ParseCaps: An Interpretable Parsing Capsule Network for Medical Image Diagnosis

Xinyu Geng[1], Jiaming Wang[1], Xiaolin Huang[2], Fanglin Chen[1], Jun Xu[1] *

[1]Harbin Institute of Technology, Shenzhen
[2]Shanghai Jiaotong University
{22s153095, 21s153144}@stu.hit.edu.cn, xiaolinhuang@sjtu.edu.cn, chenfanglin@gmail.com, xujunqgy@hit.edu.cn

Abstract

Deep learning has excelled in medical image classification, but its clinical application is limited by poor interpretability. Capsule networks, known for encoding hierarchical relationships and spatial features, show potential in addressing this issue. Nevertheless, traditional capsule networks often underperform due to their shallow structures, and deeper variants lack hierarchical architectures, thereby compromising interpretability. This paper introduces a novel capsule network, ParseCaps, which utilizes the sparse axial attention routing and parse convolutional capsule layer to form a parse-tree-like structure, enhancing both depth and interpretability. Firstly, sparse axial attention routing optimizes connections between child and parent capsules, as well as emphasizes the weight distribution across instantiation parameters of parent capsules. Secondly, the parse convolutional capsule layer generates capsule predictions aligning with the parse tree. Finally, based on the loss design that is effective whether concept ground truth exists or not, ParseCaps advances interpretability by associating each dimension of the global capsule with a comprehensible concept, thereby facilitating clinician trust and understanding of the model's classification results. Experimental results on three medical datasets show that ParseCaps not only outperforms other capsule network variants in classification accuracy and robustness, but also provides interpretable explanations, regardless of the availability of concept labels.

1 Introduction

Deep learning methods of medical image classification provide consistent, rapid predictions that often exceed human in detecting subtle abnormalities. However, obtaining extensive, high-quality datasets is challenging, and poor interpretability limits their clinical application. Capsule networks (CapsNets) have shown potential to enhance interpretability by maintaining hierarchical relationships and spatial orientations within images (Sabour, Frosst, and Hinton 2017; Hinton, Sabour, and Frosst 2018). They also improve classification accuracy by capturing relationships between disease markers and normal anatomical structures (Akinyelu et al. 2022; Ribeiro et al. 2022; Patrick et al. 2022). CapsNets utilize vectors called capsules to replace neurons. Each capsule vector's length represents the presence probability of specific entity in the input image, and its direction encodes the captured features (Sabour, Frosst, and Hinton 2017). Each dimension of the capsule vector, termed an instantiation parameter, represents the direction of the capsule, conferring inherent physical meanings. So, it holds potential for concept interpretability, as each corresponds to a human-understandable and meaningful concept.

Existing CapsNets face challenges to assign clear meaning to instantiation parameters; however, integrating a parse-tree-like structure could map part-to-whole relationships similarly to human cognitive processes (Sabour, Frosst, and Hinton 2017), thereby enhancing concept interpretability. In this structure, each node is a capsule, and active capsules select parent capsules from the upper layer through routing. Because (Sabour, Frosst, and Hinton 2017) does not adhere to a strict tree structure where each child node connects to only one parent node, it is referred to as parse-tree-like. The top layer features a single "global capsule" that provides a comprehensive view of the entire image, encapsulating entities and their hierarchical relationships. This allows each instantiation parameter to align with an interpretable and conceptually meaningful image entity. **The parse tree provides a carrier for interpretability through the global capsule. Coupled with loss constraints, they create a capsule network with concept interpretability.**

However, the current implementation of parse tree faces challenges. First, it lacks a suitable routing algorithm that supports the parse-tree-like structure. Dynamic routing process tends to create a fully connected structure between sub-capsules and parent capsules (Jeong, Lee, and Kim 2019), which undermines the desired selective connection essential for clear hierarchical relationships. Additionally, although existing attention routing sparsifies the coupling coefficients between capsules (Geng et al. 2024), it does not see the importance of instantiation parameters within capsules, which adversely affects the global capsule's instantiation parameters. Moreover, certain CapsNet layers contradict the parse tree by increasing the number of capsules without correspondingly enhancing their dimension (Rajasegaran et al. 2019; Choi et al. 2019). They fail to concentrate features of entire image into a global capsule; instead, as the number of capsules increases, features become more dispersed, leading to redundancy in feature representation and a diluted

*Corresponding author.

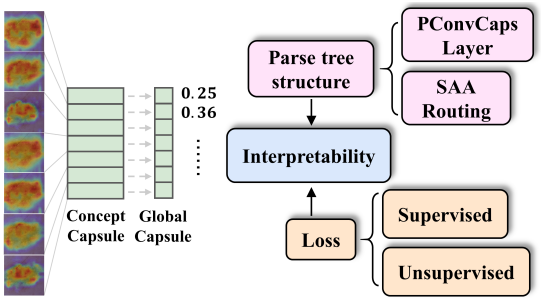

Figure 1: Using a parse-tree-like structure as a carrier, combined with loss functions, a capsule network with concept interpretability has been achieved. Each instantiation parameter of the global capsule corresponds to a human-understandable concept.

hierarchical structure. Furthermore, existing CapsNets lack loss functions to promote concept interpretability.

This paper presents ParseCaps, a novel capsule network featuring three enhancements corresponding to above challenges. First, following (Geng et al. 2024), a sparse axial attention (SAA) routing is proposed, which not only sparsifies coupling coefficients but also weighs instantiation parameters within each capsule. The sparsity inherent in SAA routing excludes sub-capsules with weaker connections to parent capsules, benefiting a hierarchical parse-tree-like structure. Second, we introduce a parse convolutional capsule (PConvCaps) layer. This layer generates capsules whose prediction strictly aligns with the parse tree, reducing the number of capsules while increasing their dimensionality as layer depth increases, thus forming an interpretable global capsule. Third, we design loss functions to align each dimension of the capsule vector with a human-understandable concept, regardless of whether concept ground truth are available or not. The motivation of this paper is shown in Fig. 1.

Contributions **1)** ParseCaps forms a parse-tree-like structure by creating PConvCaps layer and SAA routing, especially focusing on instantiation parameter weights in SAA routing. **2)** It enhances interpretability with a parse-tree-like structure and loss functions, aligning instantiation parameters of global capsule with human-understandable concepts. **3)** It outperforms existing CapsNets in medical image classification and provides interpretable explanations regardless of concept label availability.

