Class Activation Mapping on Diffusion MRI

Zhiyun Ling, *M.S. of Computer and Information Science and Engineering, University of Florida*

***Abstract*—This paper revisits Global Average Pooling (GAP) and Class Activation Mapping (CAM), and uses both strategy in medical diffusion tensor imaging era. CAM is widely used as object detection strategy in weakly supervised localization task on color images, such as places and animals. In this paper, CAM is transferred from place/animal classification on RGB images into age classification on brain diffusion MRI images. Six models are trained and compared using data of 26 subjects from Human Connectome Project. Several regularizing strategy were applied, such as drop-out, pooling and data augmentation, to prevent over-fitting on small dataset. With only image label provided, results demonstrate that while maintaining good classification performance, our method is able to localize the most discriminative part for different age groups to some extent.**

***Index Terms*—Weakly Object Localization, CNN, dMRI.**

I. Introduction

WITH neural network, we can train models by fitting them with large amount of data, and use the model as an inference tool in almost every field. Recently, CNNs are heavily used in image classification. And many network models are proposed, such as AlexNet [1] and GoogLeNet [2]. Recent work by Zhou et al has shown that convolutional units in CNNs actually behave as object detectors, without supervision on location of the object. [3] Before neural networks, medical image localization often requires human expert to annotate, which costs labor and time. In turn, medical imaging datasets are often small in quantity, and take long time to obtain.

For real-life images, dataset increments rapidly and many novel CNN models are proposed in competitions. In 2012, AlexNet brought deep CNN in popularity among image classification practitioners by winning ImageNet 2012. [1] As in Fig.1, VGGnet secured first place secured the first and the second places in the localization and classification tracks respectively [4] Coming to 2015, GoogLeNet introduces inception to aid in object discovery and detection. [2] Later, Inception V3 outperforms GoogLeNet by adding batch normalization and factorization. [5] In the same year, Spatial Transformer Networks enhances resistance to image transformation such as scaling and rotation, by simply introduces them on the input. [6] Afterwards, data augmentation using affine transformation on the input is common in training CNNs.

As deeper networks requires factorial increasing parameter space to represent, CNNs are slowly growing in medical imaging era. Now, with refreshing the state-of-art CNN models and more importantly, volume-increasing MRI datasets, neural networks are gradually applied in medical imaging era to assisting in treating diseases such as Alzheimer's. [7]

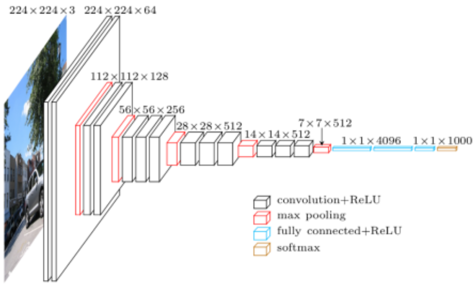

Fig. 1. Network architecture for VGG16 network

For object localization, usually it requires annotated object labels, which is even more challenging for MRI images. So, weakly object localization fits best in this era. In 2015, Zhou et al verified localization ability of Global Average Pooling (GAP) [8] and proposed Class Activation Mapping (CAM) to exploit object localization ability of CNNs. [9]

Traditionally, CNNs use fully connected layer at the end, e.g. AlexNet. [1] First, Conventional neural networks perform convolution in the lower layers. Then, the feature maps of the last convolutional layer are flattened (dense layer) and fed into multiple fully connected layers. In the end, a softmax layer handles the classification. With convolutional layer as feature extractor, fully connected layers act as a confidence map. Thus classification can be done by choosing the highest confidence class.

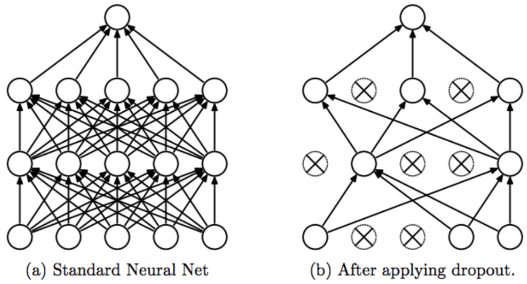

Fig. 2. Several fully connected layers and Dropout

However, as in Fig.2(a), fully connected layers are prone to over-fitting due to dense architectures. Besides, they hamper localization ability of CNNs. In 2014, Dropout is proposed as a regularizing tool for CNNs to prevent over-fitting on fully connected layers. As in Fig.2(b), Dropout randomly excludes some nodes in each layer to enforce generalization ability under skipped connections.

