Brain Tumor Detection

Damilola Olaiya[*]
Carleton University
Ottawa, Canada
damilolaolaiya@cmail.carleton.ca

Hamza Cecen[*]
Carleton University
Ottawa, Canada
hamzacecen@cmail.carleton.ca

WooSeok Kim[*]
Carleton University
Ottawa, Canada
wooseokkim@cmail.carleton.ca

[*]These authors contributed equally to this work.

Abstract

Magnetic Resonance Imaging (MRI) has a central role in the detection and clinical assessment of brain tumors. Recent advancements in machine learning and object detection, especially YOLO-related frameworks, have significantly improved automated tumor localization in 2D multiplanar MRI slices. One of these, Pretrained Knowledge Guided YOLO (PK-YOLO) has demonstrated state-of-the-art performance by integrating SparK-pretrained RepViT backbone and specialized optimization techniques. However, the original PK-YOLO design exhibits notable limitations, primarily stemming from a plane-shift domain gap, with its backbone pretrained exclusively on axial images, resulting in substantial performance disparities across planes.

To solve this limitation, we propose a modified model architecture, Mixture PK-YOLO. In this method, three different backbones, each of which is pretrained on an anatomical plane (*i.e.*, axial, coronal, or sagittal), are used in a frozen manner. Then the outputs of the backbones are combined using an adaptive router module that learns how much weight to give each plane. Although experimental results show that the proposed Mixture PK-YOLO doesn't surpass that of the original PK-YOLO, it significantly reduces the number of trainable parameters while effectively showing competent model performance. The code is available at https://github.com/3seoksw/Brain-Tumor-PK-YOLO.

1 Introduction

Over the past few years, advances in machine learning have opened up the possibilities to integrate deep learning methods into the medical domain[1]. Throughout active research, several papers employ multiple approaches (*e.g.*, image processing, convolutional networks, attention-based networks) to improve brain tumor detection performance [19].

Spotting brain tumors matters a lot when looking at medical scans, since finding them early and correctly helps doctors figure out what's wrong, decide on treatments, and keep track of how patients are doing [8]. MRIs are common for this job because they show fine details and distinguish soft tissues well [17]. Especially useful are multi-angle MRI views, like top, front, and side cuts, which give different body viewpoints, making it easier for physicians to pin down where the tumor sits and understand its shape.

Table 1: PK-YOLO's performance comparison table across different models and MRI planes.

Dataset	Precision	Recall	mAP$_{50}$	mAP$_{50-95}$
axial	0.858	0.896	0.947	0.681
coronal	0.834	0.793	0.805	0.689
sagittal	0.476	0.845	0.582	0.382

Alongside recent computer vision technologies, YOLO-based models have drawn significant attention due to their efficiency, robust detection performance, and real-time capabilities. However, applying YOLO architectures in medical imaging remains challenging because MRI scans differ substantially from natural images in terms of intensity patterns, noise characteristics, and underlying anatomical structures.

To address these challenges, M. Kang *et al.* [14] proposed a new You Only Look Once (YOLO)-based [20] brain tumor detection model by leveraging 2D multiplanar Magnetic Resonance Imaging (MRI) slices for model training and fine-tuning. To tackle these issues, Kang's team introduced PK-YOLO - a fresh take on YOLO built just to detect brain tumors through 2D MRI views from different angles.

2 Limitations of the Previous Work

While PK-YOLO shows state-of-the-art performance, the performance across three different planes exhibits its limitation. This model is trained only on horizontal slices, although brain imaging naturally covers three slices such as horizontal (axial), front-to-back (coronal), and side-to-side (sagittal). Since every angle or view shows different information, using only one slice can lead to gaps when processing others.

This becomes more obvious when looking at the experimental results from the paper. As shown in Table 1, PK-YOLO performs well in axial views, which is expected because the backbone is already pretrained in axial views. However, the performance drop can be found when switching planes from axial views to coronal or sagittal views. For instance, PK-YOLO's mAP$_{50}$ scored 0.947 in axial dataset, but its metric dropped to 0.805 and 0.582 in coronal and sagittal planes dataset, respectively.

This performance gap between planes is due to how PK-YOLO relies on one backbone pretrained in axial views for every plane. Since this backbone is optimized only for axial views, it ends up using identical features even when dealing with images that look quite different. Without a way to adapt to characteristics of each plane, performance of the model changes depending on which MRI view is used.

