of the evaluation metrics

- **Precision**

Meaning: measures the accuracy of positive prediction.

Formula: Precision = TP/(TP+FP)

- **Sensitivity (Recall)**

Meaning: measures the ability of the model to identify positive instances.

Formula: Sensitivity = TP/(TP+FN)

- **F1 Score**

Meaning: F1 Score considers both the precision and recall of the classification model. The F1 Score can be seen as a harmonic mean of the model's precision and recall, with a maximum value of 1 and a minimum value of 0.

Formula: F1 Score = 2*TP / (2*TP+FP+FN)

- **Specificity**

Meaning: the ability of the model to identify negative instances.

Formula: Specificity = TN/(TN+FP)

- **Accuracy**

Meaning: Of all the samples, what proportion was predicted correctly, either as positive or negative.

Formula: Accuracy = (TP+TN)/(TP+TN+FP+FN)

- **Matthews Correlation Coefficient (MCC)**

Meaning: MCC (Matthews Correlation Coefficient) is essentially a correlation coefficient between the actual classification and the predicted classification, with a value range of [-1,1]. A value of 1 represents a perfect prediction of the subjects, a value of 0 indicates that the prediction is no better than random guessing, and -1 means that the predicted classification is in complete disagreement with the actual classification.

Formula: MCC = (TP*TN-FP*FN)/ sqrt((TP+FP) *(TP+FN)*(TN+FP)*(TN+FN))

- **Diagnostic Odds Ratio (DOR)**

Meaning: The ratio of the odds of the prediction being positive in the case of a positive instance relative to the odds of the prediction being positive in the case of a negative instance.

Formula: DOR=(Sensitivity/(1−Sensitivity)) / ((1−Specificity)/Specificity)

2. Verify the accuracy of the data.

- Check if the TP+FP+FN+TN equals to the quantity of the whole dataset (71565312). All passed.
- Calculate the evaluation metrics with TP, TN, FP, and FN. Check if there's some wrong data. Except for some data may be divided by 0. All the other data is correct. Just ignore the Nan data when summarize the results.

3. **Preprocess and clean the data.**

- Delete the columns with more than 50% missing value.

['layerDepth', 'focalLoss', 'filterSize', 'kernelIncrease', 'normalization', 'model_name', 'backbone_name', 'encoder_freeze', 'decoder_use_batchnorm', 'middleBlocks', 'time_evaluated']

- Change the discrete column "model", "loss" and "augmentation" to class integer label.
- For some experiments, need to drop the NaN value.

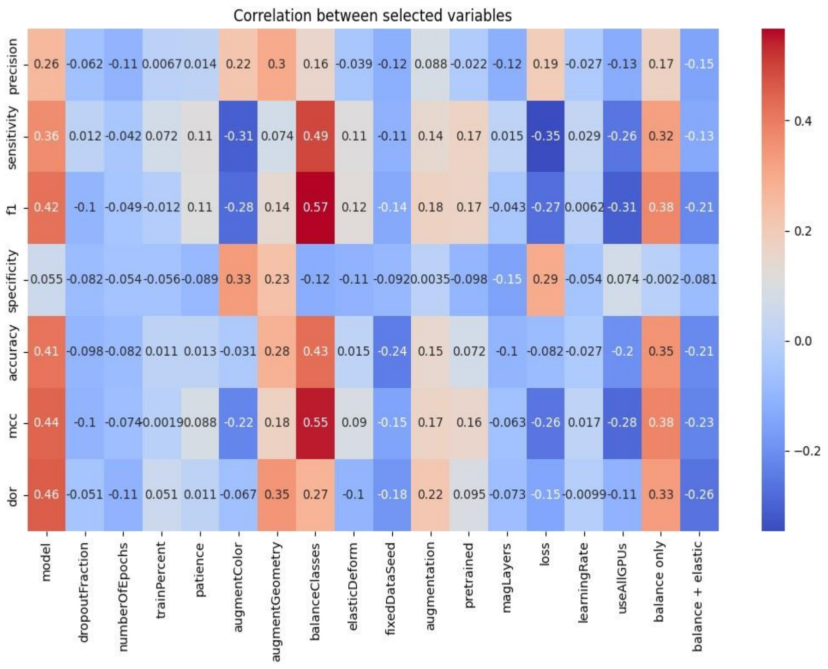
4. **statistical analysis on provided data.**

a) Calculate the confusion matrix between parameters and matrix.

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A heatmap of a correlation matrix is a graphical representation where colors are used to show the statistical correlation coefficients between pairs of variables, with varying shades or colors representing the strength of the correlations.

The meaning of the value:

- **1:** A value of 1 indicates a perfect positive correlation between two variables, meaning as one variable increases, the other one increases at the same rate.
- **-1:** A value of -1 indicates a perfect negative correlation between two variables, meaning as one variable increases, the other one decreases at the same rate.
- **0:** A value of 0 indicates no correlation between the two variables, meaning the variables do not affect each other.
- To maintain clarity and simplicity in the heatmap, we chose the experimental parameters as the x-axis and the evaluation metrics as the y-axis.



As shown in the above Figure, roughly we can get some conclusions on did the experimental parameters related to some of the evaluation metrics. I set the threshold absolute value of corr > 0.15 as the related parameters and did further analysis in the following section.