In this paper, we added GAP layer to six CNN models mentioned above for weakly object localization on brain diffusion MRI images. Dataset comes from 26 subjects from Human Connectome Life Span Project. [10] We are able to compare

Zhiyun Ling was with the Department of Computer and Information Science and Engineering, University of Florida Gainesville, FL, 32608 USA.

different models against same test set and visualize their localization abilities using CAM. Evaluation of localization ability is done by drawing bounding box and examining how localized ROI affects the classification result.

II. Methods

A. Related Works

Inspired by Zhou et al's paper [9], Class Activation Mapping and Global Average Pooling are used. GAP-CNN not only tells us what object is contained in the image - it also tells us where the object is in the image. Based on this property of GAP, we can express localization as a heat map (referred to as a class activation map) computation, where the color-coding scheme identifies regions that are relatively important for the GAP-CNN to perform the object identification task.

1) Global Average Pooling: As one of the pooling strategy in CNNs, GAP is inherently a regularizer. Pooling means down-sampling, which is used in CNNs for parameter reduction ability and easy implementation property. It tries to do feature selection by reducing the dimension of the input. Overfitting can be thought as fitting patterns that do not exist due to the high number of features or the low number of training examples. So by selecting a subset of features, CNNs are less likely to find false patterns.

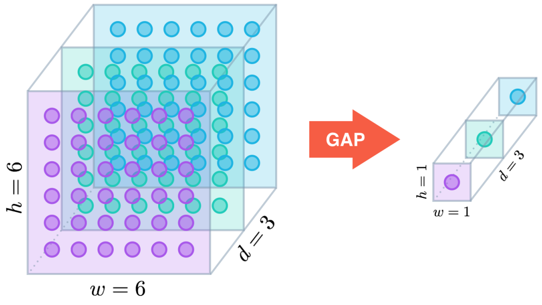

Fig. 3. Global Average Pooling

Max-pooling is the most popular one. Each time it reduces parameter by 3 quarters (2x2 kernel), by selecting the max value in a sliding window. Compared to local pooling, global pooling is more extreme, for it set kernel size the same as the feature map. So, GAP averages for each feature map. In this manor, GAP summarizes and bridges natively between convolutional layer and confidence map.

$$F_k = \sum_{x,y} f_k(x,y) \tag{1}$$

,wheref_k is the output from previous convolutional layer and F_k is the confidence score for feature unit k.

Using GAP instead of fully connected layer gives many benefits. a) Pooling introduces translational tolerance. By choosing only the max or average value in a grid, small translation on the object location can be neglected. b) Overfitting in GAP layer is exterminated. By using input size as kernel size and averaging, no weights are learnable in GAP layer. c) GAP is more native to convolution structure. Through enforcing correspondence between feature maps and catagories/classes, GAP interprets feature maps in last layer as confidence maps for each class. [8]

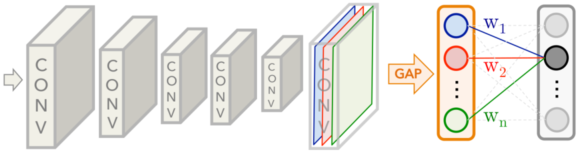

Fig. 4. Class Activation Mapping

2) Class Activation Mapping: A class activation map for a particular class means the discriminative regions used by the CNN to identify that class. Firstly, our networks are mostly convolutional layers, just like GoogLeNet and Network-in-Network. Then, a GAP layer is added between the last convolutional layer and the output layer. This connection is simple and many information remains intact. Based on that, we can decide the importance of regions, by projecting back the weights of the output layer onto the convolutional feature maps.

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y}\sum_k w_k^c f_k(x,y) \tag{2}$$

,where w_k^c is the weight of feature unit k for class c, and S_c is the confidence score for class c across all feature maps. When firstly summarizing over same point (x,y) across all feature maps, we can define the importance of each location for class c.

$$M_c(x,y) = \sum_k w_k^c f_k(x,y) \tag{3}$$

Thus, by un-sampling to input image size, we can generate the heatmap (CAM) representing importance of each location.

In this paper, we use two age groups as classification target. In the meantime, we use CAM to localize the most discriminative region between the two age groups in brain MRI images.

