ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

Analyzing the Decoder Bottleneck in SAM-Style 3D Segmentation Distillation

Haojun Pei*

School of Information Science and Engineering, Dalian Polytechnic University, Dalian, China
**Corresponding author: phj2219075277@gmail.com*

Abstract: *SegVol builds on the Segment Anything Model (SAM) for 3D medical image segmentation. It handles many anatomical targets well, but at 180.9M parameters, it is too heavy for routine clinical hardware. We study how encoder weight initialization and decoder configuration affect the outcome of knowledge distillation for this model. We test 7 configurations. The results point to two decisive factors—the decoder needs to be trainable, and encoder weights should be copied from alternating teacher layers rather than consecutive ones. Our best student uses an independent decoder fine-tuned with pseudo-labels, plus skip-layer weight transfer from teacher layers {0,2,4,6,8,10}. It keeps 95.1% of the teacher's Dice (0.557 vs. 0.586), cuts 23.5% of parameters (138.4M vs. 180.9M), and runs 43.1% faster. Centered Kernel Alignment (CKA) shows why a frozen shared decoder fails: the deepest student layer can only match teacher layer 10, whereas the independent decoder's last layer reaches layer 11. Decoder trainability is the critical lever in volumetric segmentation distillation.*

Keywords: *Knowledge distillation, SAM, 3D medical image segmentation, Model compression, Vision transformer*

1. Introduction

Medical image segmentation is essential for clinical diagnosis and treatment planning. The Segment Anything Model (SAM) [1] and its 3D adaptations such as SegVol [2] demonstrate remarkable zero-shot capabilities across diverse anatomies. However, these models are computationally demanding—SegVol's 12-layer Vision Transformer (ViT) encoder contains 180.9M parameters, challenging deployment on clinical hardware such as edge GPUs and hospital workstations.

Knowledge distillation (KD) [3] compresses large models by transferring knowledge from a teacher to a smaller student network. DistilBERT [4] compressed BERT by halving layers via KD with a frozen shared prediction head, and MobileSAM [5] distilled SAM's image encoder into a lightweight CNN while reusing the frozen mask decoder. In both cases, the teacher's decoder remains frozen and shared—this approach works well for classification tasks (softmax logits) and 2D SAM segmentation (lightweight decoder). However, applying the same approach to SAM-style 3D segmentation faces a unique challenge: the volumetric mask decoder operates in a higher-dimensional feature space where feature mismatches compound across the spatial depth of 3D volumes.

Furthermore, TinySAM [6] demonstrated full-stage KD for 2D SAM but preserved the original decoder architecture without investigating decoder-level effects, and DeiT [7] showed that distillation tokens can guide ViT training. Both approaches assume the decoder component can readily adapt to student encoder features. We show this assumption does not hold for 3D medical volumes—the volumetric decoder creates a representation bottleneck that fundamentally limits distillation performance.

In this paper, we systematically investigate how decoder configuration and encoder initialization affect knowledge distillation for 3D SAM-style segmentation. We compare shared frozen decoder versus independent trainable decoder, and evaluate five encoder initialization strategies. Figure 1 provides an overview of our experimental setup. Our results show that decoder trainability is a critical factor—a shared frozen decoder underperforms an independent decoder by 0.057 Dice (0.500 vs 0.557)—and that skip-layer initialization provides the best starting point for encoder features by spanning the full teacher depth.

Our contributions are: (1) systematic ablation quantifying the effect of decoder configuration (shared vs. independent, frozen vs. trainable) and encoder initialization strategy on SAM-style 3D

ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

distillation, (2) demonstration that decoder trainability contributes up to 0.057 Dice over a frozen shared decoder, (3) validation that skip-layer initialization outperforms first/last/random strategies (0.557 vs. 0.515/0.392/0.277), and (4) CKA-based analysis revealing the mechanism—shared decoder constrains representation depth in student layers, and independent decoder training enables the student's final layer to reach the deepest teacher representations.

2. Related Work

2.1 Knowledge Distillation for Model Compression

Knowledge distillation, introduced by Hinton et al. [3], transfers dark knowledge from a large teacher model to a smaller student by matching softened output distributions. The framework has been extended in numerous directions: FitNets [8] introduced intermediate feature alignment, Attention Transfer [9] matched attention maps, and Contrastive Representation Distillation [10] proposed contrastive objectives for representation transfer. In natural language processing, DistilBERT [4] demonstrated that layer-level initialization from the teacher combined with distillation losses enables effective compression of transformer models. Sanh et al. showed that initializing the student's layers from selected teacher layers (a strategy we adopt) provides a strong inductive bias that purely random initialization cannot match.