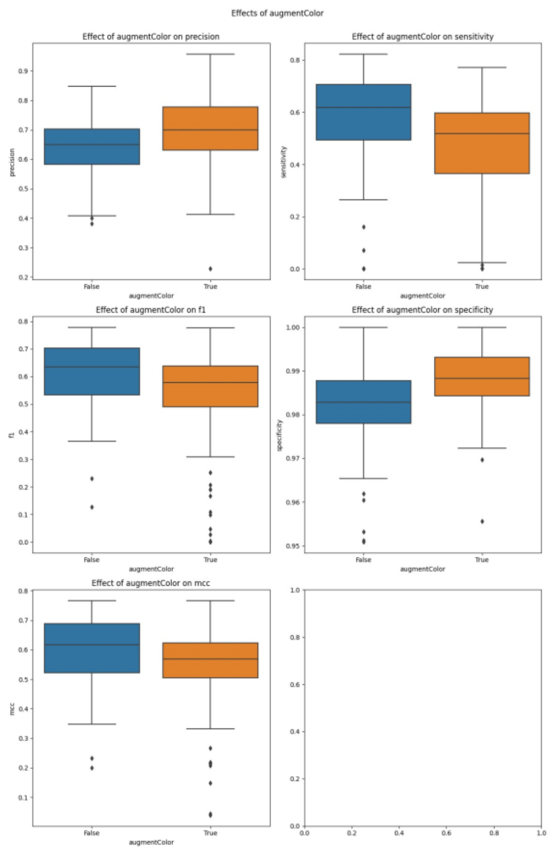
- **Related parameters:** model, augmentColor, augmentGeometry, balancedClasses, fixedDataSeed, augmentation, pretrained, loss, useAllGPUs, balance Only, balance + elastic.
- **Inrelated parameters:** dropoutFraction, numberOfEpochs, trainPercent, patience, elasticDeform, magLayers, learning_rate.

We will identify how those related parameters affects those matrix in the next section.

b) identify what are the effects on the corresponding metrics for those parameters.

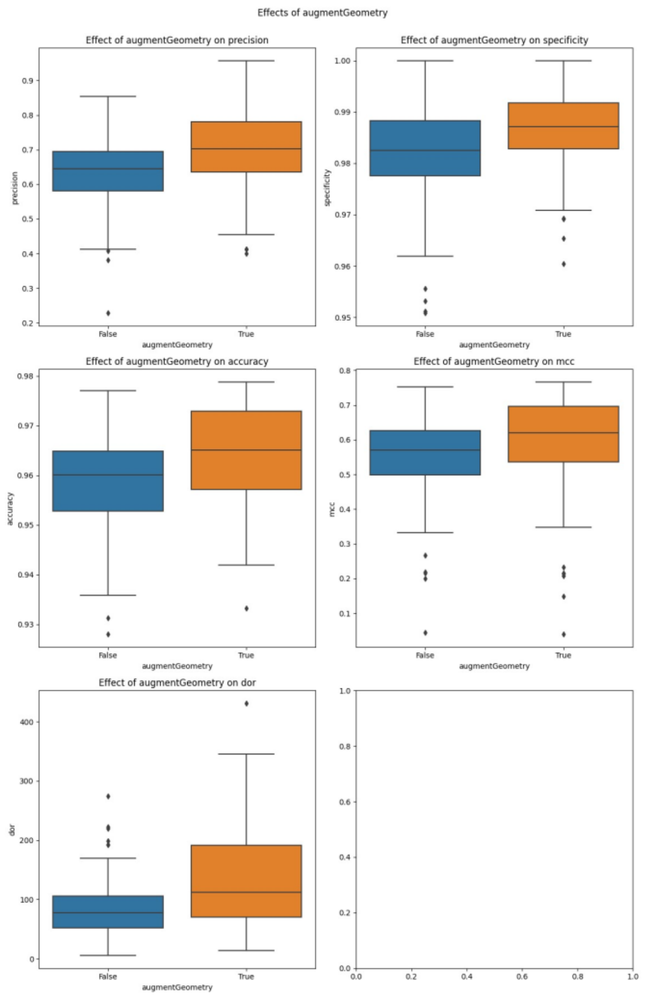
In this section, we will use boxplot to identify the effects. A boxplot, also known as a box-and-whisker plot, is a graphical representation of a dataset that displays the distribution of data based on a five-number summary: the minimum, first quartile, median, third quartile, and maximum. In a boxplot, a box is created from the first quartile to the third quartile, a segment inside the box shows the median, and "whiskers" extend from the box to show the range of the data. It's a helpful tool for identifying the central tendency and variability of data, as well as potential outliers. In a boxplot, the smaller the intersection of the boxes, the greater the correlation between the variables, and conversely, a larger intersection indicates a lesser correlation.

- **augmentColor**



From the above figure, we can see that using “augmentColor” will decrease the sensitivity(recall). I guess that the class “lession” have a specific color distribution. The use of "augmentColor" might disrupt the model's ability to discern these color patterns, consequently affecting other metrics such as the F1 score and MCC, leading to an overall decline in performance.
On the other hand, the specificity and precision exhibit an increase. This suggests that more pixels are being classified as the negative class (such as tissue or background), resulting in a rise in these metrics. While this may appear beneficial, the primary objective is to accurately identify lesions, so this increase might not be in line with the desired outcome.
Besides, on the Figure of f1 and mcc, using augmentColor have more outliers, which means that in some situations, using augmentColor will make the network hard to converge.
Considering those conclusions and the goal, it might be advisable to refrain from using "augmentColor" during the training process.