B. Dataset visualization and augmentation

The dataset we use is from Human Connectome Life Span Project. [10] It contains 3T diffusion MRI scans of 26 subjects from 5 age groups. For each subject, diffusion data contains two phase types (LR,RL) and two direction counts (dir80,dir81). Aside from raw diffusion images, images incorporated with gradient distortion correction are provided. The biggest difference in applying CAM is that diffusion tensor images are 3D images, while normally CNNs models deal with 2D colored images. The data we have is in the format of 4D volume, which mean 3D voxel images from all 80 or 81 directions are stacked together.

In order to apply CNNs on these data, we start by preprocessing using DiPY [11] to load and convert to fractional anisotropy map (3D). Then, a 2.5D representation is used for FA maps.

1) 2.5D Representation: Observe that we have very small dataset with 26 subjects, so we need to deal with over-fitting. Aside from using data augmentation and regularizer, we take multiple slices from FA maps. First, we start by transferring

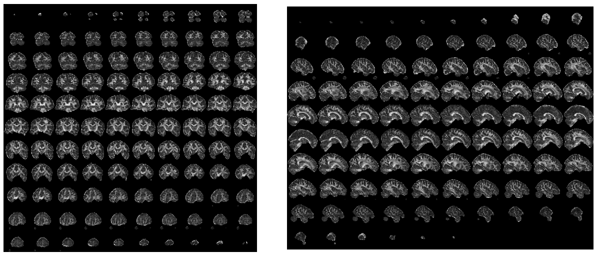

Fig. 5. Visualization of one subject in coronal (left) and sagittal (right) directions.

FA maps from voxel coordinates into world coordinates. Then, we slice 16 center images from each coordinate direction. Last, we do this for all types of configurations for each subject. In the end, we get 26 subjects x 2 phases x 2 direction counts x 2 preprocessing level x 16 slices x 3 directions = 9984 images (2D). In the manor, we represent 3D FA maps using 2D images, which can then be fed into CNNs.

2) Data Augmentation: Applying selected affine transformation on input is proven to be helpful for small dataset. [6] Observe diffusion images have two phases, LR and RL, so our network models should be tolerant to horizontal flipping. Also, gaussian noise is randomly added to training data to make our model robust to noise. Resizing and cropping the input randomly helps with finding most significant region of images. Besides, random erasing some region of the input also benefits the generalization of classification. [12]

C. Modification

Alongside with models used in Zhou et al's work, we add an improved inception network named Inception V3. It adds Batch Normalization and Factorization to GoogLeNet.

1) Batch Normalization: Batch Normalization normalize the input layer by adjusting and scaling the activations. Usually we perform normalization on input to accelerate learning. When applying the same strategy to hidden layers, we get 10 times or more improvement in the training speed. [13] Batch normalization reduces the amount by what the hidden unit values shift around (covariance shift). Also, batch normalization allows each layer of a network to learn by itself more independently of other layers. Thus, we can use higher learning rates because batch normalization. Besides, it reduces over-fitting because it has a slight regularization effects. Similar to dropout, it adds some noise to each hidden layers activations.

Factorization also speeds up training process.

2) Implementation: Instead of using MATLAB and caffee platform to implement the networks, we use PyTorch, which is gaining popularity recent years. Besides, the transferring into and from other models are easier in this platform. The code implementation for this project is host on Github: github.com/zhiyunl/CAM-dMRI

III. Results

A. Dataset

First, all subjects are preprocessed using strategy define above. Then, 5 age groups subjects is divided into two groups, with young:8-15 and old:25-75. Among all 9984 images, 8448 is taken as training data and 1536 is kept as test data. During training, data augmentation is used intensively to prevent over-fitting. Besides, 20% of training data is used as cross-validation to ensure a trustable classification result.

B. Model Configuration

Observe that higher localization ability is attained when last convolutional layer has higher spatial resolution. So, inputs are resized into 224x224 grey images during preprocessing, with batch size at 16. We modify all original networks AlexNet, VGGnet, GoogLeNet, InceptionV3 to have about 14x14 at last convolutional layer. Specifically, we remove layers:
1) Alexnet: after conv5. ⇒ resolution 13x13
2) VGGnet: after conv5-3. ⇒ resolution 14x14
3) GoogLeNet: after inception4e. ⇒ resolution 14x14
4) InceptionV3: after inception4e. ⇒ resolution 14x14
Then, add one convolutional layer with kernel 3x3, stride=1, pad=1 with 1024 units, followed by GAP and softmax layer. Besides, AlexNet with batch normalization and GoogLeNet with Global Max Pooling are added with similar setting.