2.2 SAM and Efficient Variants

SAM [1] introduced a promptable segmentation framework with a ViT-based image encoder and a lightweight mask decoder. MobileSAM [5] distilled SAM's image encoder into a lightweight CNN while keeping the mask decoder frozen and shared. TinySAM [6] further pushed efficiency through full-stage KD but maintained the original decoder architecture. Importantly, both MobileSAM and TinySAM operate in the 2D domain, where the mask decoder is relatively simple and feature mismatches have limited spatial impact. In the 3D volumetric setting, the decoder processes significantly higher-dimensional features, making it more sensitive to encoder feature distribution shifts.

2.3 3D Medical Image Segmentation

U-Net [11] and its self-configuring variant nnU-Net [12] remain strong baselines for 3D medical image segmentation. SAM-inspired medical models include MedSAM [13] (2D medical adaptation), SAM-Med3D [14] (3D adaptation from scratch), and SegVol [2] (3D adaptation with text prompting). SegVol is particularly relevant as it achieves state-of-the-art zero-shot performance across 200+ anatomical categories through large-scale pre-training on 96K unlabeled CT volumes and fine-tuning on 6K labeled volumes. However, its 180.9M parameter count poses deployment challenges.

3. Method

3.1 SegVol Architecture Overview

Figure 1 illustrates the overall framework. SegVol [2] extends SAM to volumetric medical images with three components:

Image Encoder. A 12-layer Vision Transformer (ViT) with hidden size 768, 12 attention heads, and MLP dimension 3072. Input volumes of size (32×256×256) are processed via convolutional patch embedding with patch size (4×16×16). The encoder contains 156.0M parameters.

Prompt Encoder. Encodes point, box, and text prompts. Text prompts use a frozen CLIP text encoder followed by a learned linear projection from dimension 512 to 768. The prompt encoder enables interactive segmentation with multiple prompt modalities.

Mask Decoder. A lightweight 2-layer transformer with iterative refinement (6 iterations by default). The decoder takes encoder features and prompt embeddings as input and produces volumetric segmentation masks. Together with the prompt encoder, it comprises 24.9M parameters.

The total model contains 180.9M parameters and supports segmentation of over 200 anatomical categories.

ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

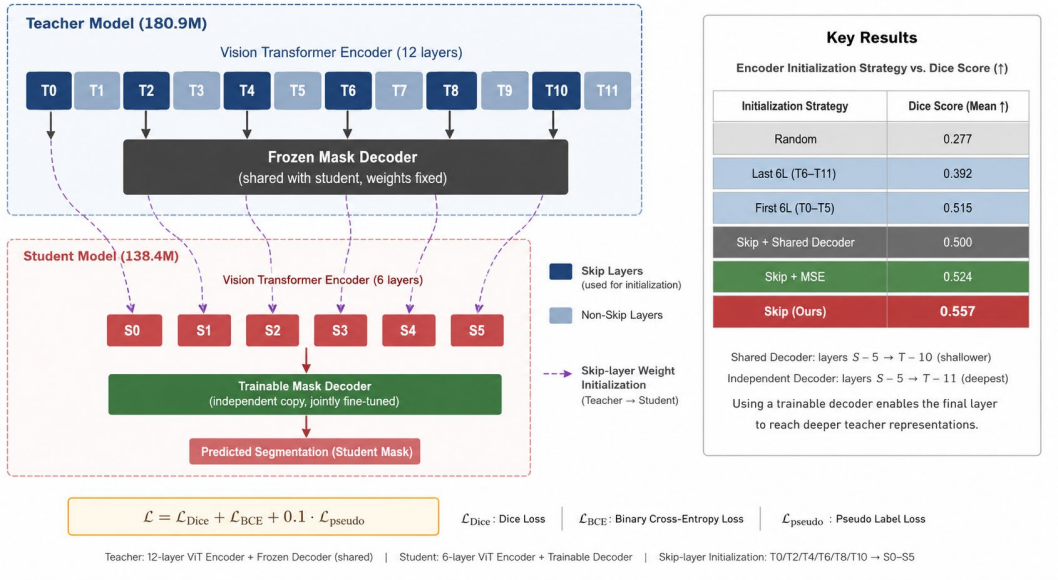

Figure 1: Overview of the SegVol knowledge distillation framework. (A) Shared frozen decoder configuration (B) Independent decoder configuration (ours)

3.2 Student Architecture Design

Our student model reduces the encoder from 12 to 6 ViT layers with 6 attention heads, resulting in 138.4M total parameters (23.5% reduction). Table 1 summarizes the architectural differences.

