Feature Modeling for Anomaly Detection in Neuroimaging

Denis Kovačević

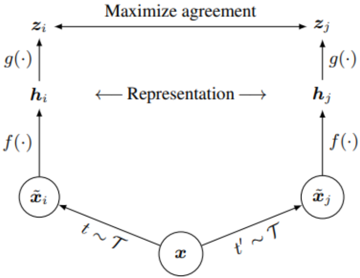
March 5, 2025

The Goal

- The model gets 2 augmented views of every data sample
- It tries to map the views from the same data sample close, and from different samples far away
- The hope is that the features from the intermediate layers will follow a Gaussian distribution
- Features that are generated from anomalous voxels should be further from the distribution

The Model

- SimCLR like approach
  - Encoder based on (3D) CNNs
  - Non-linear FC projection head
- Data augmentation for generating 2 views



Loss

- Using the same loss as in the SimCLR paper
- NT-Xent - the normalized temperature scaled cross-entropy loss
    - Augmented views from the same are considered positive, all other negative
    - Only the loss between the positive pairs is used

$$l_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{k \neq i} \exp(\mathrm{sim}(z_i, z_k)/\tau)}$$

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} (l_{2k-1,2k} + l_{2k,2k-1})$$

Feature extraction and Gaussian fitting

1. After training the projection head is discarded
2. Some of the layers are selected to extract features from them for every non-augmented data sample
3. All of the features are projected to the dimensions of one of the layers (target layer) and stacked along the channel dimension
    - For this trilinear interpolation is used
4. Fit a Gaussian for every voxel in the target layer of dimension $\sum_{i \in \text{Chosen layers}} C_i$
    - To make it efficient, a modified version of Welford's algorithm is used

Validation

- For a validation sample extract the features from the chosen layers
- Project them to the dimensions of the target layer and stack along the channel dimension
- Calculate the Mahalanobis distance for every voxel
  - The distances are not in the [0,1] range and unevenly distributed
  - Rescale them as $\hat{x} = \frac{x - \text{median}}{q_{95} - q_5}$ where q_a is the a-th quantile
  - Apply the sigmoid function to get the normalized distance (probability)
- Find the best threshold based on the dice scores for the validation set

Final results

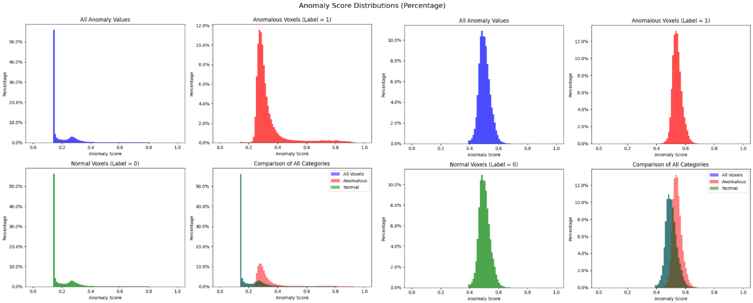
- The process is the same as for validation
- The threshold used for dice scores is the best one from validation

Problems

- So far the validation results are bad
- The target layer must be deeper (downsampled) or less layers must be used (smaller Gaussian) because of memory constraints
- Limited batch size, small compared to the classic SimCLR due to the size of MRI volumes
- Maybe the receptive field is too small

Problems

- Uneven distribution of the distances - optimizing the threshold search?

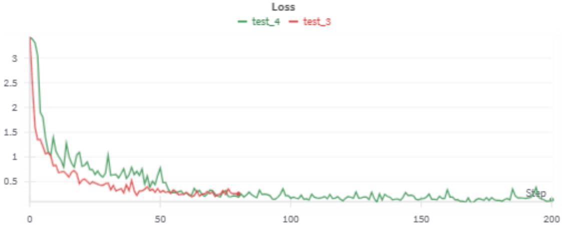


Next Steps

- Different augmentation choice - affine, elastic deformation, gamma, patch swap
- Using dilution in some of the deeper layers
- Threshold search for the dice scores?
- Changing the temperature parameter since the batch size is small
- Larger model, in SimCLR they use a ResNet - 3D ResNet?

Next Steps

- Learning rate scheduler might help - the loss decreased quickly for every model
- Making the task harder for the model, augmentations more probable and more extreme


Loss
test_4
test_3
Step