C. Training Parameters

Loss function is cross entropy, so no need to convert class label to one-hot encoding. Activation function is ReLU, as is used in original networks. Regularization method is dropout. The optimizer is SGD (Stochastic Gradient Descent), with learning rate at 0.001 and momentum at 0.9.

D. Metrics

1) Classification: Compare accuracy curves and loss curves during training and validation for different models. Most importantly, test set are used for final examination of all network models. Since test set is never seen during training. So the final classification score on test set for each model is reliable. Fig.6 shows the classification results. Accuracy on test set is shown in Table I. (AlexNet with batch normalization performs bad in validation, so the result of it is not shown.) Result shows that GoogLeNet, GoogLeNet-GMP and AlexNet give best result on classification.

2) Localization: Since we don't have annotated object labels for each images, we couldn't compare with baseline. So, firstly, we compare across all models for their localization heatmaps. Secondly, bounding box covering largest connected component with 20% threshold is generated for each model. Lastly, we add perturbation to localization region to verify by evaluating how much it affects classification. As is shown in Fig 7, VGG and AlexNet produce best localization bounding box across all models. Due to the time constraints, some of our models haven't converged completely, especially VGGnet, as it trains much slower than others.

3) Verification: Through adding Gaussian noise to our localized region, we are able to see a great decline in classification accuracy. But translational perturbation affects very little. Since CNNs are tolerant to some extent of translational transformation, this phenomenon is understandable.

TABLE I
Classification Accuracy on Validation and Test Set

ACC%	AlexNet	VGG	GoogLeNet	GoogLeNet-GMP	InceptionV3	AlexNet-BN
Validation	98.08	95.16	99.86	99.75	97.03	NaN
Test	**96.03**	89.9	**94.53**	**94.92**	88.41	77

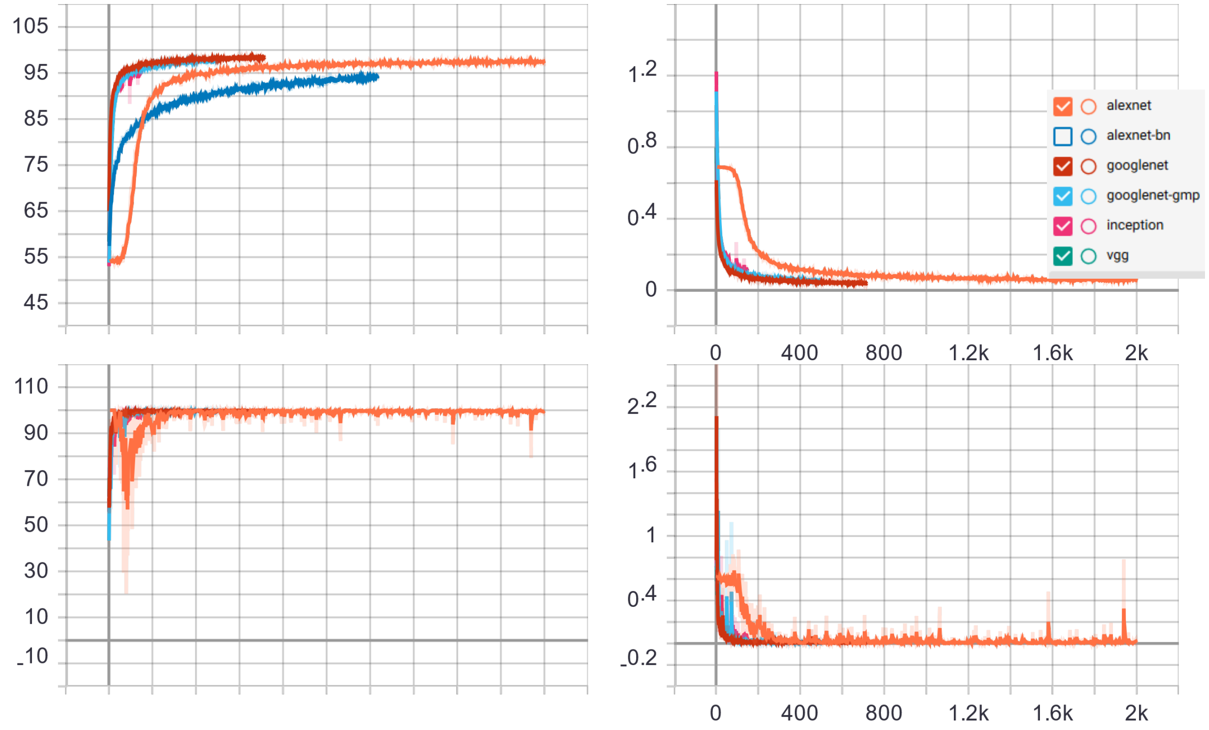

Fig. 6. Training accuracy (top-left) and loss (top-right), Validation accuracy (bottom-left) and loss (bottom-right).