Table 1: Architectural comparison between teacher and student models.

Component	Teacher	Student	Change
Encoder layers	12	6	−50%
Attention heads	12	6	−50%
Hidden size	768	768	Unchanged
MLP dimension	3072	3072	Unchanged
Patch size	4×16×16	4×16×16	Unchanged
Encoder params	156.0M	113.5M	−27.2%
Decoder/prompt	**24.9M**	**24.9M**	**Unchanged**

The hidden size (768), MLP dimension (3072), and patch embedding configuration are preserved to maintain compatibility with the teacher's feature space and the mask decoder's input expectations. The mask decoder and prompt encoder are instantiated as separate instances, initialized from teacher weights, and are fully trainable during the distillation process.

3.3 Weight Initialization Strategies

We systematically explore five encoder initialization strategies to understand their impact on distillation performance:

- **Random:** Standard random weight initialization (Kaiming uniform). Serves as the lower bound.
- **First:** Initialize student layers 0,…,5 from teacher layers 0,…,5. Captures low-level features.
- **Last:** Initialize student layers 0,…,5 from teacher layers 6,…,11. Captures high-level semantic features.
- **Skip:** DistilBERT-style alternating initialization from teacher layers {0,2,4,6,8,10}. Covers the full encoder depth from low-level texture to high-level semantics.
- **Random Decoder:** Skip encoder initialization but with randomly initialized decoder (measures the contribution of decoder pre-training).

For the decoder component, we compare three configurations: (1) Teacher-initialized, trainable—weights are copied from the teacher but updated during training; (2) Teacher-initialized, frozen ("shared")—the student reuses the teacher's decoder without any weight updates; and (3) Random

ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

initialization—serving as a baseline for decoder contributions.

3.4 The Decoder Bottleneck Hypothesis

A central hypothesis of this work is that a shared frozen decoder creates a representation mismatch that limits distillation performance. When the decoder is frozen, the student encoder must produce features that exactly match the teacher's 12-layer feature space distribution. However, with only 6 layers, the student's encoder outputs reside in a fundamentally different embedding manifold that the frozen decoder was never trained to accommodate. This creates a bottleneck where the student encoder is forced to compromise between learning efficient compressed representations and matching the frozen decoder's preferred input distribution.

Our solution is to create an independent decoder instance, initialize its weights from the teacher, and allow joint fine-tuning with the student encoder. This enables the decoder to adapt its parameters to the student's compressed feature space. The independent decoder adds no new parameters beyond the student's existing 24.9M decoder budget—it simply changes the training regime from frozen to trainable.

Caveat on experimental design. We note that the "shared decoder" configuration simultaneously shares the architecture and freezes the weights, while the independent decoder both decouples the architecture and enables training. The observed 0.057 Dice gap between these conditions therefore reflects both effects—weight trainability and architectural independence—and the two factors cannot be isolated within our current ablation design. A control experiment with a shared but trainable decoder would disentangle these contributions and is noted as future work.

3.5 Training Setup

We train with ground-truth supervision (Dice loss plus binary cross-entropy) augmented with teacher-generated pseudo-labels at a weight of 0.1:

$$L = L\_Dice + L\_BCE + 0.1 \cdot L\_pseudo \quad (1)$$

Notably, we omit explicit logit-level distillation losses such as KL divergence or mean squared error on logits. Our ablation study (Section 4.2) demonstrates that pseudo-label data-level distillation combined with weight initialization is sufficient for the student to capture the teacher's knowledge. Adding logit-level MSE on softened outputs provided no benefit and in fact slightly degraded performance (0.524 vs 0.557), suggesting that the combination of good initialization and label-level supervision already provides a complete training signal.