2 Related Work

Capsule networks and parse-tree-like structure (Hinton, Ghahramani, and Teh 1999) first proposed the parse-tree-like structure for image processing , where lower-level nodes represent parts of an entity (i.e., eyes, nose, etc.), and top-level nodes depict the entire entity (i.e., the entire face). Most existing CapsNets with parse trees are constructed within shallow networks. (Sabour, Frosst, and Hinton 2017) introduced CapsNet with a basic two-layer parse-tree-like structure. (Peer, Stabinger, and Rodriguez-Sanchez 2018) proposed a dynamic parse tree for a four-layer CapsNet, and (Bui, Yu, and Jiang 2021) utilized shared weights routing for constructing syntax trees in code comprehension tasks, leading to a two-layer capsule network. (Yu et al. 2022) explored unsupervised facial parsing with a two-layer capsule encoder. However, deeper capsule networks depend on convolutional layers disrupt the parse-tree-like structure (Rajasegaran et al. 2019; Everett, Zhong, and Leontidis 2023). Although (LaLonde, Torigian, and Bagci 2020) investigate the interpretability of capsule networks in the medical field, it did not address the conceptual significance of instantiation parameters or the parse tree structure. In conclusion, the potential for interpretability and parsing structure in deep capsule networks remains underexplored.

Explainable medical image classification Despite the excellent performance of existing deep learning algorithms, their clinical deployment remains limited primarily due to the non-transparency of their decision-making processes (Patrício, Neves, and Teixeira 2023b), which is often described as "black-box" model (Lipton 2017). This encourages the development of interpretable deep learning algorithms that combine explainable models with high medical diagnostic accuracy (Li et al. 2024; Zhang et al. 2021). Early methods of interpretability involved perturbing input images to see how model outputs changed (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017), or analyzing model activations to determine essential lesions for predictions (Zhou et al. 2016; Selvaraju et al. 2017). However, these post-hoc explanation methods were criticized for potentially missing the complex dynamic process within models (Adebayo et al. 2018; Rudin 2019). Consequently, there is a growing trend towards designing models that inherently explain their decision-making processes (Kim et al. 2021; Wickramanayake, Hsu, and Lee 2021; Gallée, Beer, and Götz 2023). (Alvarez Melis and Jaakkola 2018; Sarkar et al. 2022) constructed models with built-in, pre-hoc interpretability, known as conceptual interpretability, which clarified classifier predictions through a set of human-understandable concepts.

3 Methodology

3.1 Overall architecture

The ParseCaps architecture is shown in Fig. 2. The input image x is fed into the initial convolutional block consisting of four convolutional layers that extract features $\Phi^0 \in \mathbb{R}^{(B,f_0,W_0,H_0)}$, where B, f_0, W_0, and H_0 denote the batch size, number of feature maps, width and height, respectively. This block maps the image features to a higher dimentional space, facilitating capsule creation (Mazzia, Salvetti, and Chiaberge 2021). Then, the primary capsule layer converts Φ^0 into capsules $U_1 \in \mathbb{R}^{(B,n_1,d_1)}$, where n_1 and d_1 are the number and dimension of primary capsules, respectively.

A capsule block consists of several capsule cells, each containing a PConvCaps layer, SAA routing, and MLP. The PConvCaps layer generates predictions $\hat{U}_{l+1}$ for higher-level capsules from lower-level U_l. SAA routing computes coupling coefficients c_{ij} based on the alignment between U_l and $\hat{U}_{l+1}$. The MLP deepens the network without changing

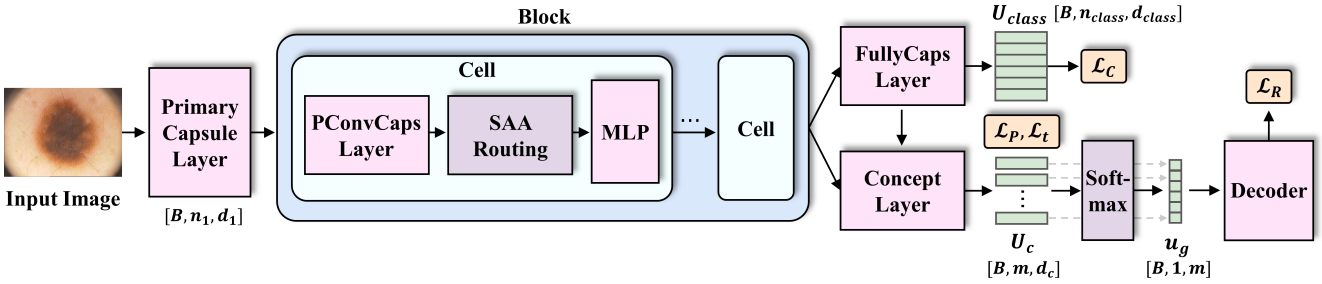

Figure 2: The architecture of ParseCaps.

capsule counts or dimensions. Capsule blocks can be stacked to extend the network depth effectively. ParseCaps includes three blocks with one, two, and five cells, respectively.

The final block connects to FullyCaps layer and concept layer, the FullyCaps layer converges all capsules into class capsules $U_{class} \in \mathbb{R}^{(B, n_{class}, d_{class})}$ to make predictions. In concept layer, inputs are mapped into m concept capsules $U_c \in \mathbb{R}^{(B,m,d_c)}$ by two linear layers. Each $u_{c,i}$ is converted into a 1D instantiation parameter of the global capsule $u_g \in \mathbb{R}^{(B,1,m)}$, representing m concepts $p_1, p_2, \ldots, p_m$. The softmax function renders each instantiation parameter to $[0, 1]$ as the activation of that concept. u_g feeds into a decoder with four deconvolution layers to reconstruct x.