- **AugmentGeometry**

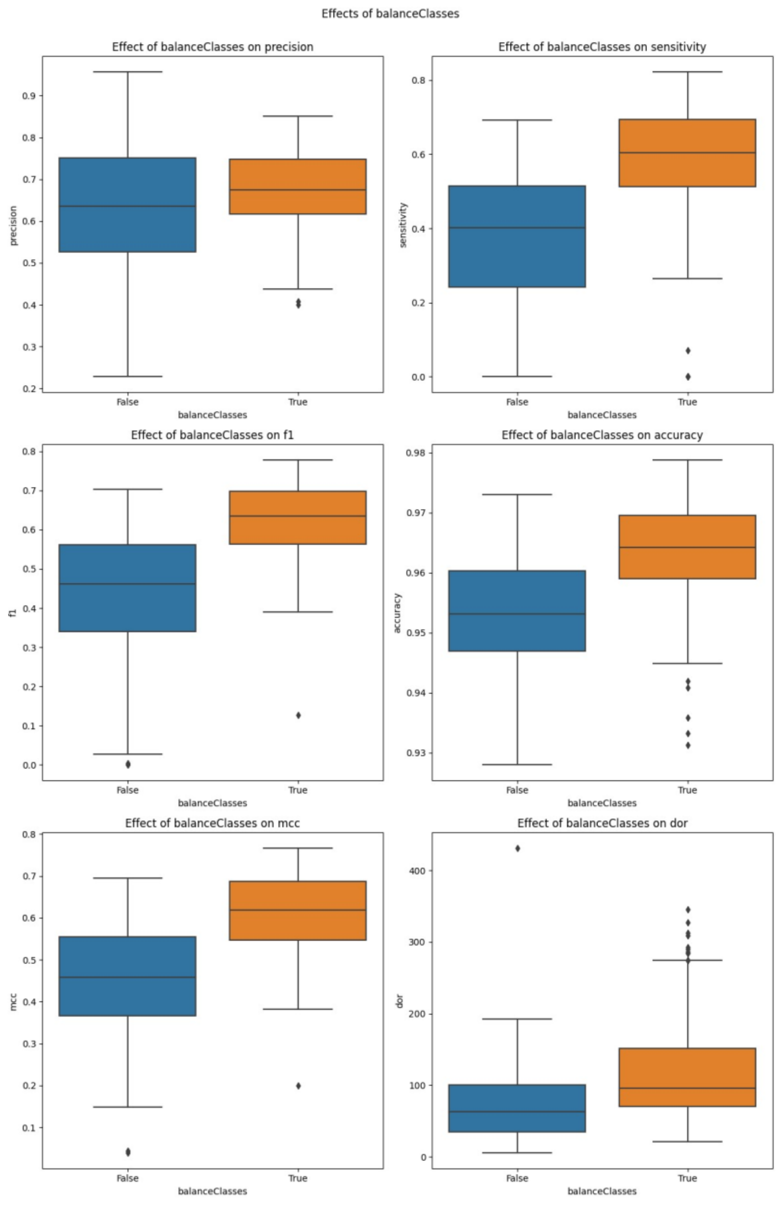


Utilizing the 'augmentGeometry' technique during the training process leads to a noticeable improvement in several key metrics, including precision, specificity, accuracy, MCC, and DOR. This enhancement is indicative of two significant improvements in the model's capabilities:

- Increased Accuracy in Lesion Prediction: The rise in precision and accuracy demonstrates the model's improved ability to correctly predict lesions. This reflects a more reliable and precise identification of the targeted abnormality.
- Enhanced Differentiation between Lesions and Other Classes: The increase in specificity, along with the MCC and DOR, signifies the model's augmented ability to distinguish lesions from other classes. This enhancement also indicates a greater capacity to correctly identify negative samples.

The collective improvement in these metrics suggests an overall enhancement in the model's performance. Therefore, incorporating 'augmentGeometry' during the training process can be considered a valuable strategy.

- **balanceClasses**



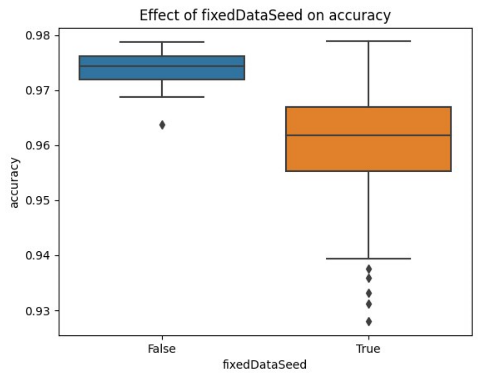
Much like the 'AugmentGeometry' method, the application of 'balanceClasses' leads to a significant increase in various evaluation metrics. However, the distinct advantage of 'balanceClasses' lies in its ability to elevate both the F1 score and sensitivity, reflecting an increase in both precision and recall specifically for the

Lesions class. This improvement is particularly substantial, as evidenced by the corresponding BoxPlot, and it aligns directly with our primary focus on the Lesion class.

Beside, the range of the “False” boxplot is wider, which means that unbalanced data will make the network unstable.

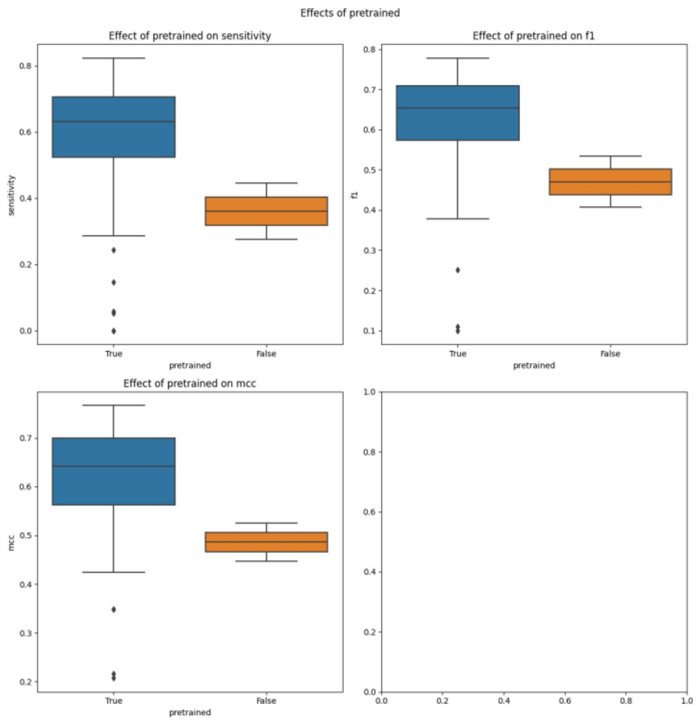
Given the critical importance of precisely classifying Lesions in our context, the employment of 'balanceClasses' during experiments is not merely beneficial but vital. It stands as a key method that complements the overall strategy, enabling a more targeted and effective approach in the study of Lesions.

- **fixedDataSeed**



FixedDataSeed can significantly decrease the accuracy of the model. I believe this might be since not fixing the seed can introduce random noise, which is beneficial to the training process.