Training hyperparameters: AdamW optimizer with learning rate 1×10^{-4}, weight decay 1×10^{-5}, batch size 1, and cosine annealing schedule with 2-epoch linear warmup over 100 total epochs. We use FP16 mixed precision training to reduce memory consumption.

4. Experiments

4.1 Experimental Setup

We train and evaluate on the AMOS22 dataset (M3D-SEG code 0002), which contains 48 CT volumes with 15 organ categories. The teacher model is the publicly available SegVol checkpoint from HuggingFace, which achieves a mean Dice score of 0.586 (±0.308 standard deviation) on this dataset. Dice scores are computed per (image, organ) pair across 642 non-empty pairs. Validation is performed on a held-out subset. All experiments use a single NVIDIA RTX 3090 GPU with text prompts for inference.

4.2 Ablation Study

Table 2 presents the results of our systematic ablation study across 7 configurations, including the teacher baseline. All student models use a 6-layer encoder. The column "%Tchr" indicates the percentage of teacher Dice retained. † indicates frozen teacher decoder; ‡ indicates random decoder initialization.

ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

Table 2: Ablation study across initialization strategies and decoder configurations.

Experiment	Enc. Init	Dice	%Tchr
Teacher (12L, 12H)	—	0.586±.308	100%
Random Init	Random	0.277±.258	47.3%
Last 6 Layers	{6,…,11}	0.392±.292	66.9%
Shared Decoder†	Skip	0.500±.311	85.4%
First 6 Layers	{0,…,5}	0.515±.314	87.9%
+MSE Distill	Skip	0.524±.308	89.4%
No Dec Init‡	Skip	0.435±.299	74.2%
Skip Init (ours)	**{0,2,4,6,8,10}**	**0.557±.306**	**95.1%**

Key findings from this comparison:

(1) Encoder initialization matters. Random init gives 0.277 Dice. Any teacher-based initialization lifts this above 0.392. Pre-trained weights supply indispensable inductive bias.

(2) Decoder setup is the largest factor. Shared frozen decoder (0.500) trails independent trainable decoder (0.557) by 0.057 Dice, or 9.7 percentage points. No other single change produces a gap this wide. A frozen decoder forces the student encoder to target a feature distribution it cannot fully match.

(3) Shallow layers transfer better. First-6 layers (0.515) beat Last-6 (0.392) by a wide margin. Low-level texture and edge patterns appear more universal than dataset-specific high-level semantics.

(4) Skip beats consecutive. Alternating selection {0,2,4,6,8,10} (0.557) outperforms {0,…,5} (0.515) by 0.042 Dice. Spanning the full depth of the teacher—from early textures to late semantics—yields richer initialization.

(5) Logit-level distillation helps little. Adding MSE to the loss drops performance to 0.524 vs. the 0.557 baseline. Pseudo-label supervision plus weight initialization already captures the teacher signal; extra output-level alignment introduces conflicting gradients.

4.3 Per-Organ Analysis

Table 3 presents per-organ Dice scores for all 15 categories, ranked by student retention rate. Tch = Teacher, Stu = Student, Ret = Retention rate. Organs are ranked by retention.

Table 3: Per-organ Dice comparison across all 15 categories.

Organ	Tch	Stu	Ret
Liver	0.923	0.920	99.7%
Aorta	0.809	0.795	98.2%
Esophagus	0.283	0.275	97.0%
R. Kidney	0.808	0.783	96.9%
L. Kidney	0.811	0.782	96.4%
Spleen	0.873	0.828	94.9%
Stomach	0.691	0.651	94.2%
Duodenum	0.326	0.305	93.6%
Pancreas	0.519	0.483	93.1%
Postcava	0.672	0.615	91.4%
Bladder	0.418	0.375	89.8%
Gallbladder	0.381	0.314	82.4%
Prostate/Uterus	0.488	0.535	109.6%
R. Adrenal	0.115	0.086	74.8%
L. Adrenal	0.142	0.092	64.8%

The student holds steady on most organs: 11 of 15 keep 89–100% of teacher Dice. Large, high-contrast targets (liver 99.7%, aorta 98.2%) lose almost nothing. Small organs drop more (adrenals 64.8–74.8%), consistent with the known difficulty of segmenting tiny structures with a compressed encoder. Prostate/uterus shows an apparent boost (0.535 vs. 0.488), but this likely reflects noise in the teacher checkpoint rather than a real improvement, given the teacher's modest overall score on these categories.

ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

4.4 Inference Speed and Memory

Table 4 reports inference benchmarks at batch size 1 with text prompt input.

Table 4: Inference benchmark (batch size 1, text prompt, NVIDIA RTX 3090).

Metric	Teacher	Student	Change
Parameters	180.9M	138.4M	−23.5%
Total inference (ms)	65.4	37.2	−43.1%
Encoder only (ms)	49.4	20.6	−58.3%
Peak VRAM (GB)	1.69	1.47	−12.7%

The 43.1% total inference speedup is driven primarily by the encoder (58.3% reduction in encoder time), since halving the number of ViT layers more than halves computation due to reduced attention overhead. Decoder and prompt encoder times remain unchanged as these components are not compressed in our current design. The modest VRAM reduction (12.7%) reflects the dominant memory footprint of the shared CLIP text encoder weights, which are identical between teacher and student.

4.5 Feature Similarity Analysis via CKA

We use Centered Kernel Alignment (CKA) [15] to measure the similarity between student and teacher encoder block representations after training. CKA provides a normalized similarity score between 0 (completely dissimilar) and 1 (identical), robust to orthogonal transformations and isotropic scaling. Figure 2 presents the side-by-side CKA comparison between the shared decoder and independent decoder conditions.

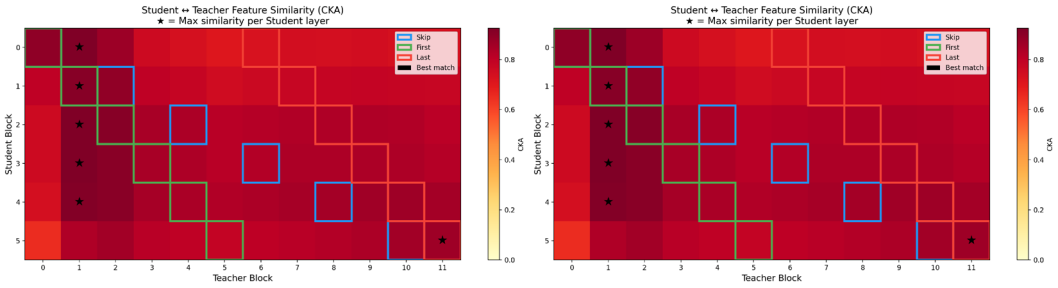

Figure 2: CKA feature similarity between Student (6-layer) and Teacher (12-layer) encoder blocks after training. Left: Shared frozen decoder. Right: Independent trainable decoder (ours). Stars mark the best-matching teacher layer for each student block.

Shared decoder limits depth. With a frozen shared decoder, student blocks S-0 to S-3 all map to teacher block T-1 (CKA >0.85). The last two blocks, S-4 and S-5, only reach T-10—the teacher's deepest layer T-11 is never matched. The frozen decoder keeps the student from developing representations at the deepest level.

Independent decoder goes deeper. Under independent decoder training, early layers still converge to T-1 (S-0 through S-4, CKA >0.90), but S-5 now reaches T-11. This is the single condition where a student layer hits the teacher's deepest block. Removing the decoder constraint lets the student push its most abstract features to the final layer.

Skip init guides, not enforces. The initial layer mapping {0,2,4,6,8,10} does not survive training intact (0/6 preserved for independent, 2/6 for shared). The student develops its own compression pathway. The value of skip initialization is exposing the student to features across the full teacher depth early in training, not locking in a fixed layer correspondence.

5. Discussion

Our experiments show a clear ordering of which factors matter most. Decoder setup (shared frozen vs. independent trainable, 0.057 Dice gap) outweighs encoder init strategy (best vs. worst among teacher-initialized variants, 0.044 Dice gap). The takeaway is that decoder trainability is the primary lever for compressing SAM-style 3D models. CKA explains why: a frozen decoder caps the student's deepest layer at T-10, while training the decoder lets it reach T-11.

Relationship to prior work. DistilBERT [4] and MobileSAM [5] both succeeded with frozen shared

ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

heads. DistilBERT's head is a single linear layer—insensitive to encoder drift. MobileSAM's 2D decoder has limited spatial dimensionality. Our 3D case is different: the high-dimensional volumetric decoder magnifies even small feature mismatches, so keeping it frozen becomes the dominant bottleneck.

