This folder contains codes used for ResNet+GG-CAM training and evaluation.

RNetCAM.py is the main file for the training and evaluation of ResNet+GG-CAM.

Python 3.8.5 is used. Packages required for RNetCAM.py are:

- PyTorch 1.8.0
- torchvision 0.9.0
- tensorboard 2.3.0
- pandas 1.1.0
- matplotlib 3.3.1
- numpy 1.19.2
- scipy 1.5.2
- sklearn 0.23.2

To run the code, simply execute RNetCAM.py through "python RNetCAM.py" in command line or with other Python IDEs. After execution, a new folder "run" will be created (if not already exists) in the execution directory. For each execution of RNetCAM.py, a new folder with a random name will be created inside folder "run" for the temporary storage of network parameters as well as recoding training details and evaluation results. The following lists the file details inside the folder:

- *params_NetLearn_.txt*, records parameters and data related to network training.
- *params_ResCAMNet_.txt*, records hyper-parameters for the network.
- *QuickHelper_summary.txt*, records other data during execution.
- *RNetCAM_S_XXXX/classification_report.json*, records classification evaluation metrics, and XXXXX are random characters generated during run time.
- *RNetCAM_S_XXXX/classification_results.csv*, records raw classification output for each test image.
- *RNetCAM_S_XXXX/NET.pt*, stores the parameters for the best performing network during or after training.
- *RNetCAM_S_XXXX/training_process.png*, visualizes the change of learning rate, losses, and validation metrics during the training process.

During execution, RNetCAM.py will also print training details and evaluation results to the console.