- **Pretrained**

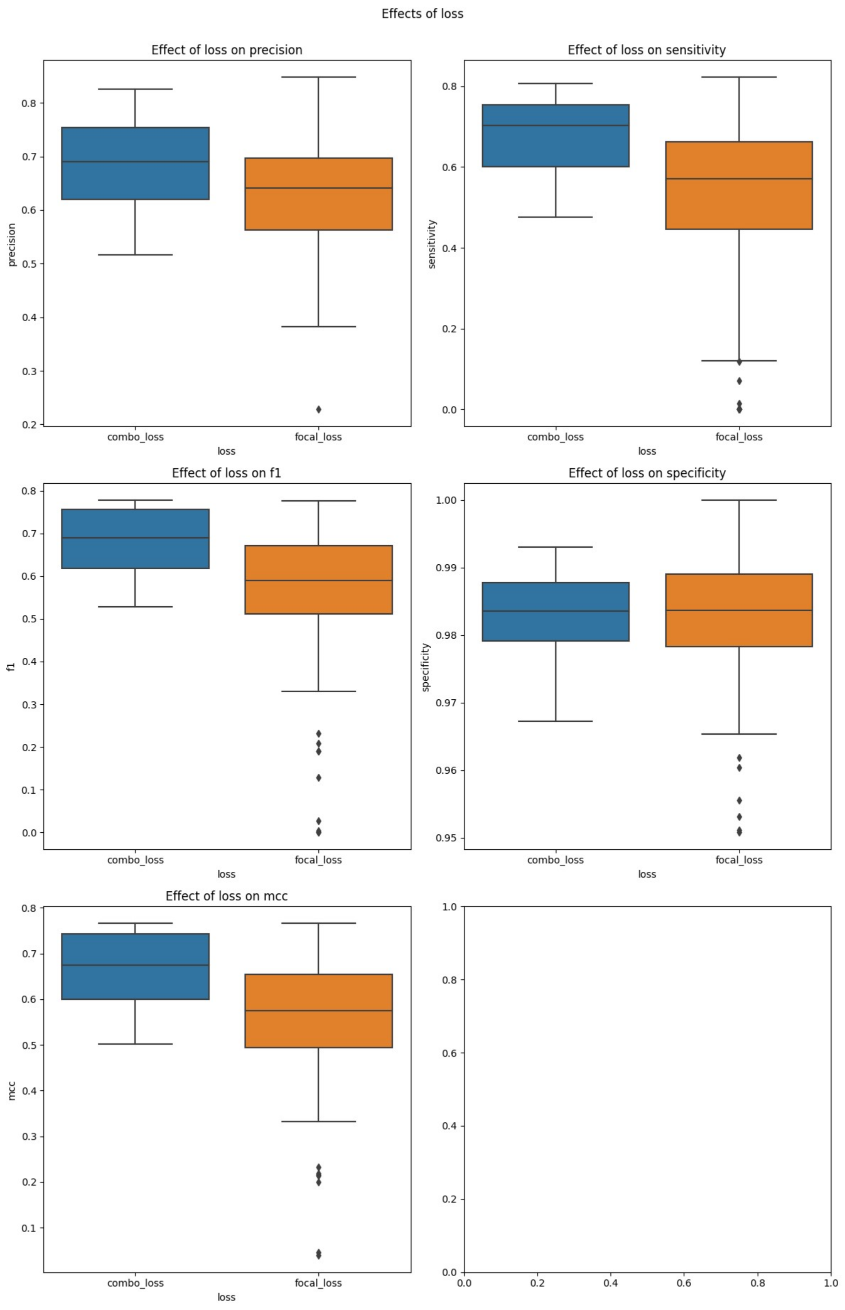


Using a pretrained model can greatly increase sensitivity and the F1 score, reflecting a substantial improvement in the ability to identify Lesions. In addition, the increase in the Matthews correlation coefficient (MCC) highlights a better balance between precision and recall within the model. The MCC is often considered a more informative metric especially for imbalanced datasets. The rise in these key metrics indicates that leveraging a pretrained model can lead to a more robust and effective learning process, enhancing the model's overall performance and its specific capability to detect and classify Lesions.

Wider range for those boxplot may be attributed to the different capability of the pretrained model.

However, the range of the boxplot is wide, it means that using pretrained model will make the training process unstable, we need to train multiple times to get a better result.

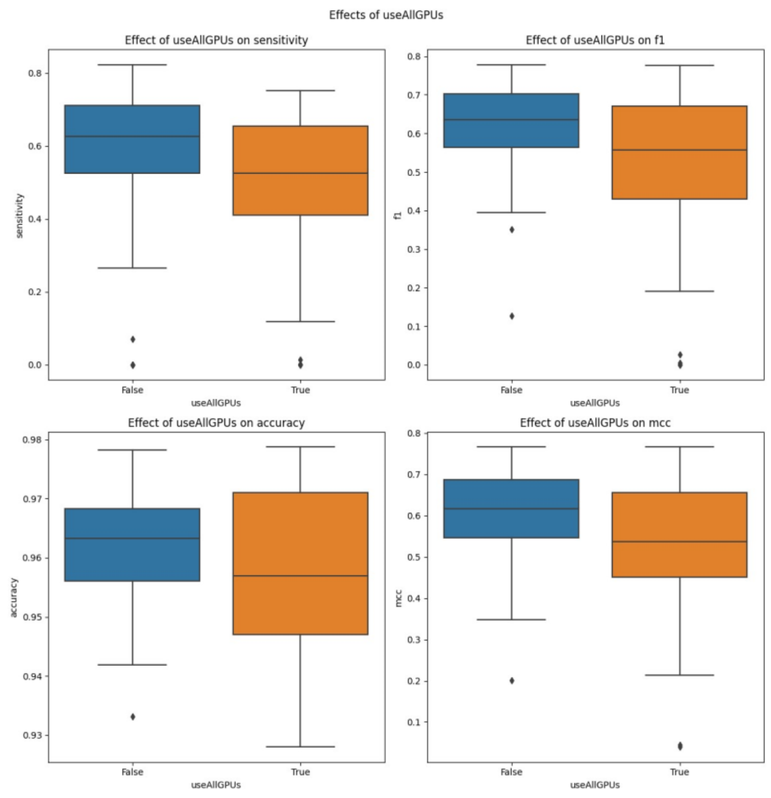
- **Loss**



The implementation of Combo Loss demonstrates an increase in key metrics such as precision, sensitivity, F1 score, and MCC. This enhancement is particularly beneficial to the Lesion class, aligning with our targeted objectives. Besides, using focal loss will make the training unstable, more outliers and wider range happened.