Even though PK-YOLO outperforms other models such as RT-DETR or LW-DETR in the axial plane, it still struggles when plane-shift occurs on MRI angles.

3 Related Work

Research on brain tumor detection using deep learning largely intersects with three key areas: DETR framework, brain tumor detection, and multiplanar MRI analysis. The PK-YOLO paper introduces contributions in all three areas, and the prior work they reference can be grouped into the following themes.

3.1 DETR Framework

Object detection methods using the DETR (DEtection TRansformer) [2] system have improved much better recently, mostly thanks to smarter training strategies and backbone pretraining. The first versions, like UP-DETR [7], introduced unsupervised pretraining which showed that self-taught features can help boost accuracy later on. Later works, such as Group DETR v2 [5] or Co-DETR [27], advanced the DETR performance ahead by exploring how both encoder and decoder are pretrained while mixing up label matching ways, scoring high on benchmarks such as MS COCO.

Along with these developments, lighter and faster DETR versions have appeared. Instead of relying on pretraining, RT-DETR [26] uses decoder layers to offer speed options on demand. LW-DETR [4] increases efficiency using special pretraining strategies made just for it. By adding salience-guided supervision during learning, Salience DETR [9] becomes stable like speeding up training while keeping the results reliable. Despite these advances, according to the PK-YOLO study, models based on DETR aren't used much yet for detecting brain tumors in MRI scans. Moreover, current DETR still struggles to match YOLO-based systems in the medical domain because they need heavy computing power and adapt poorly to new domains.

3.2 Brain Tumor Detection

YOLO-based models pop up a lot in regular image detection, but they don't show up much in medical imaging, especially when it comes to finding brain tumors. Lately, researchers tried to adapt them for hospital use. For example, RCS-YOLO [12] uses focused area blocks that help pin down where tumors sit. On the other hand, BGF-YOLO [13] upgrades YOLOv8 by blending multi-level detail filters to sharpen tiny tumor clues. Finally, different versions like YOLOv5 [10], YOLOv8, or YOLOv9 [24] got tested on brain MRIs too. Still, problems stick around, especially with spotting smaller growths and working well across varied scan angles like axial, coronal and sagittal.

3.3 Multiplanar MRI Analysis

Multiplanar MRI scans use axial, coronal and sagittal slices to detect tumors that might look different depending on angle. Prior research has looked into ways to combine these angles effectively. Instead of one method, Piantadosi [18] used groups of 2D networks to better separate MRI tissue types in brain images. Similarly, the MPS-FFA model [16] introduces mixed features from various scales and planes to help sort Alzheimer's cases, showing how blending directional data boosts accuracy. Some research into detecting objects like Barbato and Menga's use of YOLOv5m, found low average accuracy when checking multi-view MRI images, showing it's difficult to detect lesions from different body angles. In those works, common issues pop up now and then: lesions change size or shift place depending on the view, thinner image layers lead to more tiny lesions appearing, and also training one system to handle all three imaging directions remains difficult.

3.4 Mixture-of-Experts

Integrating various knowledge into one for machine learning is especially important as it can enable utilization of pretrained knowledge. N. Shazeer *et al.* [21] proposed a neural network architecture called Mixture-of-Experts which considers each model as an expert and uses a gating layer to combine the models into one. Its potential lies in computational efficiency, scalability, and leveraging specialized expert modules. G. Chen *et al.* [3] used router module for multimodal large language model (MLLM) called LION, treating each AdaptFormer [6] as an expert. Each AdaptFormer is allocated to perform image-level and region-level task given the input image and prompt, and LION adopts a router module to control the ratio between image-level and region-level knowledge. In terms of medical machine learning area, some works [15, 25] employ mixture-of-experts architecture to handle brain tumor detection problem.

4 Method

In this section, we present the Mixture PK-YOLO model. The proposed model aims to utilize 3D MRI slice knowledge to bridge the gap caused by the domain-shift from the axial plane to other planes. As discussed in Section 2, one of the most notable limitations of the original PK-YOLO stems from its performance drop when sagittal plane images are given to the model compared to axial plane images. Inspired by the mixture-of-experts models, Mixture PK-YOLO employs three backbone models, compared to the PK-YOLO with one single backbone model, each is treated as an expert to its corresponding plane data, as shown in Figure 1. This approach is to capture each plane-specific visual cues for each backbone models.