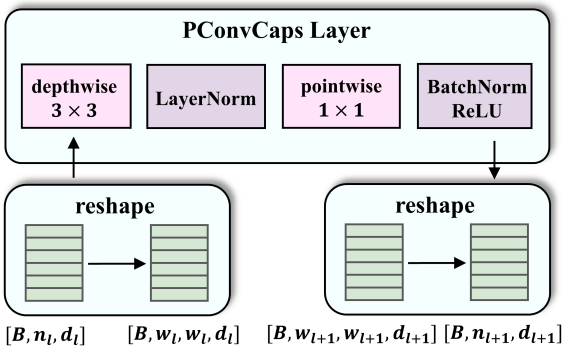

Figure 3: The process of PConvCaps Layer.

3.2 Parse-tree-like structure

(Peer, Stabinger, and Rodriguez-Sanchez 2018) identified two essential criteria for a parse tree structure : First, there should be fewer parent capsules than sub-capsules, with parent capsules having greater dimensions to encapsulate more features. Second, each sub-capsule should connect to only one parent capsule. PConvCaps layer meets the first criterion by buliding a model structure that widens at the base and narrows at the top. For the second criterion, it causes significant feature loss and overfitting to maintain one-to-one connections between parent capsules and sub-capsules. SAA routing is used to reduce unnecessary connections.

Parse convolutional capsule layer Recent studies on CapsNets commonly employ convolutional capsule (ConvCaps) layer (Rajasegaran et al. 2019; Geng et al. 2024; Choi et al. 2019). The input of the ConvCaps layer l is $U_l \in \mathbb{R}^{(B,w_l,w_l,d_l,n_l)}$, transforming into the output $\hat{U}_{l+1} \in \mathbb{R}^{(B,w_{l+1},w_{l+1},d_{l+1},n_{l+1})}$, where B, w_l, d_l and n_l are the batchsize, width of the input feature map, the dimension and the number of capsules, respectively. When performing convolution, U_l is reshaped into a 4D tensor $U_l \in \mathbb{R}^{(B,w_l,w_l,d_l \times n_l)}$, where $d_l \times n_l$ serves as the channel dimension of the tensor. As the convolutional layers deepen, $d_l \times n_l$ increases while w_l decreases. Typically, d_l is often set as a constant, thus the number of capsules n_l increases.

This phenomenon creates challenges in CapsNets. First, the parse-tree-like structure prefers that lower-layer capsules should be short (i.e. small vector dimension d) and numerous (i.e. large vector count n) to represent partial features, higher-layer capsules are fewer but longer, encompassing entire image entities. However, ConvCaps layers fail to adhere to this structure. Furthermore, increasing capsule counts without expanding feature space results in an overabundance of feature carriers. Capsules represent overlapping areas of feature distribution, leading to redundancy. Lastly, the 5D capsule (B, w_l, w_l, d_l, n_l) combines feature maps and capsules, which complicates the meaning of instantiation parameters and reduces interpretability.

We introduce the PConvCaps Layer detailed in Fig. 3. This layer ensures that all extracted features are represented by capsules, with the input capsule defined as $U_l \in \mathbb{R}^{(B,n_l,d_l)}$. To adapt to convolution, n_l is split into $\sqrt{n_l} \times \sqrt{n_l}$, and d_l corresponds to the number of channels in the convolutional layer. We define the width of the spatial grid be $w_l = \sqrt{n_l}$, and reshape U_l into (B, w_l, w_l, d_l).

Consequently, when the stride is larger than 1, the number of capsules n_l naturally decreases and the dimension d_l increases as the layers deepen, aligning with the parse tree. PConvCaps layer uses depthwise convolution (Chollet 2017) with a 3×3 kernel and stride of 2, followed by layer normalization (Ba, Kiros, and Hinton 2016) and pointwise convolution with a 1×1 kernel and stride of 1.

Sparse axial attention routing We develop the sparse axial attention routing (SAA routing) which consists of two parts: sparse attention and axial attention.

Sparse attention The essence of sparse attention routing is capturing the coupling coefficients C^s between lower-level capsules and higher-level capsules via an attention map, as shown in Fig. 4, where s denotes the sparse attention. For the l-th layer capsules $U_l \in \mathbb{R}^{(B,n_l,d_l)}$, the PConvCaps layer generates predictions of $(l+1)$-th layer capsules $\hat{U}^s_{l+1} \in \mathbb{R}^{(B,n_{l+1},d_{l+1})}$, consisting of n_{l+1} capsules,

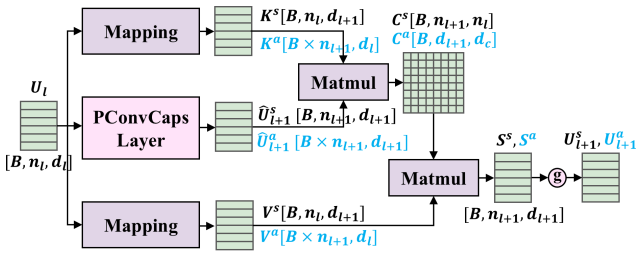

Figure 4: The process of SAA routing. The tensor shapes for sparse attention and axial attention are represented in black and blue, respectively. g is the activation function Squash.

each with d_{l+1} dimensions. $\hat{U}^s_{l+1}$, $K^s \in \mathbb{R}^{(B,n_l,d_{l+1})}$ and $V^s \in \mathbb{R}^{(B,n_l,d_{l+1})}$ serve as the query, key and value in the attention mechanism, respectively. For matrix multiplication, K^s and V^s are mapped to (B, n_l, d_{l+1}) through pointwise convolution with a kernel size of 1×1 and stride 1. The computation of attention scores corresponds to calculate coupling coefficients c_{ij} in dynamic routing. The coupling coefficient matrix $C^s \in \mathbb{R}^{(B,n_{l+1},n_l)}$ is calculated in Eq. 1:

$$C^s = \alpha\text{-Entmax}(\frac{\hat{U}^s_{l+1}(K^s)^{\mathrm{T}}}{\sqrt{d_{l+1}}}) \tag{1}$$

α-Entmax is the sparse softmax (Correia, Niculae, and Martins 2019) and is defined as Eq. 2:

$$\alpha\text{-Entmax}(x)_i = \max\left(\frac{x_i - \tau}{\alpha}, 0\right)^{\frac{1}{\alpha-1}} \tag{2}$$

τ is a self-adaption threshold and α controls the sparsity of the attention map. α-Entmax adaptively sets smaller coupling coefficient c_{ij} to zero, encouraging sub-capsules to connect to most relevant parent capsules. This selective connectivity is crucial for the parse-tree-like structure. After assigning c_{ij} to $\hat{U}^s_{l+1}$, the votes are generated through $S^s = C^s(V^s)^{\mathrm{T}}$, then passing through a nonlinear activation function g to obtain the output $U^s_{l+1} = g(S^s)$. The sparsity in routing preserves the richness of features and aligns with a parse-tree-like structure.