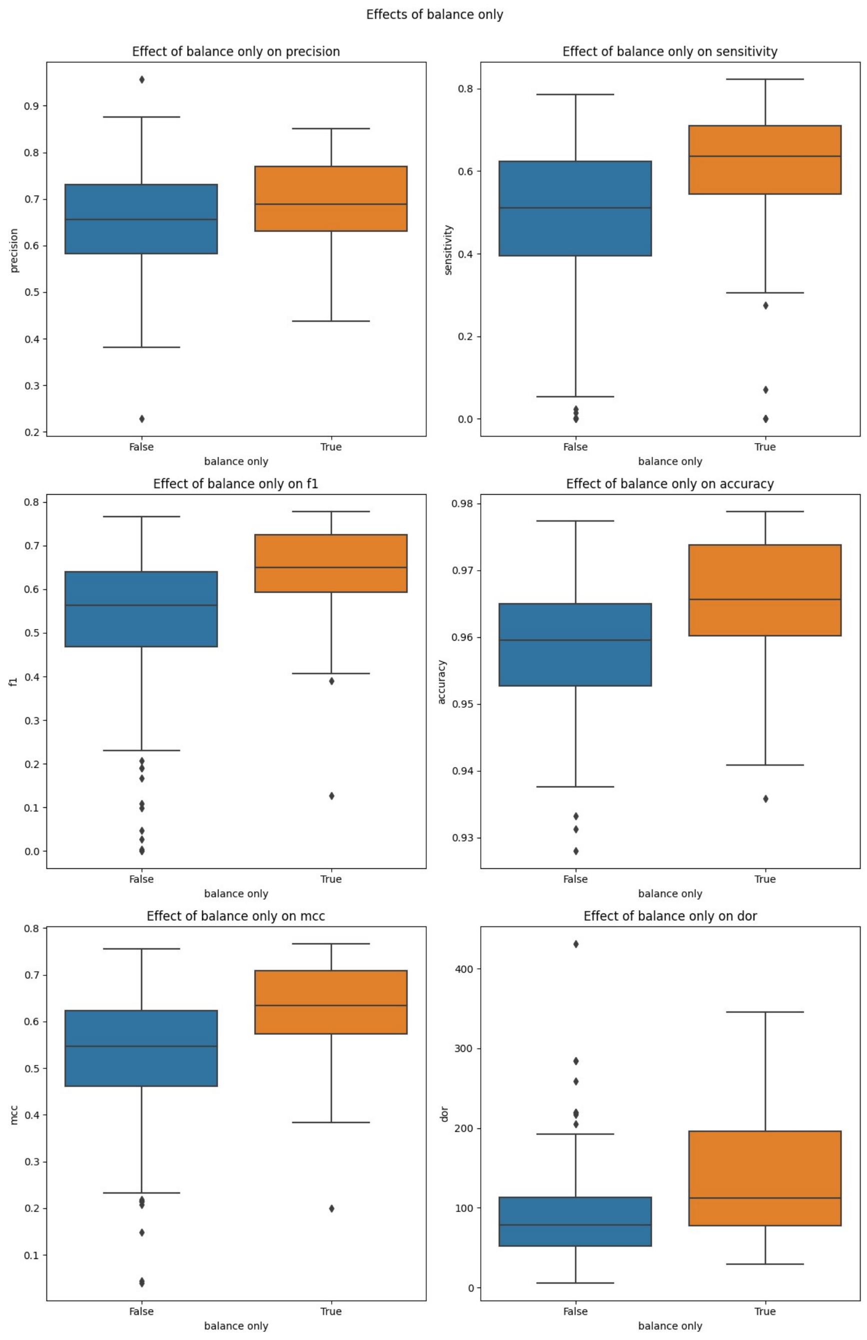
- **UseAllGPUs**

We can observe that using a single GPU leads to an increase in sensitivity, F1 score, and MCC, enhancing key evaluation metrics. Additionally, it's important to note that when utilizing all available GPUs, the range of the boxplot is notably large. This wide range may signify potential instability in the training process. Therefore, opting for a single GPU approach appears to provide a more controlled and reliable training environment, aligning more closely with our analytical goals.



