

Hello,

This folder contains all the experimental data included in the "Table 2" section of the paper. Each code file has specific instructions within the file itself. Here, I will explain the execution order of the code files for the "Table 2" experiment. The "AlexNet" folder contains experimental data and code files based on the AlexNet model, the "ResNet" folder contains experimental data and code files based on the ResNet model, and the "VGGNet" folder contains experimental data and code files based on the VGGNet model. Since the operations for the code files in these three folders are similar, I will use "AlexNet" as an example for explanation. The remaining code files are common to the AlexNet, ResNet, and VGGNet models and are mainly used to calculate the scores of images generated by various methods in these three models. The "images" folder contains as many original experimental images as possible within the size limitations of the supplementary materials.

Step 1: You need to execute the code files in the "guid_xmage" folder to generate experimental images for the "guid_Xmage" method. In the "guid_xmage" folder, you need to execute the "[Calculate the Pixel Weight Matrix of Multiple Images.py](#)" file to generate the pixel weight matrices for multiple images.

Step 2: You need to execute the "[Calculate the bias of multiple images.py](#)" file to generate the biases for multiple images.

Step 3: You need to execute the "[Verify the accuracy of classification values.py](#)" file to verify the accuracy of the calculated pixel weight matrices and biases.

Step 4: You need to execute the "[generating images with increase classification values guid_Xmage.py](#)" and "[generating images with decrease classification values guid_Xmage.py](#)" files to generate experimental evaluation images for the guid_Xmage method that increase and decrease classification values, respectively.

Step 5: You need to execute the "[grad_CAM tamper.py](#)" file in the "grad_CAM" folder to generate experimental evaluation images for the grad_CAM method.

Step 6: You need to execute the "[grad_CAM.py](#)" file to generate the saliency maps using the grad CAM method.

Step 7: Similarly, for "grad_CAM++" and "score CAM", follow the similar steps in Steps 5 and 6 to generate the respective experimental evaluation images and saliency maps. Since the execution steps are basically the same, they are not repeated here.

Step 8: After generating the experimental evaluation images for each method, you can execute the "[score AlexNet decrease.py](#)" and "[score AlexNet increase.py](#)" files to calculate the scores for each method in increasing and decreasing classification values. The average of these two scores is the total score for the method on this model.