Axial attention Sparse attention routing assigns coupling coefficient c_{ij} to each lower-level capsule $u_{l,i}$, uniformly scaling all dimensions (i.e. instantiation parameter) within $u_{l,i}$ by the same constant c_{ij}. However, as not all instantiation parameter contribute equally to routing, the weight distribution for instantiation parameters within each capsule is needed. Nevertheless, traditional attention mechanisms based on dot products between vectors fail to account for dependencies between individual elements within capsules.

Axial attention decomposes a 3D input into capsule-wise (n-axis) and dimension-wise (d-axis) components, then applying attention independently to each axis (Ho et al. 2019; Wang et al. 2020; Valanarasu et al. 2021). We focus attention along d-axis, targeting the instantiation parameters in each capsule. Following (Ho et al. 2019), unrelated axis n is rearranged into the batch dimension B. Other process is similar to sparse attention, detailed in Alg. 1. The shape of each tensor of axial attention is shown in Fig. 4 with blue. The outputs from both attentions, U^s_{l+1} and U^a_{l+1}, are combined to get parent capsules U_{l+1}, with both attentions computed in parallel to optimize training time.

Complexity analysis For $U \in \mathbb{R}^{(B,n,d)}$, sparse attention has a computational complexity of $O(n^2)$, while axial attention along d-axis incurs $O(d)$. Given $O(d) < O(n^2)$, the overall complexity remains $O(n^2)$. Compared with traditional attention routing where $\hat{U} \in \mathbb{R}^{(B,n,W\times H\times d)}$ with complexity of $O(W^2H^2d^2)$ (Pucci, Micheloni, and Martinel 2021), $\hat{U} \in \mathbb{R}^{(B,n_{l+1},n_l,d)}$ with $O(n_l^2d^2)$ (Mazzia, Salvetti, and Chiaberge 2021), and $\hat{U} \in \mathbb{R}^{(B,w,h,n,d)}$ with Conv3D complexity of $O(WHND^2)$ (Choi et al. 2019), SAA routing reduces computational costs.

Algorithm 1: SAA Routing

Require: $U_l \in \mathbb{R}^{(B,n_l,d_l)}$
Ensure: $U_{l+1} \in \mathbb{R}^{(B,n_{l+1},d_{l+1})}$
Compute sparse attention:
$\hat{U}^s_{l+1} \leftarrow \text{PConvCaps}(U_l)$
$K^s, V^s \leftarrow \text{PointwiseConv}(U_l)$
$C^s \leftarrow \alpha\text{-entmax}(\frac{\hat{U}^s_{l+1}(K^s)^{\mathrm{T}}}{\sqrt{d_{l+1}}})$
$S^s \leftarrow C^s(V^s)^{\mathrm{T}}$
$U^s_{l+1} \leftarrow g(S^s)$
Axial attention along d-axis:
$\hat{U}^a_{l+1} \leftarrow Reshape(B \times n_{l+1}, d_{l+1})$
$K^a, V^a \leftarrow \text{PointwiseConv}(U_l)$
$C^a \leftarrow \alpha\text{-Entmax}(\frac{\hat{U}^a_{l+1}(K^a)^{\mathrm{T}}}{\sqrt{n_{l+1}}})$
$S^a \leftarrow C^a(V^a)^{\mathrm{T}}$
$U^a_{l+1} \leftarrow g(S^a)$
return $U_{l+1} \leftarrow U^s_{l+1} + U^a_{l+1}$

3.3 Loss function

The loss function ensures that each concept capsule represents a specific concept, linking visual features in the image to the textual characteristics of the concept. It accommodates scenarios with and without concept ground truth labels.

When concept labels are available

Presentation loss To ensure the i-th concept capsule $u_{c,i}$ uniquely signifies the concept p_i, the $u_{c,i}$ should be activate when p_i is present in the image. Let $Z = (z_1, z_2, \ldots, z_m)$ represent the indicator for concept labels, where $z_i = 1$ indicates the presence of concept p_i, and $z_i = 0$ otherwise. Accordingly, the presentation loss L_p is defined in Eq. 3:

$$\begin{aligned} L_p = \sum_{i=1}^{m} & z_i \max(0, t_p^+ - ||u_{c,i}||)^2 \\ & + (1 - z_i)\max(0, ||u_{c,i}|| - t_p^-)^2 \end{aligned} \tag{3}$$

Inspired by (Sabour, Frosst, and Hinton 2017), activation margins $t_p^+ = 0.9$ and $t_p^- = 0.1$ are set to ensure the length of the capsule vector $||u_{c,i}||$ exceeds t_p^+ for the correct concept, and remains below t_p^- for the incorrect concept.

Models	ACC↑	Avg. Precision↑	Avg. Recall↑	Avg. F1 Score↑	Avg. Specificity↑
ParseCaps	**99.38**	**98.77**	**98.69**	**98.57**	**99.33**
BrainCaps(Vimal Kurup, Sowmya, and Soman 2020)	92.60	92.67	94.67	93.33	-
OrthCaps (Geng et al. 2024)	92.57	86.18	86.27	85.84	94.25
DeepCaps (Rajasegaran et al. 2019)	93.69	93.95	93.60	93.58	97.49
CapsNet (Sabour, Frosst, and Hinton 2017)	92.16	93.36	91.47	92.08	93.84
Modified GoogleNet (Sekhar et al. 2021)	94.90	94.76	93.69	94.30	97.22
Block-wise VGG19 (Swati et al. 2019)	94.82	89.52	94.25	91.73	94.71

Table 1: Comparison of ParseCaps with other models on the CE-MRI dataset. Avg. is a macro average.