Pretraining process of the SparK RepViT backbone models is the same with the original PK-YOLO model. Using the RepViT [23] model as a foundation, it applies SparK [22] technique to incorporate

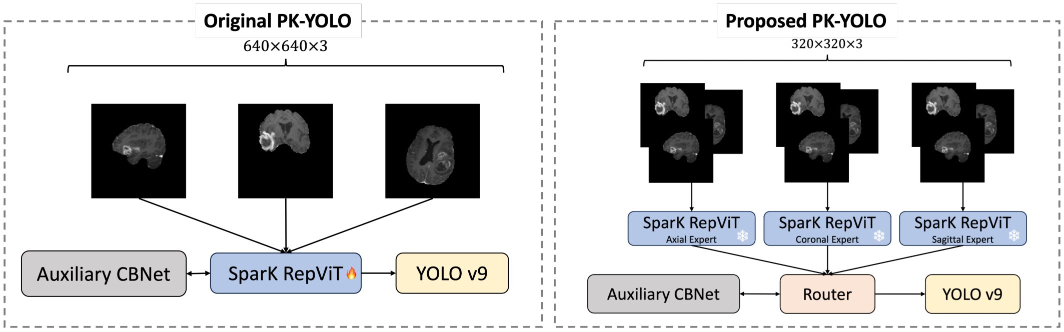

Figure 1: Comparison between the original PK-YOLO and the proposed PK-YOLO. Original PK-YOLO model uses single pretrained backbone model, namely SparK RepViT, fed with axial plane data. Mixture PK-YOLO employs three backbones treating each backbone as an expert to its corresponding plane data. Then a router module integrates the outputs from the backbones into one, then pass it to YOLO model.

domain-specific knowledge. During the SparK pretraining process, the model is fed with plane-specific brain tumor images to reconstruct the images with randomly masked patches. Given this, each SparK RepViT backbone in Mixture PK-YOLO is equipped with specialized plane-specific brain tumor knowledge, acting as an expert for its corresponding plane data.

Since the proposed model have added additional two more backbone models, it is necessary to combine the outputs from the backbone models into one. As shown in Figure 1, router module is adopted to combine the outputs from the SparK RepViT backbone models, then feed the outputs to the auxiliary CBNet and YOLOv9 as PK-YOLO did. Detailed visual illustration of the router module from the proposed Mixture PK-YOLO can be found in 2.

We assume three anatomical MRI planes, such that $k \in \{1, 2, 3\}$ corresponding to the planes $\{\text{axial}, \text{coronal}, \text{sagittal}\}$. Given the planes, each plane-specific SparK RepViT backbone expert is denoted as $F_k(X)$, where $X \in \mathbb{R}^{C\times H\times W}$. In this work, we assume all three channels are used (*i.e.*, $C = 3$) and fix the image size to $H \times W = 320 \times 320$. Each backbone produces a set of multi-scale feature maps $F_k^{(n)}(X) \in \mathbb{R}^{C_n\times H_n\times W_n}$, $\forall n = 1, \cdots, 5$, representing an image representation with different dimension sizes for scale n. For each scale n and plane k, the spatial feature map is reduced to a channel vector,

$$\tilde{F}_k^{(n)}(X) = AAP(F_k^{(n)}(X)) \in \mathbb{R}^{C_n}, \tag{1}$$

where $AAP(\cdot)$ is adaptive average pooling layer. Then, the reduced feature maps are given to the score function that maps,

$$\begin{aligned} z_k^{(n)} &= g_n(\tilde{F}_k^{(n)}(X)) \in \mathbb{R} \qquad (2)\\ \rightarrow \boldsymbol{z}^{(n)} &= [z_1^{(n)}, z_2^{(n)}, z_3^{(n)}] \in \mathbb{R}^3. \qquad (3) \end{aligned}$$

In this work, the score function $g_n(\cdot)$ is replaced with two-layered simple linear layers with ReLU layer inplaced in between.

