**Dataset Pre-processing:**

**Video Preprocessing:**

First, run the following script to reshape all RGB videos to 256x256 with 30 fps.

Python3 datasets/prepare\_ntu.py --dir=<dir of RGB videos>

**Audio Preprocessing:**

Extract all the audio files of the respective video and then create a source.txt file that contains the file name of all audio files in the npy extension and their path. Then you can run the preprocess\_1.py file to get all the numpy files.

Now keep all the videos in one directory and audio files in another directory but the same parent directory. Run the compute\_mean\_std.py file in the datasets folder to get the mean and std of all the npy files.

Create a “label2.txt” file that contains the video name along with their actual labels.

**Run Experiment:**

First, search the hypernets. You can use --parallel for data-parallel. The default setting will require about 128GB of GPU memory, you may adjust the --batch size according to your budget.

$ python main\_darts\_searchable\_ntu.py –parallel

Then train the searched fusion network. You need to assign the searching experiment by --search\_exp\_dir.

$ python main\_darts\_found\_ntu.py --search\_exp\_dir=<dir of search exp>

If you want to just run the test process (no training of the fusion network), you can also use this script, you need to assign both the searching and evaluation experiments directories.

$ python main\_darts\_found\_ntu.py --search\_exp\_dir=<dir of search exp> --eval\_exp\_dir=<dir of eval exp>

**Roc curve calculation:**

You can run the get\_auc\_val.py file to get the AUC curve.

**Explanation of the code:**

The file auc\_data.txt contains the last layer weights for 2 class classifications i.e. real and fake along with the original label. We perform softmax over the last layer values, then we use the sklearn library roc\_auc\_score() method by passing the true labels, the predicated probability labels, and the multi-class argument as “ovr” to get our AUC score.