Triplet loss To link specific image regions with corresponding conceptual phrases, we use a linear embedding layer to map the instantiation capsule $u_{c,i}$ and concept p_i into a joint latent space, creating $\tilde{u}_{c,i}$ and $\tilde{p}_i$. We want the embedding of the i-th instantiation capsule $\tilde{u}_{c,i}$ corresponds to the i-th concept p_i, while the embedding of the j-th capsule $\tilde{u}_{c,j}$ does not, distinguishing the different embeddings. The triplet loss L_p ensures that the distance between $\tilde{u}_{c,i}$ and $\tilde{p}_i$ is less than the distance between $\tilde{u}_{c,j}$ and $\tilde{p}_i$ by a margin t_t. The triplet loss L_t is defined in Eq. 4:

$$\begin{aligned} L_p = \sum_{i=1}^{m} z_i \max(0, t_p^+ - ||u_{c,i}||)^2 \\ + (1 - z_i) \max(0, ||u_{c,i}|| - t_p^-)^2 \end{aligned} \tag{4}$$

Overall loss Overall loss function L is defined in Eq. 5:

$$L = L_c + \lambda L_p + \eta L_t \tag{5}$$

L_c is the classification loss, which is the cross-entropy loss. λ and η are the weight hyperparameters.

When concept labels are unavailable When explicit concept labels are unavailable, L_p and L_t cannot be used. Nevertheless, the parse-tree-like structure, where each node represents a specific image region or component, aids the model in understanding the relationships and hierarchy among different image parts. Additionally, the instantiation parameter can inherently signify the presence of specific concepts. Therefore, it is adequate to motivate the model to learn concepts that reflect the semantics of the input image x_i. To measure reconstruction error, the reconstruction loss L_r is added to the overall loss L. If the model's conceptual understanding is insufficient for an accurate reconstruction of x_i, the L_r penalizes the model. We utilize an L_2 loss for this purpose and assign a weight of γ. The overall loss function L is defined as Eq. 6:

$$L = L_c + \gamma L_r \tag{6}$$

4 Experiments

4.1 Experimental setup[1]

Setup and datasets ParseCaps was developed in PyTorch 12.1 and Python 3.9, accelerated by eight GTX-3090 GPUs. We set the learning rate to 2.5e-3, batch size to 64, and weight decay to 5e-4. The model was trained for 300 epochs using the AdamW optimizer and a 5-cycle linear warm-up. We evaluated ParseCaps with Contrast Enhanced Magnetic Resonance Images (CE-MRI) (Cheng 2017), PH^2 (Mendonça et al. 2013) and Derm7pt (D7) (Kawahara et al. 2019) datasets. CE-MRI has 3064 MRI T1w post GBCA images from 233 patients. PH^2 and D7 are two skin diagnostic datasets with 200 and 2000 images, respectively. we split all datasets into 80% training, 10% testing, and 10% validation.

Concept label acquisition By dividing textual descriptions using a concept parser, we get conceptual phrases composed of a noun and corresponding adjectives, serving as concept-level annotations (Wickramanayake, Hsu, and Lee 2021). Based on this rule and the annotations in the dermoscopic standards (Patrício, Neves, and Teixeira 2023a), we select "Atypical Pigment Network" (APN), "Typical Pigment Network" (TPN), "Blue Whitish-Veil" (BWV), "Irregular Streaks" (ISTR), "Regular Streaks" (RSTR), "Regular Dots and Globules" (RDG), and "Irregular Dots and Globules" (IDG) as concept labels.

4.2 Classification performance comparison

We choose CE-MRI dataset to evaluate the classification ability of ParseCaps in Tab. 1, because it is typical for comparable CapsNets variants. ParseCaps outperforms other models in accuracy (ACC), with a notable score of 99.38%. ParseCaps also leads in average precision, recall, F1 score, and specificity, underscoring its robustness and effectiveness in handling CE-MRI dataset. Tab. 2 presents class-specific metrics for Meningioma, Glioma, and Pituitary tumor, where ParseCaps shows high performance across all metrics, achieving a specificity of 99.79% for Meningioma and a precision of 99.74% for Glioma. Tab. 3 shows ParseCaps' performance on skin datasets, including PH^2, D7, and a combined PH^2D7. Compared with Coherent CNN(Patrício, Neves, and Teixeira 2023a), Skin VGGNet(Lopez et al. 2017), CCNN(Wickramanayake, Hsu, and Lee 2021), and SqueezeNet(Abayomi-Alli et al. 2021), ParseCaps has the highest scores in two datasets, achieving 97.53% and 87.96% respectively. Although the performance on D7 is lower than the coherent CNN model, it is still competitive considering the interpretability. Overall, these results affirm ParseCaps' efficacy across various medical image tasks, with additional classification results in the supplementary material.

[1]The code is released at https://github.com/ornamentt/Parsecaps. The supplementary material is available at https://arxiv.org/pdf/2411.01564.

Classes	Meningioma				Glioma				Pituitary tumor			
Metrics	P↑	R↑	S↑	F1 ↑	P	R	S	F1	P	R	S	F1
ParseCaps	**96.92**	**95.64**	**99.79**	**96.12**	**99.74**	**99.39**	**99.63**	**99.54**	**98.08**	**99.06**	**99.39**	**98.89**
OrthCaps (Geng et al. 2024)	89.14	83.15	97.84	85.37	96.35	97.66	91.77	97.03	91.13	96.19	96.83	96.40
DeepCaps (Rajasegaran et al. 2019)	94.71	91.79	98.47	91.04	98.33	97.12	95.41	96.71	90.79	96.10	95.22	95.14
CapsNet (Sabour, Frosst, and Hinton 2017)	88.57	94.33	86.33	91.06	86.78	88.09	87.70	92.36	91.10	92.54	86.05	95.14
BrainCaps (Vimal Kurup, Sowmya, and Soman 2020)	85.00	94.00	-	89.00	98.00	96.00	-	97.00	95.00	94.00	-	94.00
Modified GoogleNet (Sekhar et al. 2021)	93.78	86.98	98.19	90.25	96.02	97.00	96.00	96.51	94.48	97.10	97.47	95.77
Block-wise VGG19 (Swati et al. 2019)	87.97	89.98	96.42	88.88	93.26	95.97	93.79	94.52	87.34	96.81	93.93	91.80

Table 2: Performance of each class on CE-MRI. P, R, S, and F1 are precision, recall, specificity, and F1 score, respectively.