Limitations. First, the speed gain is encoder-only (58.3%). The decoder and prompt encoder stay the same size. Shrinking those components is a separate problem not addressed here.

Second, we only test on AMOS22 (M3D-SEG 0002, 15 organ classes). Whether the decoder bottleneck generalizes to other anatomies or imaging protocols needs confirmation on additional datasets.

Third, our student still has 138.4M parameters. The 23.5% drop is real but smaller than the 40–50% reductions DistilBERT achieved in NLP. Pushing compression further without losing segmentation accuracy remains hard.

Fourth, the teacher checkpoint (public HuggingFace) scores 0.586 Dice here, well below the originally reported 0.85 [2]. We attribute the gap to checkpoint version mismatch and evaluation protocol differences (single-dataset fine-tuning vs. full benchmark). Because of this, the absolute Dice numbers in this paper are not competitive with SOTA segmentation models, and cases where the student appears to beat the teacher likely reflect teacher noise. The core contribution—the relative comparison between decoder strategies—is an internal ablation whose validity rests on controlled experimental conditions, not on the teacher's absolute performance.

Fifth (noted in Section 3.4), the shared-decoder condition bundles architecture sharing with weight freezing. A shared-but-trainable decoder control would separate these two factors and might shrink the 0.057 gap. This experiment is the most immediate next step.

Future work. Beyond the shared-trainable control, we plan: (1) validation on BTCV and AMOS benchmarks, (2) replication with a stronger teacher, (3) decoder-level compression, and (4) combining pseudo-label distillation with feature-level and relation-level KD objectives.

6. Conclusion

From 7 controlled experiments, we conclude that decoder trainability and broad encoder weight coverage drive effective compression of SAM-style 3D models. Our independent decoder with skip-layer init keeps 95.1% of the teacher's Dice (0.557 vs. 0.586), reduces parameters by 23.5%, and accelerates inference by 43.1%. CKA shows the mechanism: a frozen decoder blocks the student's deepest layer from reaching beyond teacher layer T-10. Making the decoder trainable removes this cap, letting the final student layer match T-11. For volumetric segmentation models, freeing the decoder to adapt is more important than precisely matching teacher layer assignments.

References

[1] A. Kirillov et al., "Segment Anything," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2023, pp. 3992–4003.
[2] Y. Du, F. Bai, T. Huang, and B. Zhao, "SegVol: Universal and Interactive Volumetric Medical Image Segmentation," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2024.
[3] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in NIPS Deep Learning Workshop, 2014, arXiv:1503.02531.
[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, Faster, Cheaper and Lighter," in Proc. EMC2 Workshop (NeurIPS), 2019.
[5] C. Zhang et al., "Faster Segment Anything: Towards Lightweight SAM for Mobile Applications," arXiv:2306.14289, 2023.
[6] H. Shu et al., "TinySAM: Pushing the Envelope for Efficient Segment Anything Model," arXiv:2312.13789, 2023.
[7] H. Touvron et al., "Training Data-Efficient Image Transformers & Distillation Through Attention," in Proc. Int. Conf. Mach. Learn. (ICML), PMLR 139, 2021, pp. 10347–10357.
[8] A. Romero et al., "FitNets: Hints for Thin Deep Nets," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.
[9] S. Zagoruyko and N. Komodakis, "Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer," in Proc. Int. Conf. Learn. Represent. (ICLR),

ISSN 2616-5775 Vol. 9, Issue 6: 94-101, DOI: 10.25236/AJCIS.2026.090611

2017.
[10] Y. Tian, D. Krishnan, and P. Isola, "Contrastive Representation Distillation," in Proc. Int. Conf. Learn. Represent. (ICLR), 2020.
[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI), LNCS 9351, 2015, pp. 234–241.
[12] F. Isensee et al., "nnU-Net: a Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation," Nature Methods, vol. 18, no. 2, pp. 203–211, 2021.
[13] J. Ma et al., "Segment Anything in Medical Images," Nature Communications, vol. 15, no. 1, Art. no. 654, 2024.
[14] H. Wang et al., "SAM-Med3D: Towards General-Purpose Segmentation Models for Volumetric Medical Images," in Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops, 2024.
[15] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of Neural Network Representations Revisited," in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 3519–3529.