MProtoNet: A Case-Based Interpretable Model for Brain Tumor Classification with 3D Multi-parametric Magnetic Resonance Imaging

Author name(s) withheld EMAIL(S) WITHHELD
Address withheld

Editors: Under Review for MIDL 2023

Abstract

Recent applications of deep convolutional neural networks (CNNs) in medical imaging raise questions about their interpretability since high-stake decisions are being made upon these inherently complex models. While most of the current explainable deep learning approaches use post hoc methods to generate feature attribution maps after training, there is a new type of case-based reasoning models originating from natural images, namely ProtoPNet and its variants, which identify prototypes during training and compare the input image patches with those prototypes. We propose the first medical prototype network (MProtoNet) to extend ProtoPNet to applications on multi-modal 3D medical images, in this case, brain tumor classification with 3D multi-parametric magnetic resonance imaging (mpMRI) data. To address different requirements between 2D natural images and 3D mpMRIs especially in terms of localizing attention regions, a new attention module with soft masking and online-CAM loss is introduced. Soft masking helps sharpen attention maps, while online-CAM loss directly utilizes image-level labels when training the attention module. MProtoNet achieves statistically significant improvements in interpretability metrics of both correctness and localization coherence without human-annotated labels during training, when compared with baseline CNN and several ProtoPNet variants. The proposed MProtoNet can be easily extended to other 3D medical image applications to generate attribution maps and provide case-based prototypical explanations.

Keywords: Interpretable Deep Learning, Explainable AI, Brain Tumor Classification

1. Introduction

With the development of architecture design and training strategy of very deep neural networks, models such as deep convolutional neural networks (CNNs) have achieved state-of-the-art performance in image classification, object detection, semantic segmentation and the respective applications in medical imaging (Zhou et al., 2021). However, the recent successful applications of CNNs in medical imaging raise questions about their interpretability since high-stake decisions are being made upon these inherently complex models (Rudin, 2019). Specifically, CNNs are known to be vulnerable when learning some undesirable shortcut connections during training (Geirhos et al., 2020). The special requirements in medical settings make the evaluation of interpretability necessary since the predictions of CNNs are not always based on intended evidence (DeGrave et al., 2021).

Most explainable deep learning applications, including those in medical imaging, use post hoc methods to generate feature attribution maps after training of CNNs. Representative methods include the well-known class activation mapping (CAM) (Zhou et al., 2016) and its

© 2023 Anonymous.

more general variant GradCAM (Selvaraju et al., 2020). These post hoc methods nevertheless are criticized for providing localization-only and often unreliable explanations (Adebayo et al., 2018; Rudin, 2019). There are also some attempts to incorporate interpretable blocks in the design of CNNs, such as case-based and concept-based reasoning models. Compared to concept-based interpretable models (Koh et al., 2020) that require predefined concepts, case-based interpretable models (Chen et al., 2019) can identify prototypes during training and compare similarities between the input image patches and the identified prototypes, resulting in both attribution maps and case-based prototypical explanations.

Related Work ProtoPNet (Chen et al., 2019) is the first case-based interpretable deep learning model. While there are many attempts to further improve the performance of ProtoPNet on multi-class natural image classification (Wang et al., 2021; Rymarczyk et al., 2021, 2022; Donnelly et al., 2022), the investigation of ProtoPNet and its variants on medical imaging applications is still relatively limited. Barnett et al. (2021) evaluates a ProtoPNet variant (IAIA-BL) on breast lesion classification with 2D digital mammography images, with additional costs of fine-grained annotations from clinicians. XProtoNet (Kim et al., 2021) is another application on chest radiography with 2D X-ray images, but there are no quantitative evaluation results on interpretability.

Contributions Our contributions can be summarized as follows: (1) We propose the first medical prototype network (MProtoNet) to apply case-based interpretable deep learning to brain tumor classification using 3D multi-parametric magnetic resonance imaging (mpMRI) data which present additional challenges compared to both natural images (Chen et al., 2019; Donnelly et al., 2022) and 2D medical images (Barnett et al., 2021; Kim et al., 2021), especially in terms of localizing attention regions. (2) A new attention module is proposed to enhance the localization of attention regions, along with soft masking that sharpens attention maps and online-CAM loss that directly utilizes image-level labels during training. (3) Interpretability metrics of correctness and localization coherence are employed throughout all experiments to evaluate the interpretability of the proposed MProtoNet and other compared models. The source code will be made publicly accessible upon acceptance.

2. Methods

2.1. Dataset

We use the multimodal brain tumor image segmentation (BraTS) (Menze et al., 2015) 2020 dataset[1] to validate our methods throughout this work. The BraTS 2020 dataset contains 369 subjects with pathologically confirmed diagnoses: 293 with high-grade glioma (HGG) and 76 with low-grade glioma (LGG). Each subject has mpMRI data of four modalities: T1-weighted, T1-weighted contrast enhancement (T1CE), T2-weighted and T2 fluid attenuated inversion recovery (FLAIR). Originally, for each subject in BraTS 2020 there are three tumor sub-regions labeled. In this work, we unify all sub-region labels as the single whole tumor (WT) region and only use this information when evaluating the interpretability metrics. The preprocessing and data augmentation details are described in Appendix A. After preprocessing, each image has a size of $128\times128\times96$ voxels ($1.5\times1.5\times1.5$ mm^3 resolution).