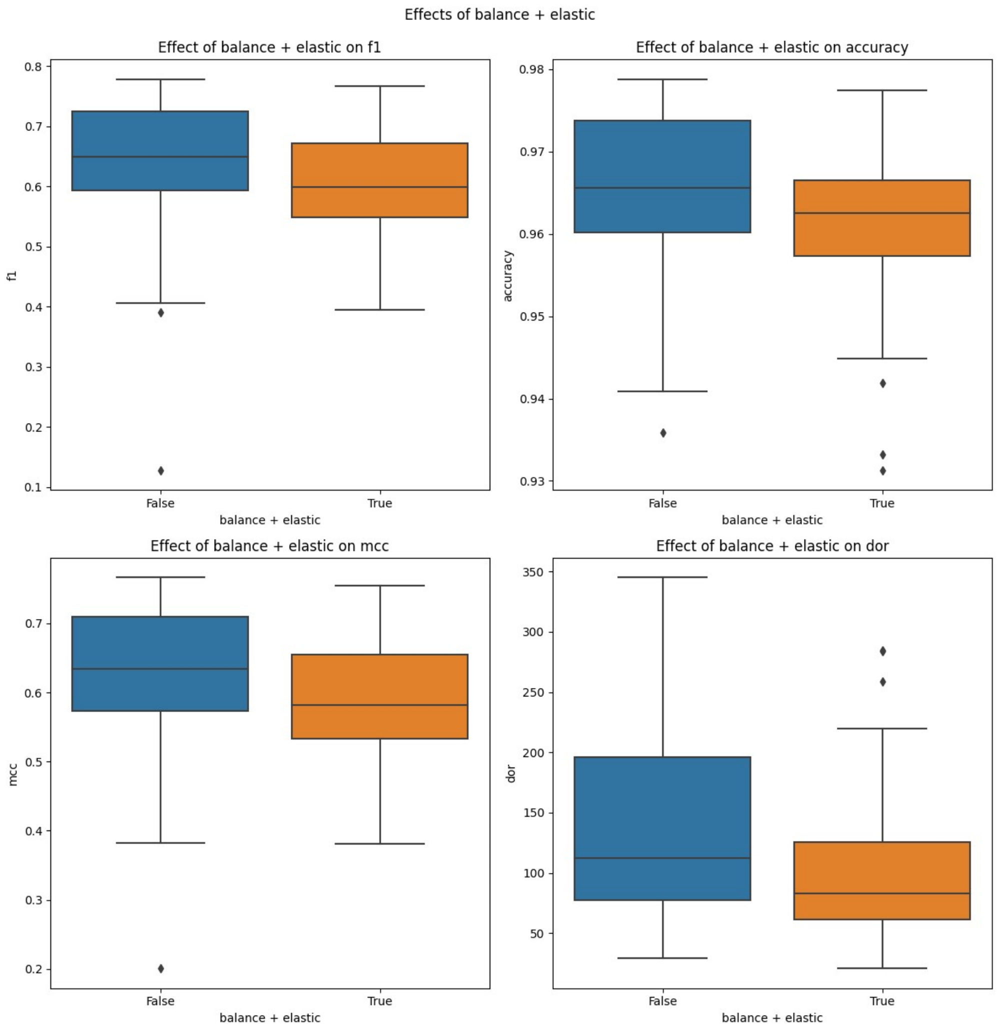
- **Balance only**

From the analysis depicted in the following figure, we notice a marked increase in multiple performance metrics, including precision, sensitivity, F1 score, accuracy, MCC, and DOR.



- **balance + elastic**

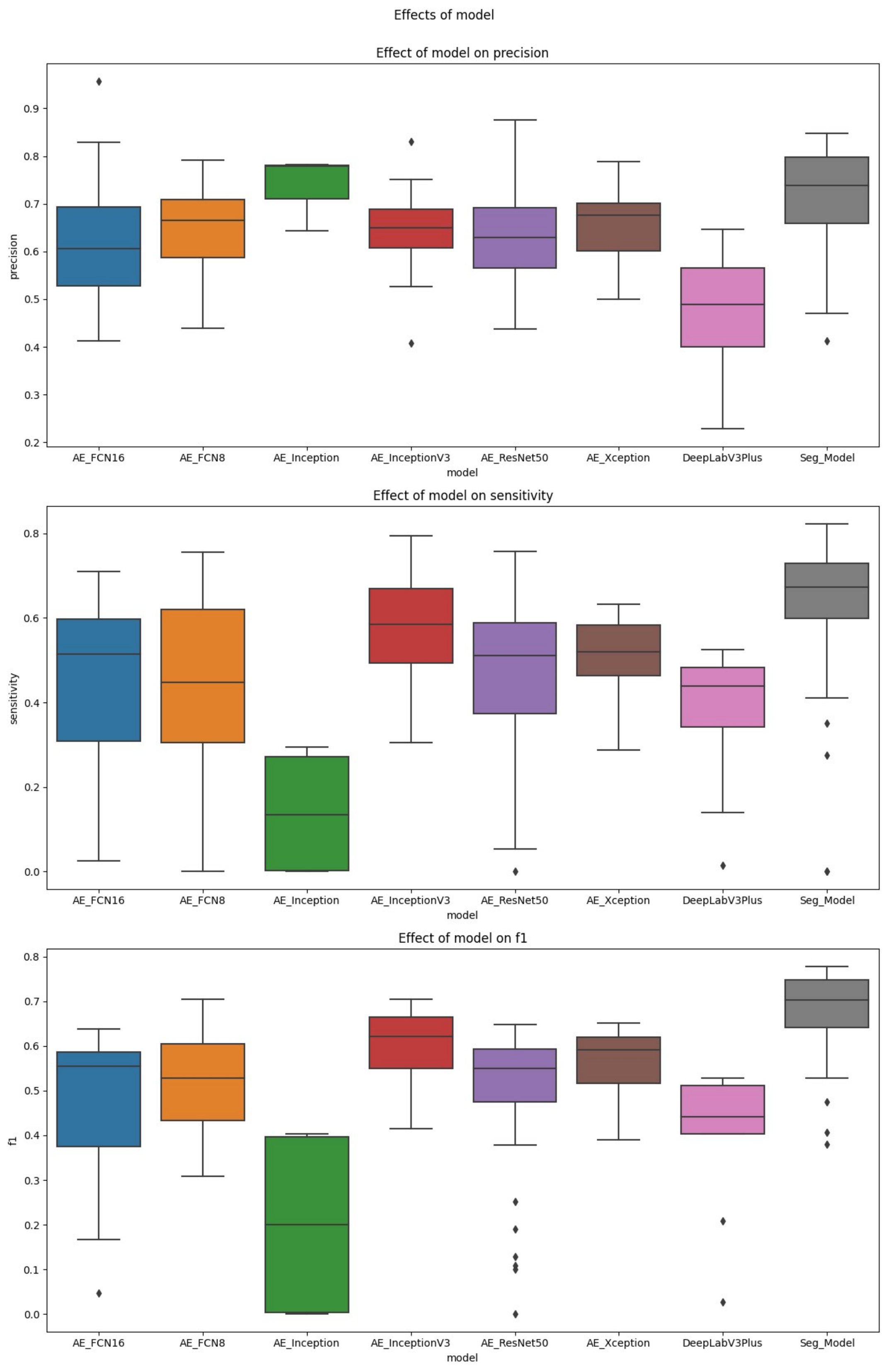
The implementation of "balance + elastic" results in a decrease in several key metrics, including F1 score, accuracy, MCC, and DOR. By comparing this outcome with the results of using "balance only", it becomes evident that the addition of "elastic" is detrimental to the overall performance of our model. Therefore, it would be advisable to avoid using elastic in conjunction with balance in this context.