Then, the weights are measured for each scores of backbone models' outputs as such,

$$w_k^{(n)} = \frac{\exp{(z_k^{(n)})}}{\sum_{i=1}^{3} \exp{((z_i^{(n)}))}}. \tag{4}$$

The weight $w_k^{(n)}$ decides which plane to give a higher importance than the other plane information. Given the weights, the router module controls the importance of each backbone's output as follows,

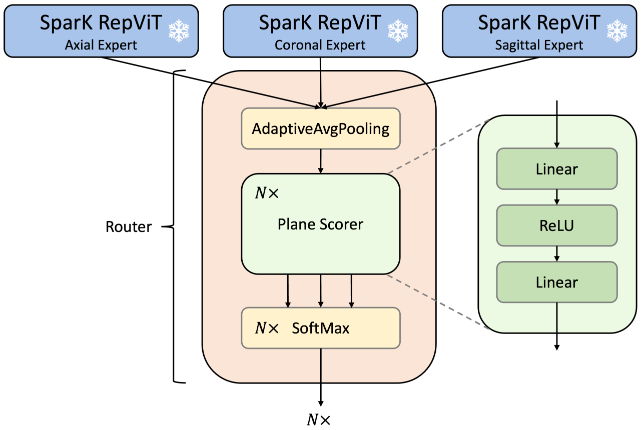

Figure 2: Router module from Mixture PK-YOLO to control the importance of plane-specific information.

Table 2: Model size comparison. The reduction rate indicates how much smaller Mixture PK-YOLO is compared to each model.

Model	# Params (M)	Reduction vs. Mixture PK-YOLO	Image Size
YOLOv8x	68.2	18.62%	640
PK-YOLO	109.4	49.26%	640
Mixture PK-YOLO	55.5	–	320

$$Z^{(n)} = \sum_{i=1}^{3} w_i^{(n)} \cdot F_i^{(n)}(X). \tag{5}$$

Unlike original PK-YOLO, Mixture PK-YOLO completely freezes the pretrained backbone models to remain the pretrained knowledge on brain tumor images across the planes. As PK-YOLO uses a single backbone model pretrained on axial data to capture the general knowledge of brain tumor, it is inevitable to unfreeze the backbone to adapt it to different plane data (*i.e.*, coronal and sagittal data). Whereas Mixture PK-YOLO doesn't require a fine-tuning process like the original PK-YOLO, since it employs three specialized backbone models, producing plane-specific brain tumor knowledge. Not only this, it can significantly reduce the amount of parameters required for training by freezing the pretrained backbone models. By freezing the SparK RepViT backbone, we are able to save almost 50% of parameters to be trained, with PK-YOLO requiring 109.4 million parameters, compared to Mixture PK-YOLO requiring 55.5 million parameters. Due to its reduction in the number of parameters to train, training overhead can be significantly reduced.

5 Experiments

In this section, we evaluate our Mixture PK-YOLO model with the original PK-YOLO and the second best performed model, YOLOv8x [11].

5.1 Ablation Study on Freezing Backbone Models

We conducted an ablation study on freezing the three SparK RepViT backbone models for the proposed Mixture PK-YOLO. Table 4 shows the number of training parameters, and the overall model performance when the backbones are frozen and unfrozen.

While the Mixture PK-YOLO with unfrozen backbones achieves a slightly better performance than the one with frozen backbones in some metrics, the frozen model's overall performance exceeds the unfrozen model. For instance, all of the metrics from sagittal plane from the frozen model show

Table 3: TODO: Put caption inside.

Model	Plane	Image Size	Precision	Recall	mAP_{50}	$\text{mAP}_{50:95}$
YOLOv8x	Axial	640	**0.894**	0.836	0.908	0.656
PK-YOLO		640	0.858	0.896	0.947	**0.681**
Mixture PK-YOLO		640	0.858	0.765	0.865	0.523
Mixture PK-YOLO		320	0.839	**0.913**	**0.950**	0.630
YOLOv8x	Coronal	640	0.672	0.650	0.697	0.524
PK-YOLO		640	**0.834**	**0.793**	**0.805**	**0.689**
Mixture PK-YOLO		640	0.457	0.719	0.495	0.265
Mixture PK-YOLO		320	0.460	0.771	0.549	0.366
YOLOv8x	Sagittal	640	0.414	0.782	0.533	**0.385**
PK-YOLO		640	**0.476**	**0.845**	**0.582**	0.382
Mixture PK-YOLO		640	0.449	0.574	0.401	0.213
Mixture PK-YOLO		320	0.452	0.780	0.455	0.243

Table 4: Ablation study on freezing backbones. Bold numbers show higher performance. Not only does freezing the backbones reduce the amount of training parameters, but also increases the overall model performance.