1. https://www.med.upenn.edu/cbica/brats2020/

2.2. Architecture

The overall architecture of MProtoNet is shown in Figure 1. It consists of four layers: a feature layer, a localization layer, a prototype layer, and a classification layer.

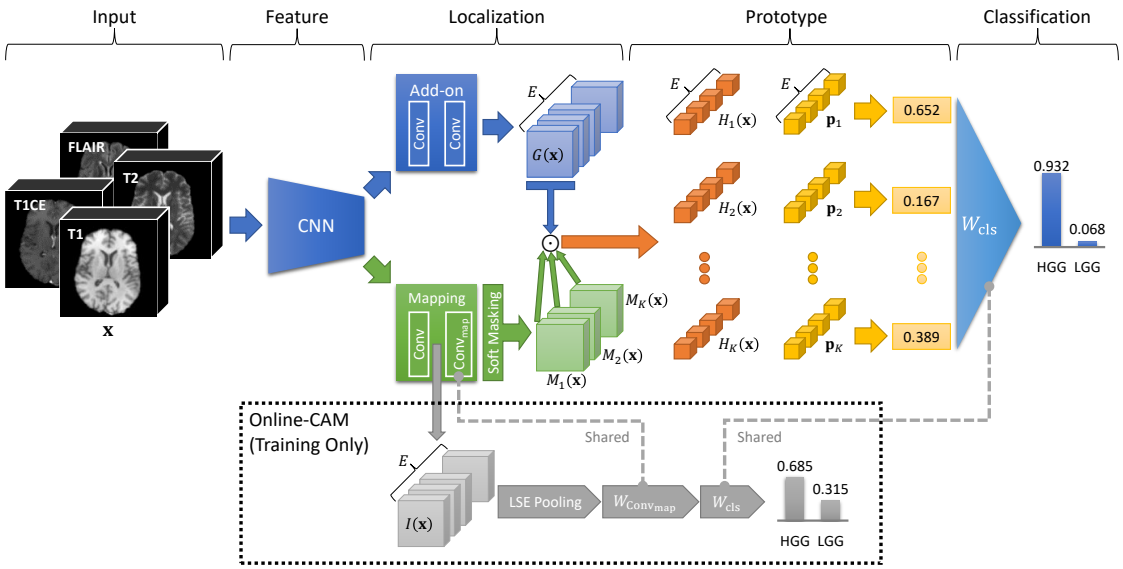

Figure 1: The overall architecture of MProtoNet.

Inference Process All layers are described in detail below. At a high level, MProtoNet delivers its prediction by comparing processed features with a set of learned prototypes $\mathbf{p}_k \subseteq \mathbf{P}$ $(k \in \{1 \dots K\})$. Given input mpMRI data $\mathbf{x}$, the feature layer extracts the embedding features $F(\mathbf{x})$. After that, $F(\mathbf{x})$ is sent to both an add-on module and a mapping module in the localization layer to get high-level features $G(\mathbf{x})$ and an attention map $M_k(\mathbf{x})$ for each prototype $\mathbf{p}_k$. The dot-product result between $G(\mathbf{x})$ and each $M_k(\mathbf{x})$ is treated as the final features $H_k(\mathbf{x})$ to be compared with each prototype $\mathbf{p}_k$. In the prototype layer, a similarity score $s(\mathbf{x}, \mathbf{p}_k)$ between each pair of $H_k(\mathbf{x})$ and $\mathbf{p}_k$ is calculated and sent to the classification layer to get the final prediction.

Feature Layer The feature layer is a CNN module that performs feature extraction on the input mpMRI $\mathbf{x} \in \mathbb{R}^{M \times H \times W \times D}$, where M is the number of modalities and H, W, D are height, width and depth of each modality. ResNet (He et al., 2015) is chosen as the backbone for its good performance on a wide variety of image tasks and relatively low model complexity. Since the original ResNet is designed with 2D operations (convolutions, batch normalizations, etc.), we directly replace them with 3D ones to make a 3D version of ResNet. Due to the relatively small input size ($128 \times 128 \times 96$) and the need for localization in the subsequent layer, we only use the layers up to the second building block of ResNet-152, resulting in a final output size of $16 \times 16 \times 12$. The output $F(\mathbf{x}) \in \mathbb{R}^{N \times 16 \times 16 \times 12}$ (where $N = 512$) is then sent to the localization layer. Because it is difficult to find a suitable pre-trained version of 3D ResNet, the feature layer is trained from scratch.

Localization Layer The localization layer has two branches that receive the same input $F(\mathbf{x})$ from previous layer: an add-on module and a mapping module. Similar to ProtoPNet (Chen et al., 2019), the add-on module consists of two [1×1×1 convolution – batch normalization – activation] combinations so as to extract high-level embedding features $G(\mathbf{x}) \in \mathbb{R}^{E\times16\times16\times12}$, where E is the number of embedding channels. The mapping module (modified from XProtoNet (Kim et al., 2021)) has similar components, and its outputs $M_k(\mathbf{x}) \in \mathbb{R}^{16\times16\times12}$ $(k \in \{1\dots K\})$ are considered as attention maps for localizing regions related to the corresponding prototypes $\mathbf{p}_k$ $(k \in \{1\dots K\})$. The values of the attention maps $M_k(\mathbf{x})$ are scaled into a range of $[0,1]$. To learn more accurate and sharper attention maps and reduce irrelevant background areas, we introduce differentiable soft masking (Li et al., 2018) to sharpen the raw values of the attention maps $M_k^0(\mathbf{x})$ right after activation