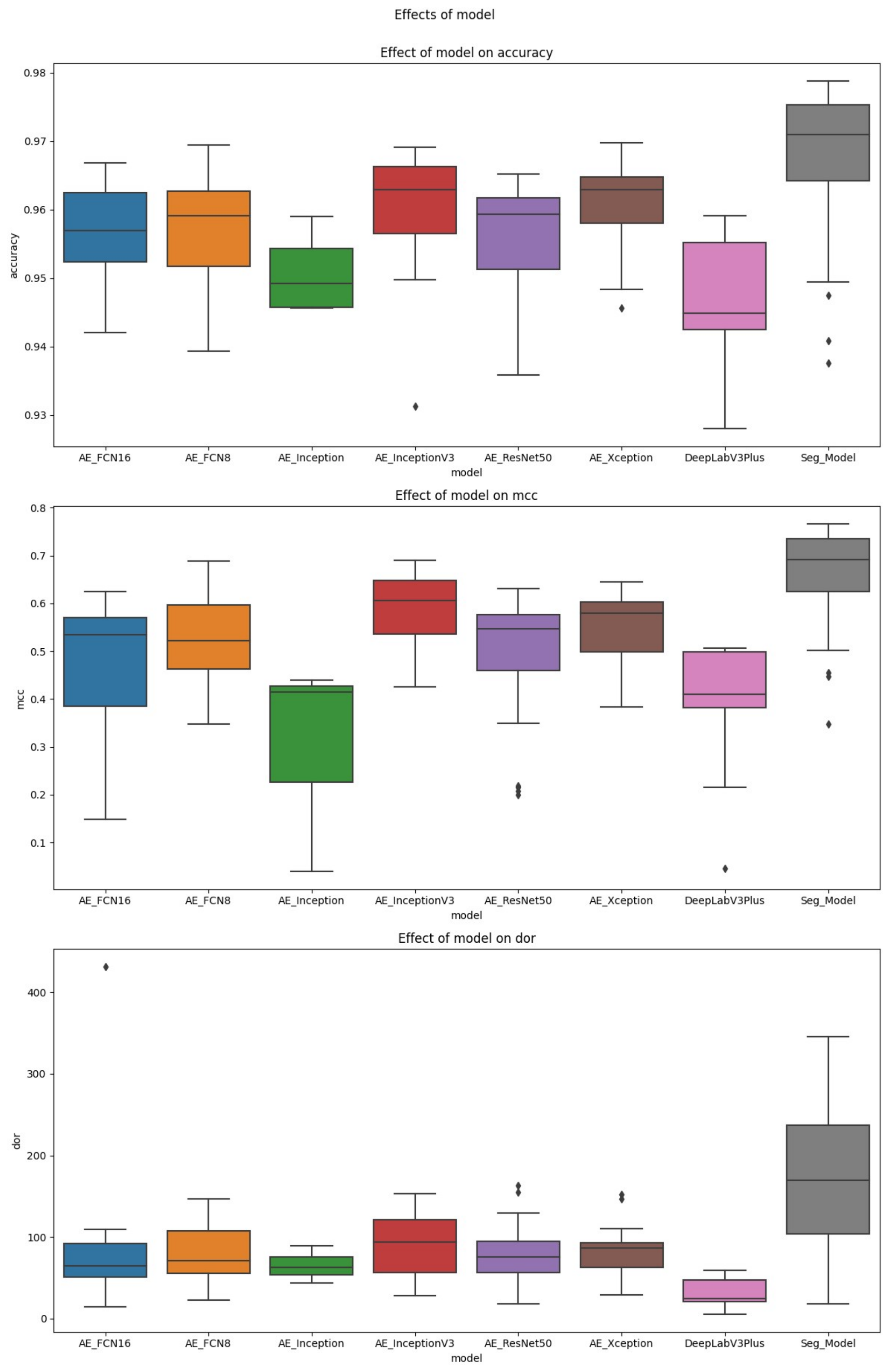


- **Model**

The Seg_Model demonstrates the most impressive performance among the models considered, boasting the highest precision, sensitivity, F1 score, accuracy, MCC, and DOR. However, it's important to note that the range of the boxplot for this model is relatively wide, indicating some instability in the results. Conversely, the performance of DeepLabV3Plus is the least satisfactory among the models examined.