Frozen Backbones	# Params (M)	Plane	Precision	Recall	mAP_{50}	$\text{mAP}_{50:95}$
✗	217.6	Axial	**0.864**	0.802	0.930	**0.632**
		Coronal	**0.483**	**0.781**	0.544	0.354
		Sagittal	0.403	0.648	0.392	0.221
✓	55.5	Axial	0.839	**0.913**	**0.950**	0.630
		Coronal	0.460	0.771	**0.549**	**0.366**
		Sagittal	**0.452**	**0.780**	**0.455**	**0.243**

higher performance than those of the unfrozen model. This is important, as the purpose of Mixture PK-YOLO is to reduce the performance gap between the axial plane and especially the sagittal plane.

Another advantage of freezing backbones is the significant reduction in the number of training parameters of the model. Unfreezing the backbones means that the fine-tuning of three different SparK RepViT models must be done, which drastically increases the number of parameters. Mixture PK-YOLO with unfrozen backbones requires 217.6 million parameters to train, which is about twice as many as the original PK-YOLO. On the other hand, the model with frozen backbones requires 55.5 million parameters to train, which significantly reduces the total amount of parameters in the model while showing its competent performance throughout the planes.

6 Conclusion

References

[1] S. Asif, Y. Wenhui, S. ur Rehman, Q. ul ain, K. Amjad, Y. Yueyang, S. Jinhai, and M. Awais. Advancements and prospects of machine learning in medical diagnostics: unveiling the future of diagnostic precision. *Archives of Computational Methods in Engineering*, 32(2):853–883, 2025.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[3] G. Chen, L. Shen, R. Shao, X. Deng, and L. Nie. Lion: Empowering multimodal large language model with dual-level visual knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26540–26550, 2024.

[4] Q. Chen, X. Su, X. Zhang, J. Wang, J. Chen, Y. Shen, C. Han, Z. Chen, W. Xu, F. Li, et al. Lw-detr: A transformer replacement to yolo for real-time detection. *arXiv preprint arXiv:2406.03459*, 2024.

[5] Q. e. a. Chen. Group detr v2: Strong object detector with encoder-decoder pretraining. *arXiv:2211.03594*, 2022.

[6] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[7] Z. Dai, B. Cai, Y. Lin, and J. Chen. Unsupervised pre-training for detection transformers. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12772–12782, 2022.

[8] S. Das and R. S. Goswami. Advancements in brain tumor analysis: a comprehensive review of machine learning, hybrid deep learning, and transfer learning approaches for mri-based classification and segmentation. *Multimedia Tools and Applications*, 84(23):26645–26682, 2025.

[9] X. Hou, M. Liu, S. Zhang, P. Wei, and B. Chen. Salience detr: Enhancing detection transformer with hierarchical salience filtering refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17574–17583, 2024.

[10] G. Jocher. Yolov5 by ultralytics, may 2020. *URL https://github. com/ultralytics/yolov5*, 2023.

[11] G. Jocher, A. Chaurasia, and J. Qiu. Yolo by ultralytics. https://github.com/ultralytics/ultralytics, 2023. Accessed: 2025-11-30.

[12] M. Kang, C.-M. Ting, F. F. Ting, and R. C.-W. Phan. Rcs-yolo: A fast and high-accuracy object detector for brain tumor detection. In *International conference on medical image computing and computer-assisted intervention*, pages 600–610. Springer, 2023.

[13] M. Kang, C.-M. Ting, F. F. Ting, and R. C.-W. Phan. Bgf-yolo: Enhanced yolov8 with multiscale attentional feature fusion for brain tumor detection. In *MICCAI*, 2024.

[14] M. Kang, F. F. Ting, R. C.-W. Phan, and C.-M. Ting. Pk-yolo: Pretrained knowledge guided yolo for brain tumor detection in multiplanar mri slices. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3732–3741. IEEE, 2025.

[15] S. Li, X. Sui, X. Luo, X. Xu, Y. Liu, and R. Goh. Medical image segmentation using squeeze-and-expansion transformers. *arXiv preprint arXiv:2105.09511*, 2021.

[16] F. Liu, H. Wang, S.-N. Liang, Z. Jin, S. Wei, and X. Li. Mps-ffa: A multiplane and multiscale feature fusion attention network for alzheimer's disease prediction with structural mri. *Computers in Biology and Medicine*, 157:106790, 2023.

[17] P. Mittal et al. From black box ai to xai in neuro-oncology: a survey on mri-based tumor detection. *Discover Artificial Intelligence*, 5(1):1–21, 2025.