$$M_k(\mathbf{x}) = \frac{1}{1+\exp\left(-\omega\left(M_k^0(\mathbf{x})-\sigma\right)\right)}, \tag{1}$$

where ω and σ are hyper-parameters that are respectively set to be 10 and 0.5 (following Li et al. (2018)). The final features $H_k(\mathbf{x}) \in \mathbb{R}^{E\times1\times1\times1}$ $(k \in \{1\dots K\})$ are obtained through

$$H_k(\mathbf{x}) = \frac{1}{16\times16\times12}\sum_{h,w,d} G_{h,w,d}(\mathbf{x})\cdot M_{k,h,w,d}(\mathbf{x}), \tag{2}$$

where h, w and d are height, width and depth of the corresponding features. They represent the highly-activated regions of high-level features for the respective prototype $\mathbf{p}_k$.

Prototype and Classification Layers The prototype layer stores a set of learned prototypes $\mathbf{p}_k \subseteq \mathbf{P}$ $(k \in \{1\dots K\})$, where $\mathbf{p}_k \in \mathbb{R}^{E\times1\times1\times1}$. The common cosine similarity is used for the calculation of similarity scores. The similarity score between each pair of $H_k(\mathbf{x})$ and $\mathbf{p}_k$ is calculated as

$$s(\mathbf{x},\mathbf{p}_k) = \frac{H_k(\mathbf{x})\cdot\mathbf{p}_k}{\|H_k(\mathbf{x})\|\,\|\mathbf{p}_k\|}. \tag{3}$$

Then, the similarity scores $s(\mathbf{x},\mathbf{p}_k) \subseteq \mathbf{S}$ $(k \in \{1\dots K\})$ are multiplied by the weight matrix $W_{\text{cls}} \in \mathbb{R}^{K\times2}$ in the classification layer to output the finally predicted probabilities for HGG/LGG, which are then normalized using the softmax function.

2.3. Training and Loss Functions

The training of MProtoNet consists of three stages: (1) training of layers before the classification layer; (2) prototype reassignment; (3) training of the classification layer. See Appendix B for details as they are the same as those employed in previous ProtoPNet variants. The following loss functions are used during these training stages: classification loss $\mathcal{L}_{\text{cls}}$, cluster loss $\mathcal{L}_{\text{clst}}$, separation loss $\mathcal{L}_{\text{sep}}$, mapping loss $\mathcal{L}_{\text{map}}$, online-CAM loss $\mathcal{L}_{\text{OC}}$, and L1-regularization loss $\mathcal{L}_{\text{L1}}$.

Classification Loss Because of the imbalanced distribution of class $c \in \{\text{HGG},\text{LGG}\}$ (HGG:LGG = 293:76), we introduce focal loss (Lin et al., 2018) and class weights to our classification loss, similar to that in XProtoNet (Kim et al., 2021)

$$\mathcal{L}_{\text{cls}} = \sum_i \frac{1}{N^c}\left(1-p^c(\mathbf{x}_i)\right)^\gamma \log p^c(\mathbf{x}_i), \tag{4}$$

where i indexes the training samples, N^c denotes the total number of training samples in the class c, $p^c(\mathbf{x}_i)$ denotes the predicted probability of the correct class c for the training sample $\mathbf{x}_i$ and γ is a hyper-parameter that is set to be 2 (following Kim et al. (2021)).

Cluster and Separation Losses K prototypes $\mathbf{p}_k \subseteq \mathbf{P}$ are evenly assigned to each class $c \in \{\text{HGG}, \text{LGG}\}$, which means prototypes $\mathbf{p}_k \subseteq \mathbf{P}^c$ are trained to represent embedding features related to class c. Since all layers are randomly initialized, the cluster and separation losses from ProtoPNet (Chen et al., 2019) and XProtoNet (Kim et al., 2021) are introduced to learn an embedding space for prototypes $\mathbf{p}_k \subseteq \mathbf{P}^c$

$$\mathcal{L}_{\text{clst}} = \sum_i \frac{1}{N^c} \max_{\mathbf{p}_k \subseteq \mathbf{P}^c} s(\mathbf{x}_i, \mathbf{p}_k), \quad \mathcal{L}_{\text{sep}} = -\sum_i \frac{1}{N^c} \max_{\mathbf{p}_k \nsubseteq \mathbf{P}^c} s(\mathbf{x}_i, \mathbf{p}_k). \tag{5}$$

The cluster loss $\mathcal{L}_{\text{clst}}$ encourages the embedding features to get close to prototypes of the same class, while the separation loss $\mathcal{L}_{\text{sep}}$ encourages them to stay away from prototypes of different classes.

Mapping Loss The mapping loss from XProtoNet (Kim et al., 2021) is also introduced to act as strong regularization for our attention maps $M_k(\mathbf{x}) \in \mathbb{R}^{16\times16\times12}$ $(k \in \{1 \dots K\})$

$$\begin{aligned} \mathcal{L}_{\text{affine}} &= \sum_i \frac{1}{16\times16\times12} \sum_{\mathbf{p}_k \subseteq \mathbf{P}} \left\| A(M_k(\mathbf{x}_i)) - M_k(A(\mathbf{x}_i)) \right\|_1, \\ \mathcal{L}_{\text{map}} &= \mathcal{L}_{\text{affine}} + \sum_i \frac{1}{16\times16\times12} \sum_{\mathbf{p}_k \subseteq \mathbf{P}} \left\| M_k(\mathbf{x}_i) \right\|_1, \end{aligned} \tag{6}$$

where $A(\cdot)$ denotes an affine transformation. $A(\cdot)$ is set to be a rescaling function with the ratio randomly sampled from $\{0.75, 0.875\}$ (following Kim et al. (2021)). While $\mathcal{L}_{\text{affine}}$ ensures the robustness of the attention map generation process over affine transformations of the input, the latter part of the mapping loss $\mathcal{L}_{\text{map}}$ encourages the generated attention map to be as compact as possible.