Models	PH^2↑	D7↑	PH^2D7↑
ParseCaps	**97.53**	83.42	**87.96**
Coherent CNN	96.00	84.06	84.44
Skin VGGNet	90.67	76.15	79.23
CCNN	93.33	**84.27**	83.96
SqueezeNet	92.18	-	-

Table 3: Classification accuracy performance on skin datasets. PH^2D7 is a combination of the PH^2 and D7 datasets created by (Patrício, Neves, and Teixeira 2023a).

Models	ParseCaps		DeepCaps	OrthCaps
Metrics	$n\|d$	$n\|d$	$n\|d$	$n\|d$
PCL	784 | 8	14 | 2	1 | 128	16 | 16
CCB1	196 | 16	7 | 4	4 | 32	32 | 16
CCB2	49 | 36	4 | 8	8 | 32	64 | 16
CCB3	13 | 64	2 | 16	8 | 32	128 | 16
FCL	1 | 64	1 | 16	10 | 16	10 | 16

Table 4: Comparison of the number of capsules n and the dimensions d. PCL, CCB and FCL represent Primary Capsule Layer, ConvCaps Block and FullyCaps Layer respectively.

4.3 Ablation study

PConvCaps layer ParseCaps, DeepCaps, and OrthCaps all have a ConvCaps Block. Tab. 4 compares the capsule counts (n) and dimensions (d) within these models to highlight PConvCaps layer's role in the parse-tree-like structure. Unlike DeepCaps and OrthCaps, which generally increase the number of capsules while maintaining or reducing their dimensions as layers deepen, ParseCaps consistently reduces the number of capsules and increases their dimensions, aligning with the parse-tree-like structure. Furthermore, ParseCaps adaptively adjusts n and d according to the dataset's complexity. For example, as shown in the left column of ParseCaps, larger and more complex images of CE-MRI lead to correspondingly larger n and d, increasing the amount of information the capsules can represent. For simpler datasets like MNIST (Lecun et al. 1998) in the right column, n and d are minimized to simplify the model and prevent overfitting, with automatic adjustments based on input image size that require no extra operation.

Loss functions We assess the impact of each loss using Explanation Error (EE) (Sarkar et al. 2022) and classification accuracy (ACC), measured EE as the L_2 distance, where a lower EE indicates better alignment with ground truth concepts. Tab. 5 shows that the combination of the classification loss L_c, the presentation loss L_p and the triplet loss L_t achieves the best performance, with ACC of 97.53% and EE of 0.98. Removing L_t leads to a drop in performance, with ACC falling to 97.25% and EE increasing to 1.13, underscoring L_t's critical role in linking concept features with visual features. The ablation study of the reconstruct loss L_r is in the supplementary material.

Method	ACC ↑	EE ↓
L_c	94.89	4.02
$L_c + L_p$	95.01	3.93
$L_c + L_t$	97.25	1.13
$L_c + L_t + L_p$	**97.53**	**0.98**

Table 5: Ablation study on loss functions. Results are tested with ParseCaps on PH^2 dataset.

Routing	FPS ↑	FLOPS ↓	ACC ↑
SAA	**387.88**	**279M**	**98.66**
Attention	277.55	10627M	98.23
Dynamic	252.78	40609M	98.35

Table 6: Ablation study on different routing algorithms. Results are tested on the MNIST dataset.

Attention routing We test the performance of routing algorithms in a basic CapsNet model, featuring a convolutional layer with a kernel size of 3 and stride of 2, a primary capsule layer, and a digit capsule layer, with the routing between the last two layers. As shown in Tab. 6, SAA routing outperforms others with an accuracy of 99.01% and performs best in FPS and FLOPS, showcasing superior efficiency. Although dynamic routing records a decent accuracy of 98.35%, it suffers from high computational costs (40609M FLOPS) and the lowest FPS (252.78).

4.4 Effectiveness of parse-tree-like structure

Robustness analysis In the parse-tree-like structure, lower-level capsules detect basic features like edges and corners, and higher-level capsules aggregate them to represent complex entities. This bottom-up feature integration enables the model to maintain consistent recognition under affine transformations, thus enhancing robustness. Following the experimental protocol of (Sabour, Frosst, and Hinton 2017), we train ParseCaps for 100 epochs on the MNIST and test it on the affNIST (Tieleman 2013), which subjects images to random affine transformations like rotations, scaling, and translations. We use a baseline capsule network with

Variants	MNIST ↑	affNIST ↑
ParseCaps	**99.30**	**84.32**
Baseline	99.23	79.00
CNN	99.22	66.00

Table 7: Robustness analysis on the affNIST dataset. MNIST and affNIST are accuracies on these datasets, respectively.

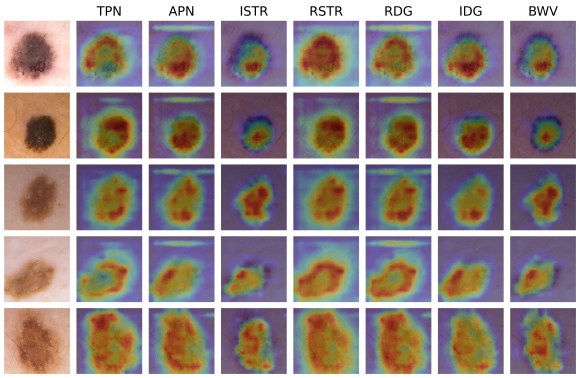

Figure 5: Visualization of the concept capsules with ground truth in ParseCaps.

SAA routing but without the parse-tree-like structure. Tab. 7 shows that ParseCaps demonstrates superior accuracy on affNIST, achieving 84.32%, compared to baseline's 79.00% and CNN's 66.00%, confirming ParseCaps' robustness.

Concept interpretability evaluation To validate the interpretability of ParseCaps, we analyze the concept capsules within the model through visualization, as shown in Fig. 5. The highlighted areas coincide with the lesion areas, proving the detection of key regions for melanoma. According to the ABCD rule (Nachbar et al. 1994), these areas clinically correspond to common signs of melanoma: Asymmetry, irregular Borders, multiple Colors, and Dermoscopic structures (Patrício, Neves, and Teixeira 2023a). We construct a baseline model with the same structure as ParseCaps except the parse-tree-like structure, directly connecting all capsules to the concept layer. As shown in Fig. 6, its concept capsules fail to capture critical areas and instead focus on edge features, which should be captured by lower-level capsules, validating the effectiveness of parse-tree-like structure.