[18] G. Piantadosi, M. Sansone, R. Fusco, and C. Sansone. Multi-planar 3d breast segmentation in mri via deep convolutional neural networks. *Artificial Intelligence in Medicine*, 103:101781, 2020.

[19] N. Rasool and J. I. Bhat. Brain tumour detection using machine and deep learning: a systematic review. *Multimedia tools and applications*, 84(13):11551–11604, 2025.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016.

[21] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[22] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan. Designing bert for convolutional networks: Sparse and hierarchical masked modeling. *arXiv preprint arXiv:2301.03580*, 2023.

[23] A. Wang, H. Chen, Z. Lin, J. Han, and G. Ding. Repvit: Revisiting mobile cnn from vit perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15909–15920, 2024.

[24] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. In *European conference on computer vision*, pages 1–21. Springer, 2024.

[25] R. K. Yadav, M. Kumar, and A. Nandi. A scalable brain tumor diagnosis from large-scale mri datasets using cnn-vit and expert-attention fusions. *International Journal of Computational Intelligence Systems*, 18(1):1–25, 2025.

[26] Y. e. a. Zhao. Detrs beat yolos on real-time object detection. In *CVPR*, 2024.

[27] Z. Zong, G. Song, and Y. Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023.

A Appendix

A.1 Detailed Performance Comparison

Dataset	Model	Precision	Recall	mAP50	mAP50:95
axial t1ce 2 class	UP-DETR [7]	-	-	0.754	0.477
	RT-DETR-X [58]	0.662	0.568	0.538	0.347
	LW-DETR-xlarge [4]	-	-	0.926	0.687
	Co-DETR with Swin L [60]	-	-	0.893	0.584
	Salience DETR with FocalNet-L [19]	-	-	0.816	0.526
	RCS-YOLO [25]	0.944	0.839	0.839	0.573
	BGF-YOLO [26]	0.941	0.789	0.941	0.579
	YOLOv5x [21]	0.741	0.679	0.828	0.596
	YOLOv8x [23]	0.894	0.836	0.908	0.656
	YOLOv9-E [50]	0.838	0.877	0.935	0.667
	YOLOv10-X [46]	0.743	0.821	0.832	0.558
	Mamba YOLO-L [52]	0.891	0.753	0.915	0.666
	PK-YOLO (Ours)	0.858	0.896	0.947	0.681
Axial 320	Ours	0.839	0.913	0.950	0.630
Axial 640	Ours	0.858	0.765	0.865	0.523
coronal t1ce 2 class	UP-DETR	-	-	0.236	0.186
	RT-DETR-X	0.742	0.592	0.575	0.407
	LW-DETR-xlarge	-	-	0.723	0.549
	Co-DETR with Swin L	-	-	0.510	0.304
	Salience DETR with FocalNet-L	-	-	0.520	0.343
	RCS-YOLO	0.493	0.884	0.574	0.369
	BGF-YOLO	0.494	0.889	0.593	0.417
	YOLOv5x	0.648	0.690	0.681	0.466
	YOLOv8x	0.672	0.650	0.697	0.524
	YOLOv9-E	0.552	0.788	0.729	0.526
	YOLOv10-X	0.519	0.656	0.605	0.423
	Mamba YOLO-L	0.641	0.770	0.758	0.539
	PK-YOLO (Ours)	0.834	0.793	0.805	0.689
Coronal 320	Ours	0.460	0.771	0.549	0.366
Coronal 640	Ours	0.457	0.719	0.495	0.265
sagittal t1ce 2 class	UP-DETR	-	-	0.231	0.137
	RT-DETR-X	0.441	0.435	0.395	0.254
	LW-DETR-xlarge	-	-	0.471	0.343
	Co-DETR with Swin L	-	-	0.496	0.290
	Salience DETR with FocalNet-L	-	-	0.509	0.346
	RCS-YOLO	0.500	0.779	0.515	0.357
	BGF-YOLO	0.485	0.792	0.545	0.347
	YOLOv5x	0.469	0.840	0.561	0.370
	YOLOv8x	0.414	0.782	0.533	0.385
	YOLOv9-E	0.437	0.869	0.534	0.383
	YOLOv10-X	0.536	0.525	0.544	0.361
	Mamba YOLO-L	0.477	0.842	0.559	0.385
	PK-YOLO (Ours)	0.476	0.845	0.582	0.382
Sagittal 320	Ours	0.452	0.780	0.455	0.243
Sagittal 640	Ours	0.449	0.574	0.401	0.213