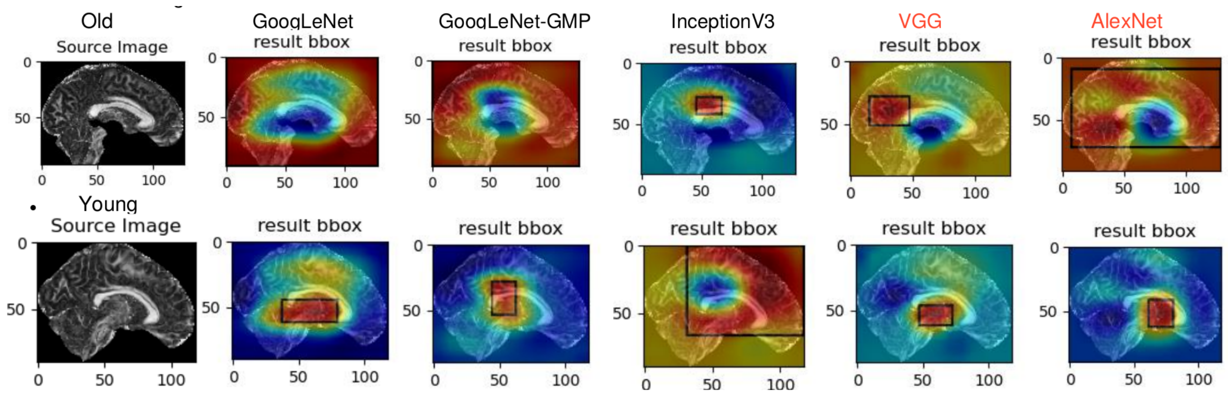

Fig. 7. Same Image with Different Models, source image: young (top) and old (bottom).

E. Analysis

Summarizing the results, we can find that both AlexNet and VGG has much more parameters that other networks. More parameter contributes to more complicated networks, thus can representing more detailed discriminative features. Besides, all inception networks are tend to separate the image by two complimentary regions. But VGG successfully localizes exact regions for each class. This may be caused by the property of two class classifier. When discriminating two groups, it's easy to form complimentary metrics. A way to improve would be adding more age groups to classifier. More over, dataset size is limited, which may still cause over-fitting to some extent.

IV. Conclusion

With many CNN models trained on a small diffusion MRI dataset, we are able to do weakly supervised object localization to some extent. VGGnet and AlexNet outperform others in localization ability, when trained on two class discrimination. Above discussion and experiments provides some insight of how Class Activation Mapping can be used in dMRI data. Various data augmentation strategy and regularizing tools are used to overcome the insufficient dataset. Still, many improvements and further study can be done in this era. The limitation of CAM is that only last convolutional layer can be visualized. This can be improved using grad-CAM or grad-CAM++. Also, other localization strategies can be compared with our models to get a better evaluation. And, 3D-CNN techniques used in video detection era can possibly be utilized to classify 3D MRI images directly. Lastly, more data is always helpful for CNNs. Further study for this paper could be in above directions.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[3] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision. 2015," *arXiv preprint arXiv:1512.00567*, 2015.

[6] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[7] F. Liu and C. Shen, "Learning deep convolutional features for mri based alzheimer's disease classification," *arXiv preprint arXiv:1404.3366*, 2014.

[8] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[9] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization. corr abs/1512.04150 (2015)," *arXiv preprint arXiv:1512.04150*, 2015.

[10] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium *et al.*, "The wu-minn human connectome project: an overview," *Neuroimage*, vol. 80, pp. 62–79, 2013.

[11] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. Van Der Walt, M. Descoteaux, and I. Nimmo-Smith, "Dipy, a library for the analysis of diffusion mri data," *Frontiers in neuroinformatics*, vol. 8, p. 8, 2014.

[12] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.