When concept ground truth is unavailable, ParseCaps leverages internal entity relationships to provide interpretable concepts within images. We choose prototypes that maximize the activation values of each concept capsule to define its specific meanings. Due to the difficulty non-medical experts face in annotating the medical concepts of the selected prototypes, we have chosen to test ParseCaps on ImageNet-mini (Vinyals et al. 2016), which has more images and concepts that are easier to recognize. As depicted in Fig. 7, p_1 captures entities with vertical line features, p_2 identifies an animal's face, p_3 focuses on green entities, p_4 represents a left-facing bird with a pointed beak, p_5 encompasses black or dark entities, p_6 captures cubic shapes, p_7 is associated with the blue sky or sea, and p_8 recognizes circular features. Although there are occasional misidentifications, such as slippers in p_4, ParseCaps consistently demonstrates interpretable ability without concept supervision.

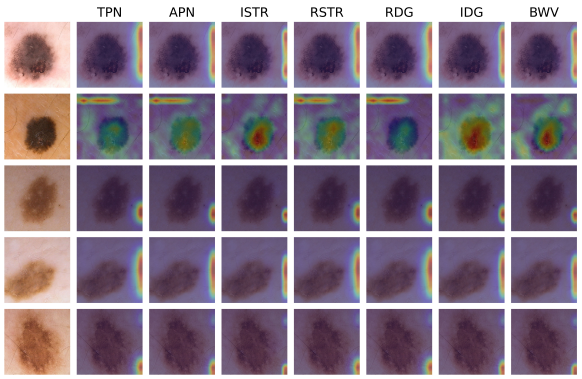

Figure 6: Visualization of concept capsules in baseline.

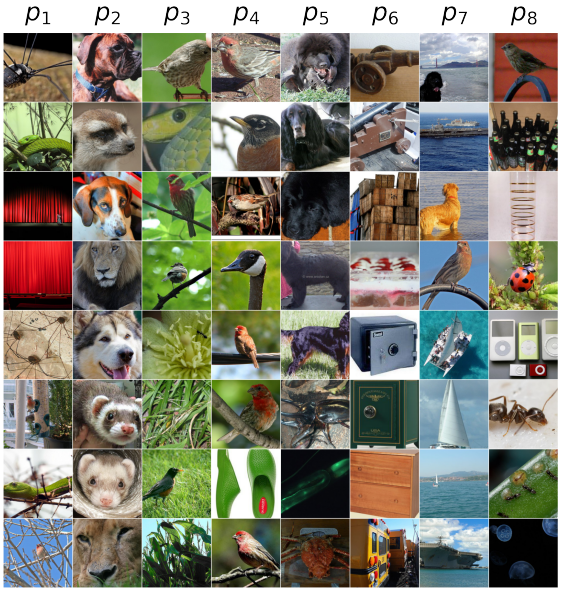

Figure 7: Visualization of the concept capsules without ground truth on ImageNet-mini.

5 Conclusion and Limitations

In this paper, we introduce ParseCaps, which incorporates a parse-tree-like structure and loss functions to build an interpretable capsule network in medical classification. However, ParseCaps has limitations. First, the parse-tree-like structure is not a strict parse tree with single-parent connections. which could enhance interpretability by more clearly explaining decision-making processes. Second, due to limited medical knowledge, identifying conceptual meanings for prototypes is challenging. ParseCaps lacks experiments in interpretability visualization without concept ground truth on medical datasets. We hope for continued exploration of unsupervised interpretability in medical datasets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62173113, 62376155 and 62476072, the Science and Technology Innovation Committee of Shenzhen Municipality under Grant GXWD20231129101652001, Natural Science Foundation of Guangdong Province under Grant 2022A1515011584, Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011706, Shenzhen Key Technical Project under Grant KJZD20230923115117033, Shanghai Municipal Science and Technology Research Program under Grant 22511105600 and Major Project under Grant 2021SHZDZX0102.

References

Abayomi-Alli, O. O.; Damasevicius, R.; Misra, S.; Maskeliunas, R.; and Abayomi-Alli, A. 2021. Malignant skin melanoma detection using image augmentation by oversamplingin nonlinear lower-dimensional embedding manifold. *Turkish Journal of Electrical Engineering and Computer Sciences*, 29(8): 2600–2614.

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.

Akinyelu, A. A.; Zaccagna, F.; Grist, J. T.; Castelli, M.; and Rundo, L. 2022. Brain Tumor Diagnosis Using Machine Learning, Convolutional Neural Networks, Capsule Neural Networks and Vision Transformers, Applied to MRI: A Survey. *Journal of Imaging*, 8(8): 205.

Alvarez Melis, D.; and Jaakkola, T. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bui, N. D. Q.; Yu, Y.; and Jiang, L. 2021. TreeCaps: Tree-Based Capsule Networks for Source Code Processing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1): 30–38.

Cheng, J. 2017. Brain tumor dataset. *figshare. Dataset*, 1512427(5).

Choi, J.; Seo, H.; Im, S.; and Kang, M. 2019. Attention Routing Between Capsules. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 0–0.

Chollet, F. 2017. Xception: Deep Learning With Depthwise Separable Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1251–1258.

Correia, G. M.; Niculae, V.; and Martins, A. F. 2019. Adaptively sparse transformers. *arXiv preprint arXiv:1909.00015*.

Everett, M.; Zhong, M.; and Leontidis, G. 2023. Vanishing Activations: A Symptom of Deep Capsule Networks. *arXiv preprint arXiv:2305.11178*.

Gallée, L.; Beer, M.; and Götz, M. 2023. Interpretable Medical Image Classification Using Prototype Learning and Privileged Information. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 435–445. Springer.

Geng, X.; Wang, J.; Gong, J.; Xue, Y.; Xu, J.; Chen, F.; and Huang, X. 2024. OrthCaps: An Orthogonal CapsNet with Sparse Attention Routing and Pruning. *arXiv preprint arXiv:2403.13351*.

Hinton, G. E.; Ghahramani, Z.; and Teh, Y. W. 1999. Learning to Parse Images. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 12. MIT Press.