Online-CAM Loss In addition to soft masking and the mapping loss, an online-CAM loss is proposed here to further assist localization of the attention maps during training. Through directly utilizing image-level labels during training of the attention module, online-CAM modules (Fukui et al., 2019; Ouyang et al., 2021) have shown great potential in improving the localization performance of CNNs. Since the last convolution operation in the mapping module of the localization layer Conv_{map} is responsible for generating the attention maps, the intermediate features $I_e(\mathbf{x}) \subseteq \mathbf{I}(\mathbf{x})$ $(e \in \{1 \dots E\}$, $I_e(\mathbf{x}) \in \mathbb{R}^{16\times16\times12})$ right before the last convolution operation are made the feature sources of online-CAM (as shown in Figure 1). To obtain a class prediction directly from $I_e(\mathbf{x})$, instead of employing average or max pooling, we choose log-sum-exp (LSE) pooling (Pinheiro and Collobert, 2015; Sun et al., 2016) which has shown improved convergence and localization performance

$$\text{LSE}(I_e(\mathbf{x})) = \frac{1}{r} \log \left[\frac{1}{16\times16\times12} \sum_{h,w,d} \exp(r \cdot I_{e,h,w,d}(\mathbf{x})) \right], \tag{7}$$

where h, w and d are height, width and depth of the corresponding features; r is a hyper-parameter that is set to be 10 (following Sun et al. (2016)). Then, $\text{LSE}(\mathbf{I}(\mathbf{x})) \in \mathbb{R}^E$ is

consecutively multiplied by the weight matrix $W_{\text{Conv}_{\text{map}}} \in \mathbb{R}^{E \times K}$ of Conv_{map} and the weight matrix $W_{\text{cls}} \in \mathbb{R}^{K \times 2}$ in the classification layer, to obtain yet another predicted probability $p_{\text{OC}}(\mathbf{x}_i)$ (which is different from the predicted probability $p(\mathbf{x}_i)$ in the classification layer). $\mathcal{L}_{\text{OC}}$ is modified from $\mathcal{L}_{\text{cls}}$ (Equation (4))

$$\begin{aligned} p_{\text{OC}}(\mathbf{x}_i) &= \text{Softmax}\left(\text{LSE}\left(\mathbf{I}(\mathbf{x}_i)\right) W_{\text{Conv}_{\text{map}}} W_{\text{cls}}\right), \\ \mathcal{L}_{\text{OC}} &= \sum_i \frac{1}{N^c} \left(1 - p^c_{\text{OC}}(\mathbf{x}_i)\right)^\gamma \log p^c_{\text{OC}}(\mathbf{x}_i). \end{aligned} \tag{8}$$

While the online-CAM loss $\mathcal{L}_{\text{OC}}$ is useful during training, the corresponding predictions $p_{\text{OC}}(\mathbf{x}_i)$ are not performed during inference.

L1-Regularization Loss The L1-regularization loss $\mathcal{L}_{\text{L1}}$ is introduced to ensure the sparsity of the weight matrix $W_{\text{cls}} \in \mathbb{R}^{K \times 2}$ in the classification layer. The sparsity encourages the final predictions to rely on as few prototypes as possible.

3. Experiments and Results

3.1. Experimental Setup

In our experiments, since the class distribution in BraTS 2020 is highly imbalanced, we use the balanced accuracy (BAC) that is the macro-average of the recall scores of the two classes HGG and LGG to evaluate the classification performance. For the interpretability metrics, we focus on correctness and localization coherence as described in Nauta et al. (2022). The correctness of a model is evaluated by the normalized area under the incremental deletion curve (Nauta et al., 2022), called the incremental deletion score (IDS). IDS measures how precisely an activation map reflects a model's decision-making process (lower is better). The localization coherence is evaluated by the activation precision (AP) (Barnett et al., 2021). AP measures the proportion of an activation map that intersects with the human-annotated label. See Appendix C for detailed definitions of IDS and AP.

To test a model's performance under different training/test splits, stratified 5-fold cross-validation is applied throughout all experiments. The number of embedding channels E is set to be 128 and the number of prototypes K is set to be 30 (15 for each class). See Appendix D for details of the hyper-parameters during training.

There are three models compared in our experiments: CNN with GradCAM (Selvaraju et al., 2020), ProtoPNet (Barnett et al., 2021; Chen et al., 2019) and XProtoNet (Kim et al., 2021). For fair comparisons across models, all of them are reimplemented to share the same experimental settings as MProtoNet, except for the distinctions specified below.

CNN (with GradCAM) The baseline CNN is built based on submodules of MProtoNet: the feature layer, the add-on module in the localization layer, and the classification layer. These submodules are marked as blue in Figure 1. The add-on module and the classification layer (with $W_{\text{cls}} \in \mathbb{R}^{E \times 2}$) are directly connected with a global average pooling layer. GradCAM is used to generate the activation map $M(\mathbf{x})$ from the CNN after training.