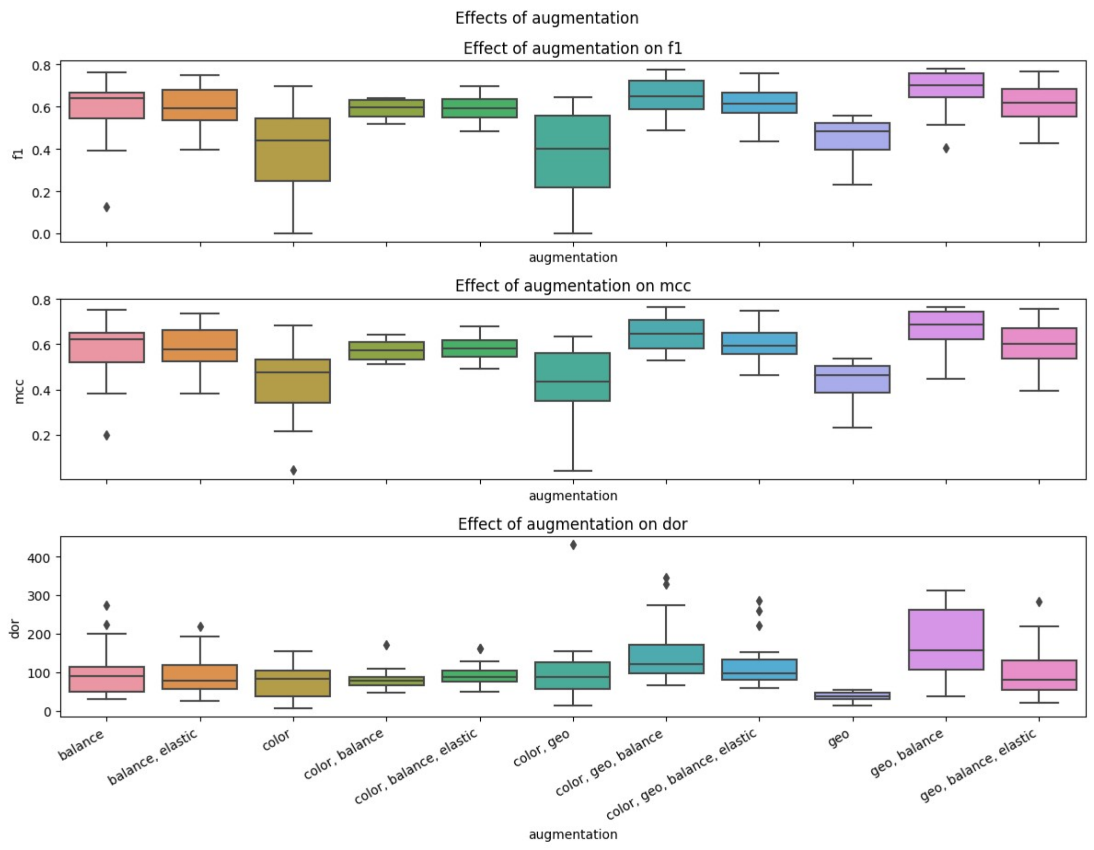
Within the AE-based modules, the Inceptive V3 backbone outshines the others, delivering the best overall performance. In contrast, the Inception model lags in almost all metrics, except for precision. If precision is the primary metric of interest for our specific goals, the Inception model might still be a viable option. This observation allows for more nuanced decision-making in the model selection process.


Effects of model
Effect of model on precision
precision
model
Effect of model on sensitivity
sensitivity
model
Effect of model on f1
f1
model
AE_FCN16
AE_FCN8
AE_Inception
AE_InceptionV3
AE_ResNet50
AE_Xception
DeepLabV3Plus
Seg_Model

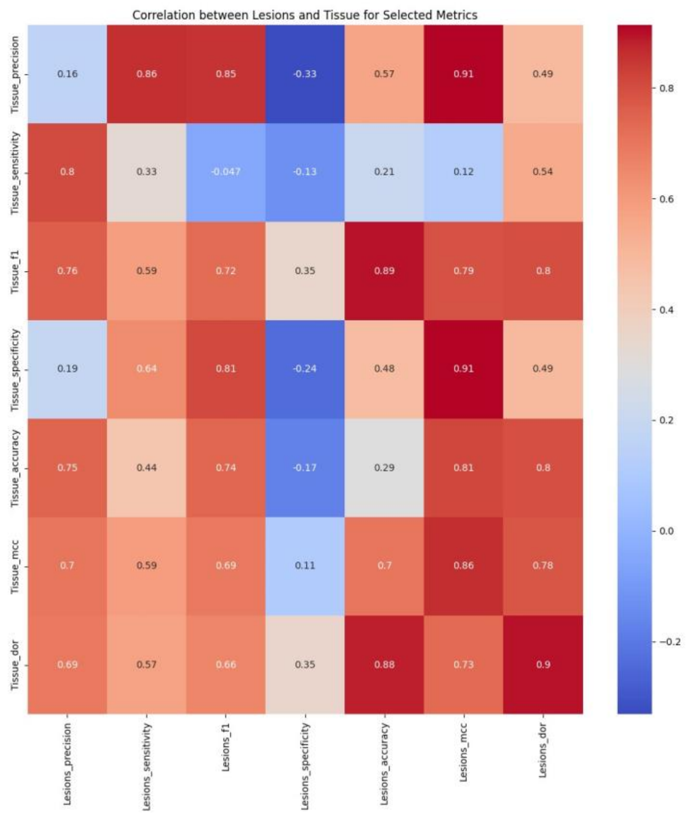


- **Augmentation**

Augmentation refers to a specific set of methods that combine various techniques, including color, balance, elastic, and geometric transformations. The findings in our analysis are entirely consistent with this understanding, confirming that the synergy between geometric transformations (geo) and balanced datasets yields the most outstanding performance. This combination seems to encapsulate the essential features that contribute to the model's effectiveness, reinforcing the conclusions we derived in the previous sections.



5. Correlation between *Tissue* and *Lesions* class result



The relationships between various metrics can be summarized as follows:

- **Lesions' Specificity:** Most metrics show a positive relationship with the tissue, except for specificity. This makes sense, as specificity measures the recall for negative classes, and a balance must be struck between identifying both positive and negative cases.
- **Lesions' Precision:** This is highly related to almost all metrics in tissue except precision and specificity itself. A higher precision for lesions means that more cases are identified as negative classes, but this will lower the precision for identifying those negative classes themselves.
- **Lesions' Sensitivity:** This shows a strong correlation with tissue-related metrics, especially tissue's precision. An increase in sensitivity for lesions means fewer lesions will be misclassified as tissue, boosting the precision for tissue identification.
- **Lesions' F1 Score:** This has a strong relationship with all metrics related to tissue except for sensitivity.
- **Lesions' Accuracy:** As a broad measure of overall model performance, accuracy shows a strong correlation with all other metrics. An increase in accuracy indicates that the model's overall capability has been enhanced.
- **Lesions' MCC:** This is highly related to all metrics concerning tissue, except sensitivity.
- **Lesions' DOR:** This shows a strong correlation with all the metrics in tissue.