Hinton, G. E.; Sabour, S.; and Frosst, N. 2018. Matrix capsules with EM routing. In *International Conference on Learning Representations (ICLR)*.

Ho, J.; Kalchbrenner, N.; Weissenborn, D.; and Salimans, T. 2019. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*.

Jeong, T.; Lee, Y.; and Kim, H. 2019. Ladder Capsule Network. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 3071–3079. PMLR.

Kawahara, J.; Daneshvar, S.; Argenziano, G.; and Hamarneh, G. 2019. Seven-Point Checklist and Skin Lesion Classification Using Multitask Multimodal Neural Nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2): 538–546.

Kim, E.; Kim, S.; Seo, M.; and Yoon, S. 2021. XProtoNet: Diagnosis in Chest Radiography With Global and Local Explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15719–15728.

LaLonde, R.; Torigian, D.; and Bagci, U. 2020. Encoding visual attributes in capsules for explainable medical diagnoses. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, 294–304. Springer.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, C.; Liu, X.; Li, W.; Wang, C.; Liu, H.; Liu, Y.; Chen, Z.; and Yuan, Y. 2024. U-kan makes strong backbone for medical image segmentation and generation. *arXiv preprint arXiv:2406.02918*.

Lipton, Z. C. 2017. The doctor just won't accept that! *arXiv preprint arXiv:1711.08037*.

Lopez, A. R.; Giro-i Nieto, X.; Burdick, J.; and Marques, O. 2017. Skin lesion classification from dermoscopic images using deep learning techniques. In *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, 49–54. IEEE.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Mazzia, V.; Salvetti, F.; and Chiaberge, M. 2021. Efficient-CapsNet: Capsule network with self-attention routing. *Scientific Reports*, 11(1): 14634.

Mendonça, T.; Ferreira, P. M.; Marques, J. S.; Marcal, A. R. S.; and Rozeira, J. 2013. PH2 - A dermoscopic image database for research and benchmarking. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 5437–5440.

Nachbar, F.; Stolz, W.; Merkle, T.; Cognetta, A. B.; Vogt, T.; Landthaler, M.; Bilek, P.; Braun-Falco, O.; and Plewig, G. 1994. The ABCD rule of dermatoscopy: High prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, 30(4): 551–559.

Patrício, C.; Neves, J. a. C.; and Teixeira, L. F. 2023a. Coherent Concept-Based Explanations in Medical Image and Its Application to Skin Lesion Diagnosis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 3799–3808.

Patrício, C.; Neves, J. C.; and Teixeira, L. F. 2023b. Explainable deep learning methods in medical image classification: A survey. *ACM Computing Surveys*, 56(4): 1–41.

Patrick, M. K.; Adekoya, A. F.; Mighty, A. A.; and Edward, B. Y. 2022. Capsule Networks – A survey. *Journal of King Saud University - Computer and Information Sciences*, 34(1): 1295–1310.

Peer, D.; Stabinger, S.; and Rodriguez-Sanchez, A. 2018. Training deep capsule networks. *arXiv preprint arXiv:1812.09707*, 1–7.

Pucci, R.; Micheloni, C.; and Martinel, N. 2021. Self-Attention Agreement Among Capsules. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 272–280.

Rajasegaran, J.; Jayasundara, V.; Jayasekara, S.; Jayasekara, H.; Seneviratne, S.; and Rodrigo, R. 2019. DeepCaps: Going Deeper With Capsule Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10725–10733.

Ribeiro, F. D. S.; Duarte, K.; Everett, M.; Leontidis, G.; and Shah, M. 2022. Learning with capsules: A survey. *arXiv preprint arXiv:2206.02664*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135—-1144.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 3856–3866.

Sarkar, A.; Vijaykeerthy, D.; Sarkar, A.; and Balasubramanian, V. N. 2022. A Framework for Learning Ante-Hoc Explainable Models via Concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10286–10295.

Sekhar, A.; Biswas, S.; Hazra, R.; Sunaniya, A. K.; Mukherjee, A.; and Yang, L. 2021. Brain tumor classification using fine-tuned GoogLeNet features and machine learning algorithms: IoMT enabled CAD system. *IEEE Journal of Biomedical and Health Informatics*, 26(3): 983–991.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 618–626.

Swati, Z. N. K.; Zhao, Q.; Kabir, M.; Ali, F.; Ali, Z.; Ahmed, S.; and Lu, J. 2019. Brain tumor classification for MR images using transfer learning and fine-tuning. *Computerized Medical Imaging and Graphics*, 75: 34–46.

Tieleman, T. 2013. The affnist dataset. *cs. toronto. edu*.

Valanarasu, J. M. J.; Oza, P.; Hacihaliloglu, I.; and Patel, V. M. 2021. Medical Transformer: Gated Axial-Attention for Medical Image Segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*, 36–46. Springer.

Vimal Kurup, R.; Sowmya, V.; and Soman, K. 2020. Effect of Data Pre-processing on Brain Tumor Classification Using Capsulenet. In *ICICCT 2019–System Reliability, Quality Control, Safety, Maintenance and Management: Applications to Electrical, Electronics and Computer Science and Engineering*, 110–119. Springer.

Vinyals, O.; Blundell, C.; Lillicrap, T.; kavukcuoglu, k.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29. Curran Associates, Inc.

Wang, H.; Zhu, Y.; Green, B.; Adam, H.; Yuille, A.; and Chen, L.-C. 2020. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *European Conference on Computer Vision (ECCV)*, 108–126. Springer.

Wickramanayake, S.; Hsu, W.; and Lee, M. L. 2021. Comprehensible convolutional neural networks via guided concept learning. In *2021 International Joint Conference on Neural Networks*, 1–8. IEEE.

Yu, C.; Zhu, X.; Zhang, X.; Wang, Z.; Zhang, Z.; and Lei, Z. 2022. HP-Capsule: Unsupervised Face Part Discovery by Hierarchical Parsing Capsule Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4032–4041.

Zhang, Y.; Li, C.; Lin, X.; Sun, L.; Zhuang, Y.; Huang, Y.; Ding, X.; Liu, X.; and Yu, Y. 2021. Generator versus segmentor: Pseudo-healthy synthesis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VI 24*, 150–160. Springer.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2921–2929.