ProtoPNet Compared to MProtoNet, the reimplemented ProtoPNet lacks the mapping module in the localization layer and all associated loss functions. The similarity scores

$s(\mathbf{x}, \mathbf{p}_k)$ are calculated by directly comparing each location in $G(\mathbf{x})$ with $\mathbf{p}_k$, followed by top-α average pooling (where $\alpha = 1\%$) (Barnett et al., 2021). The average of the intermediate results right before top-α average pooling is used as the activation map $M(\mathbf{x})$.

XProtoNet Compared to MProtoNet, the major components that are not in the reimplemented XProtoNet are soft masking and the online-CAM loss.

3.2. Results

Table 1 shows all quantitative results from the 5-fold cross-validation experiments, including three versions of MProtoNet (A: without online-CAM loss, B: without soft masking, C: complete version). The results represent the contributions of all modalities since the fusion of modalities occurs at the very beginning. Given that XProtoNet is most similar to MProtoNet, 5-fold cross-validated paired Student's t-tests (null hypothesis: paired results have identical means) against XProtoNet are performed on all results for MProtoNet A/B/C.

Table 1: A summary of all results in the form of [mean ± standard deviation] from 5-fold cross-validation. Bold indicates the best performance. For MProtoNet A/B/C, * indicates $p < 0.05$ in the paired t-test against XProtoNet and ** indicates $p < 0.01$. Keys: AM – attention map, SM – soft masking, OC – online-CAM, BAC – balanced accuracy, IDS – incremental deletion score, AP – activation precision.

Model	Condition			Classification	Interpretability	
	AM	SM	OC	BAC	IDS	AP
CNN (with GradCAM)				0.865±0.026	0.112±0.049	0.099±0.030
ProtoPNet				0.868±0.032	0.609±0.164	0.007±0.001
XProtoNet	✓			**0.870±0.021**	0.170±0.041	0.203±0.030
MProtoNet A	✓	✓		0.868±0.050	0.150±0.088	0.568±0.125**
MProtoNet B	✓		✓	0.865±0.015	0.103±0.020	0.204±0.028
MProtoNet C	✓	✓	✓	0.858±0.048	**0.079±0.034***	**0.713±0.058****

Comparing classification performance (BAC), there are no statistically significant differences among all models, most notably between MProtoNet A/B/C and XProtoNet. This is desirable as the same 3D ResNet backbone is shared across all models and appears to be well-suited for this task. The introduction of the interpretable components in our MProtoNet does not necessarily improve or hurt the classification performance. However, one of the most valuable outcomes is that it can provide case-based prototypical explanations during inference, as shown in Figure 2.

The complete version MProtoNet C achieves the best performance in terms of both correctness (IDS) and localization coherence (AP), excelling in interpretability. Specifically, both soft masking and the online-CAM loss are very important for statistically significant improvements in IDS ($p < 0.05$) and AP ($p < 0.01$) over XProtoNet. Since we do not use any annotation information during training, the poor performance of the original ProtoPNet (even with top-α average pooling) is expected as previously shown in Barnett et al. (2021). See Appendix E for some visualization examples of the localization coherence results.

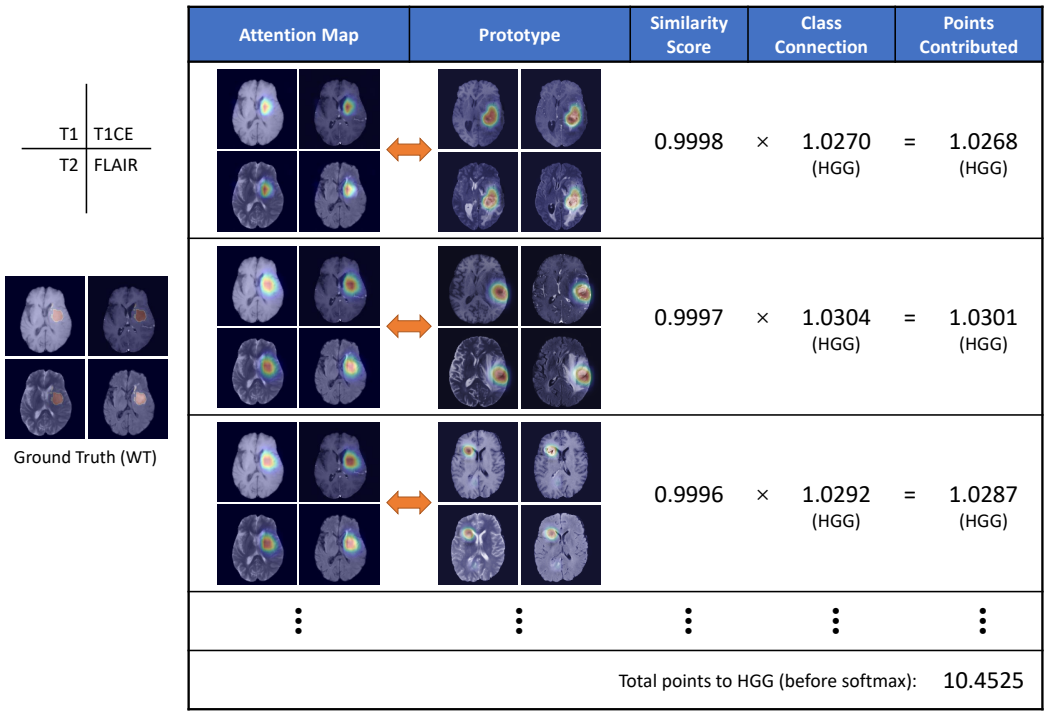

Figure 2: Demonstration of the case-based reasoning in MProtoNet.

4. Conclusions and Future Work

In this work, we propose MProtoNet to apply case-based interpretable deep learning to brain tumor classification with 3D mpMRIs, targeting a goal of precisely localizing attention regions. The proposed new attention module with soft masking and online-CAM loss helps MProtoNet achieve the best performance on interpretability in terms of both correctness and localization coherence. Meanwhile, the classification accuracy is on par with baseline CNN and other ProtoPNet variants specifically designed for medical images. No annotation information is required during training, showing the great potential of MProtoNet on 3D medical image applications wherein fine-grained annotation labels are difficult to obtain. The ability to simultaneously output attribution maps and case-based prototypical explanations also makes MProtoNet a potential alternative to current 3D CNNs in many 3D medical imaging tasks.

There are several potential aspects for future improvements over MProtoNet. Firstly, rather than fixed assignments of prototypes before and after training, a dynamic or even shared assignment (Rymarczyk et al., 2021, 2022) might be more suitable for some medical image applications. Secondly, instead of fusing the multiple modalities at the beginning, we shall also test fusion at the end to analyze each individual modality like some gradient-based methods (Jin et al., 2021). Lastly, potential combinations with other interpretable methods such as concept-based models (Koh et al., 2020) might further improve MProtoNet's interpretability on much more complicated medical image applications.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, Montreal, Canada, December 2018. Curran Associates, Inc. URL https://arxiv.org/abs/1810.03292.

Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y. Lo, and Cynthia Rudin. A case-based interpretable deep learning model for classification of mass lesions in digital mammography. *Nature Machine Intelligence*, 3 (12):1061–1070, December 2021. doi: 10.1038/s42256-021-00423-x.

Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32, Vancouver, Canada, December 2019. Curran Associates, Inc. URL https://arxiv.org/abs/1806.10574.

Alex J. DeGrave, Joseph D. Janizek, and Su-In Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, July 2021. doi: 10.1038/s42256-021-00338-7.

Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable ProtoPNet: An interpretable image classifier using deformable prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, July 2022. URL https://arxiv.org/abs/2111.15000.

Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10697–10706, Long Beach, United States, June 2019. IEEE. doi: 10.1109/CVPR.2019.01096.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. doi: 10.1038/s42256-020-00257-z.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, December 2015. URL https://arxiv.org/abs/1512.03385.

Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, February 2021. doi: 10.1038/s41592-020-01008-z.

Weina Jin, Xiaoxiao Li, and Ghassan Hamarneh. One map does not fit all: Evaluating saliency map explanation on multi-modal medical images, July 2021. URL https://arxiv.org/abs/2107.05047.

Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. XProtoNet: Diagnosis in chest radiography with global and local explanations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15714–15723, Nashville, United States, June 2021. IEEE. doi: 10.1109/CVPR46437.2021.01546.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348, Virtual, July 2020. PMLR. URL https://arxiv.org/abs/2007.04612.

Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, Salt Lake City, United States, June 2018. IEEE. doi: 10.1109/CVPR.2018.00960.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, February 2018. URL https://arxiv.org/abs/1708.02002.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, June 2022. URL https://arxiv.org/abs/2201.03545.

Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elizabeth Gerstner, Marc-André Weber, Tal Arbel, Brian B. Avants, Nicholas Ayache, Patricia Buendia, D. Louis Collins, Nicolas Cordier, Jason J. Corso, Antonio Criminisi, Tilak Das, Hervé Delingette, Çağatay Demiralp, Christopher R. Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andac Hamamci, Khan M. Iftekharuddin, Raj Jena, Nigel M. John, Ender Konukoglu, Danial Lashkari, José António Mariz, Raphael Meier, Sérgio Pereira, Doina Precup, Stephen J. Price, Tammy Riklin Raviv, Syed M. S. Reza, Michael Ryan, Duygu Sarikaya, Lawrence Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos A. Silva, Nuno Sousa, Nagesh K. Subbanna, Gabor Szekely, Thomas J. Taylor, Owen M. Thomas, Nicholas J. Tustison, Gozde Unal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koen Van Leemput. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, October 2015. doi: 10.1109/TMI.2014.2377694.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *arXiv:2201.08164 [cs]*, January 2022. URL https://arxiv.org/abs/2201.08164.

Xi Ouyang, Srikrishna Karanam, Ziyan Wu, Terrence Chen, Jiayu Huo, Xiang Sean Zhou, Qian Wang, and Jie-Zhi Cheng. Learning hierarchical attention for weakly-supervised

chest X-ray abnormality localization and diagnosis. *IEEE Transactions on Medical Imaging*, 40(10):2698–2710, October 2021. doi: 10.1109/TMI.2020.3042773.

Pedro O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1713–1721, Boston, United States, June 2015. IEEE. doi: 10.1109/CVPR.2015.7298780.

Torsten Rohlfing, Natalie M. Zahr, Edith V. Sullivan, and Adolf Pfefferbaum. The SRI24 multichannel atlas of normal adult human brain structure. *Human Brain Mapping*, 31 (5):798–819, 2010. doi: 10.1002/hbm.20906.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. doi: 10.1038/s42256-019-0048-x.

Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. ProtoPShare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, Singapore, August 2021. ACM. doi: 10.1145/3447548.3467245.

Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *Computer Vision – ECCV 2022*, 2022. doi: 10.1007/978-3-031-19775-8_21.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, February 2020. doi: 10.1007/s11263-019-01228-7.

Chen Sun, Manohar Paluri, Ronan Collobert, Ram Nevatia, and Lubomir Bourdev. ProNet: Learning to propose object-specific boxes for cascaded neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3493, Las Vegas, United States, June 2016. IEEE. doi: 10.1109/CVPR.2016.379.

Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 875–884, Montreal, Canada, October 2021. IEEE. doi: 10.1109/ICCV48922.2021.00093.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, Las Vegas, United States, June 2016. IEEE. doi: 10.1109/CVPR.2016.319.

S. Kevin Zhou, Hayit Greenspan, Christos Davatzikos, James S. Duncan, Bram van Ginneken, Anant Madabhushi, Jerry L. Prince, Daniel Rueckert, and Ronald M. Summers. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, 109(5): 820–838, May 2021. doi: 10.1109/JPROC.2021.3054390.

Appendix A. Preprocessing and Data Augmentation

A.1. Preprocessing

All mpMRI data in BraTS 2020 go through a standardized preprocessing pipeline including co-registration to a common anatomical template (SRI24 (Rohlfing et al., 2010)), resampling to a $1\times1\times1$ mm^3 resolution, and skull-stripping. After original preprocessing, each image has a size of $240\times240\times155$ voxels ($1\times1\times1$ mm^3 resolution). The images are cropped around the borders to $192\times192\times144$ voxels, and then further down-sampled to $128\times128\times96$ voxels ($1.5\times1.5\times1.5$ mm^3 resolution) to enable efficient training on GPUs. After that, intensity normalization is performed on each mpMRI image of a specific modality by subtracting the individual mean and dividing by the individual standard deviation.

A.2. Data Augmentation

During training, we follow the online data augmentation pipeline of nnU-Net (Isensee et al., 2021), which has already been successfully applied to many 3D medical imaging tasks, to overcome the limited size issue of the training data. The augmentations are applied stochastically in the following sequence: (1) rotation and scaling; (2) Gaussian noise; (3) Gaussian blur; (4) brightness augmentation; (5) contrast augmentation; (6) simulation of low resolution; (7) gamma augmentation; (8) mirroring. Detailed parameters for each augmentation operation can be found in Isensee et al. (2021).

Appendix B. Training Stages

Stage-1: Training of Layers Before the Classification Layer During the first training stage, we train all layers except the classification layer to obtain a meaningful embedding space for the prototypes. The overall loss function is

$$\mathcal{L}_{\text{stage-1}} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{clst}}\mathcal{L}_{\text{clst}} + \lambda_{\text{sep}}\mathcal{L}_{\text{sep}} + \lambda_{\text{map}}\mathcal{L}_{\text{map}} + \lambda_{\text{OC}}\mathcal{L}_{\text{OC}}, \tag{9}$$

where λ_{cls}, λ_{clst}, λ_{sep}, λ_{map} and λ_{OC} are coefficients of respective losses.

Stage-2: Prototype Reassignment To achieve the goal of case-based reasoning, i.e., to be able to view each prototype as a specific patch of a training sample, we need to replace the representation of each prototype $\mathbf{p}_k \subseteq \mathbf{P}$ ($k \in \{1 \dots K\}$) with the nearest processed feature $H_k(\mathbf{x})$ (without any online data augmentation for $\mathbf{x}$) after several iterations in stage-1. Therefore, we can treat the corresponding training sample $\mathbf{x}$ with the attention map $M_k(\mathbf{x})$ superimposed on it as the visualization of prototype $\mathbf{p}_k$.

Stage-3: Training of the Classification Layer After reassigning prototypes in stage-2, we train the classification layer separately to learn a weight matrix $W_{\text{cls}} \in \mathbb{R}^{K\times2}$ that represents the contributions of the prototypes for classification without changing their representation. The overall loss function is

$$\mathcal{L}_{\text{stage-3}} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{L1}}\mathcal{L}_{\text{L1}}, \tag{10}$$

where λ_{cls} and λ_{L1} are coefficients of respective losses.

Appendix C. Evaluation Metrics for Interpretability

C.1. Correctness: Incremental Deletion Score (IDS)

Incremental deletion (Nauta et al., 2022) is a commonly adopted method to assess the interpretability of an activation map by incrementally deleting features (e.g., replaced with 0) in the input according to the activation map values (from high values to low values). The top-valued features are deleted at the beginning. If the curve of classification metric drops faster, the activation map more precisely reflects a model's decision-making process. To quantify the sharpness of the curve of incremental deletion, incremental deletion score (IDS) is defined here as the normalized area under the curve and within the bounds of start and end, as shown in Figure 3. The lower the IDS, the better performance in terms of correctness.

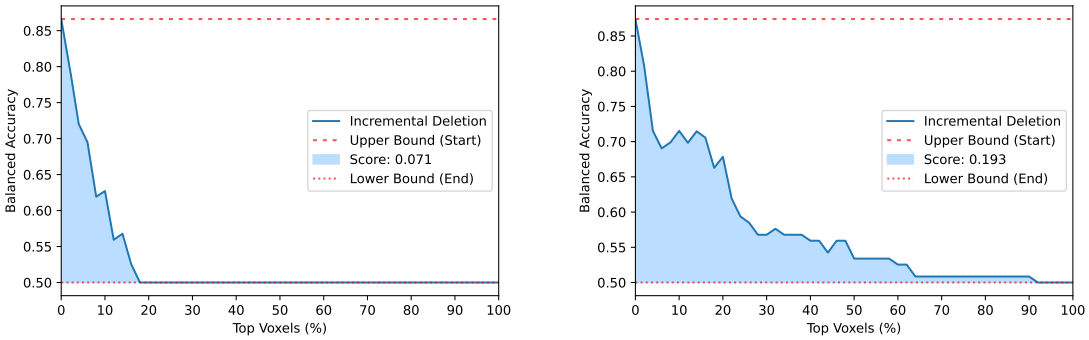

Figure 3: Demonstration of the incremental deletion score (IDS).

C.2. Localization Coherence: Activation Precision (AP)

The localization coherence is evaluated by the activation precision (AP) (Barnett et al., 2021)

$$\mathrm{AP} = \frac{|H(\mathbf{x}) \cap T(M(\mathbf{x}))|}{|T(M(\mathbf{x}))|}, \tag{11}$$

where $H(\mathbf{x})$ is the human-annotated label for the WT region of $\mathbf{x}$, $M(\mathbf{x})$ is the activation map for $\mathbf{x}$ and $T(\cdot)$ is a threshold function. In this work, $T(\cdot)$ is set to be a binary function with the threshold 0.5. For MProtoNet

$$M(\mathbf{x}) = \frac{1}{K^c} \sum M_k^c(\mathbf{x}), \tag{12}$$

where c denotes the correct class of $\mathbf{x}$ and K^c denotes the total number of prototypes assigned to class c.

Appendix D. Hyper-parameters During Training

Following some latest practices from Liu et al. (2022), MProtoNet is trained by the AdamW optimizer with a baseline learning rate of 0.001 and a weight decay coefficient of 0.01. During

each cross-validation iteration, MProtoNet is trained for 100 epochs (stage-1) with a batch size of 32 and a specific learning rate scheduler (linear warm-up for the first 20 epochs and cosine annealing for the remaining 80 epochs). After every 10 epochs of stage-1, there is a step of stage-2 (with only prototype reassignment) and 10 epochs of stage-3 (where MProtoNet is trained by the Adam optimizer with a constant learning rate of 0.001). The coefficients of loss functions are set as follows: λ_{cls} as 1, λ_{clst} as 0.8, λ_{sep} as 0.08, λ_{map} as 0.5, λ_{OC} as 0.05 and λ_{L1} as 0.01.

Appendix E. Visualization of Localization Coherence

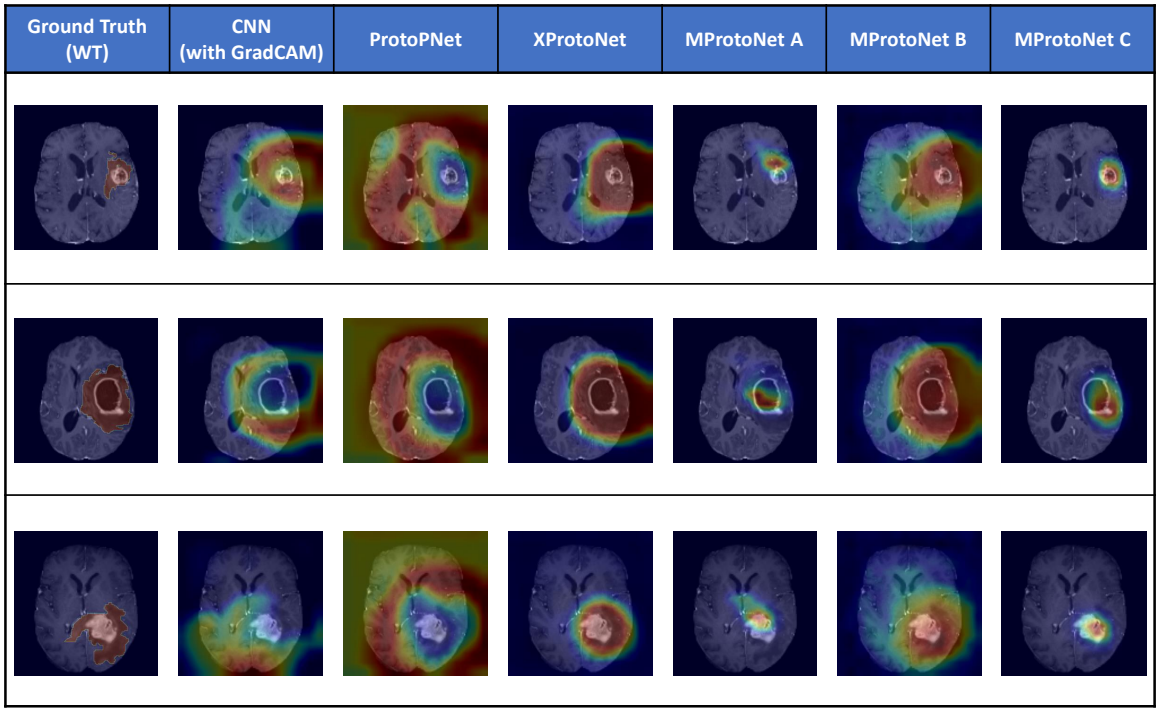

Figure 4: Visualization examples of the localization coherence results from different methods (shown with the T1CE modality).