Article

Not peer-reviewed version

Optimizing Swin-UNet with Search and Rescue Algorithm for Memory-Efficient Liver Tumor Segmentation on Edge Devices

Wail M. Idress , Yuqian Zhao * , Laeeq Aslam , Muhammad Asim * , Sayyed Shahid Hussain * , Sajid Shah , Mohammed ELAffendi

Posted Date: 13 March 2025

doi: 10.20944/preprints202503.0955.v1

Keywords: liver tumor segmentation; Swin-UNet optimization; memory-efficient deep learning; edge computing for medical imaging; Search and Rescue algorithm; clinical deployment of deep learning; edge AI

 doi:10.20944/preprints202503.0955.v1

Article

Optimizing Swin-UNet with Search and Rescue Algorithm for Memory-Efficient Liver Tumor Segmentation on Edge Devices

Wail M. Idress [1,2], Yuqian Zhao [1,*], Laeeq Aslam [1], Muhammad Asim [3,*], Sayyed Shahid Hussain [1], Sajid Shah [3] and Mohammed ELAffendi [3]

[1] School of Automation, Central South University, Changsha 410083, Hunan, China

[2] Department of Electrical and Electronic Engineering, Omdurman Islamic University, Omdurman 14415, Sudan

[3] EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

* Correspondence: zyq@csu.edu.cn (Y.Z.); masim@psu.edu.sa (M.A.)

Abstract: Liver cancer poses a significant global health challenge, necessitating precise tumor segmentation in CT scans for diagnosis and treatment. While deep learning models like U-Net and Vision Transformers show promise, their computational demands hinder edge deployment. To address this gap, we propose an optimized Swin-UNet framework enhanced by the Search and Rescue (SAR) algorithm, enabling real-time edge computing without compromising the model's performance. This work proposes a hybrid objective function with quadratic penalties for model compression and area under the curve(AUC). The models are trained using focal AUC loss to mitigate class imbalance. Evaluations on 3DIRCADb, LiTS, and MSD datasets show state-of-the-art performance, with Dice scores of 94.78%, 89.06%, and 88.95%, respectively, and an 80.3% parameter reduction versus baselines. The solution achieves efficient segmentation on edge devices (e.g., Jetson Nano), with a Volume Overlap Error of 1.73% (MSD) and Relative Volume Difference of 0.23% (3DIRCADb), outperforming existing methods. This work advances memory-efficient deep learning for clinical deployment, enabling AI-driven diagnostics in low-resource settings.

Keywords: liver tumor segmentation; Swin-UNet optimization; memory-efficient deep learning; edge computing for medical imaging; Search and Rescue algorithm; clinical deployment of deep learning; edge AI

1. Introduction

Liver cancer (LC) is a significant global health concern, ranking as one of the most frequently diagnosed cancers and the leading cause of cancer-related deaths. Recent global statistics position liver cancer as the sixth most prevalent cancer, following breast, lung, colorectal, prostate and gastric cancers and as the third leading cause of cancer mortality. In 2020, approximately 905,700 new cases and 830,200 deaths from LC occurred worldwide [1]. Hepatocellular carcinoma (HCC), the most common form of LC, accounts for nearly 90% of all cases [2]. Various imaging techniques such as ultrasound, elastography, MRI and CT scans detect LC, with CT scans providing detailed images of internal structures [3]. However, CT imaging encounters challenges in detecting and diagnosing liver abnormalities accurately. The liver's varying sizes and shapes cause incorrect segmentation due to the similarity in intensity between tumors and surrounding tissues and unclear lesion boundaries. These factors make manual annotation by radiologists time-intensive and error-prone, leading to inconsistencies in diagnosis.

Medical image segmentation achieves progress in improving cancer diagnosis accuracy and computer-aided diagnosis (CAD) systems assist in the detection, classification and segmentation of tumours on medical images, reduces radiologists' workload and increases diagnostic consistency [4–9]. Deep learning models, particularly Convolutional Neural Networks (CNNs), automate this segmentation process. Fully Convolutional Networks (FCNs) and U-Net architectures also segment liver tumors by performing pixel-wise classification [10]. Despite their high accuracy compared to manual segmentation, such models require significant computational resources, complicating deployment on

 doi:10.20944/preprints202503.0955.v1

edge devices with limited processing power and memory. Large servers are necessary to run these models, causing issues like bandwidth usage, data security, high server costs and substantial carbon footprints due to increased energy consumption compared to resource-constrained embedded systems. Thus, resource-efficient models are needed for accurate liver tumor segmentation on constrained devices.

Liver tumor segmentation employs traditional image processing, supervised, and unsupervised learning methods. Traditional techniques like thresholding [11], Canny edge detection [12], and watershed segmentation [13] differentiate tumors from normal tissue via edge detection and intensity thresholding. Thresholding uses intensity values to separate objects; watershed segmentation applies gradients to define boundaries. These methods struggle with medical image complexity: tumors have irregular shapes, varied sizes, and intensity values similar to surrounding tissues, causing errors [14]. Traditional approaches often require manual or semi-automatic intervention, increasing reliance on expert input for accuracy.

Unsupervised learning methods address limitations in traditional techniques by segmenting tumors without labeled data. Notable methods include clustering-based techniques and edge-based algorithms. For instance, Al-Kofahi et al. [15] introduced a multi-scale Laplacian of Gaussian (LoG) filter for histopathology images to detect nuclei of varying sizes. Kong et al. [16] developed a generalized LoG filter (gLoG) to detect elliptical nuclei in histopathology images, which can apply to LC segmentation in CT scans. Despite their potential, unsupervised methods require careful parameter tuning, are sensitive to noise and struggle to define tumor boundaries with low contrast [17]. Combining unsupervised techniques with region-based methods, such as Active Contour Models (ACM) [18] and marker-based watershed transforms [19], improves segmentation accuracy but remains computationally expensive and less generalizable across datasets.

CNNs, FCNs and U-Net variants dominate the field of medical image analysis due to their ability to learn hierarchical features from complex medical images. For instance, Saha Roy et al. [20] proposed an automated model that utilizes Mask R-CNN followed by Maximally Stable Extremal Regions (MSER) for tumor identification, enabling multi-class tumor classification. Chen et al. [21] proposed MS-FANet, a multi-scale feature attention network that performs liver tumor segmentation through multi-scale attention mechanisms which boost segmentation capabilities while capturing both global and local context. Lakshmi et al. [22] designed the Adaptive SegUnet++ (ASUnet++) framework and optimized it with the Enhanced Lemurs Optimizer (ELO) for tumor segmentation and classification. The authors' model tackles traditional machine learning hurdles including slow training times and gradient explosion issues as well as overfitting using both residual connections and multiscale approaches. Reyad et al. [23] proposed an architecture optimization framework for hybrid deep residual networks in liver tumor segmentation, utilizing a Genetic Algorithm (GA) to improve segmentation accuracy and model efficiency. Di et al. [24] developed a framework for automatic liver tumor segmentation which integrates 3D U-Net architecture with hierarchical superpixels and SVM-based classification, achieving robust performance on noisy and low-contrast CT images. Liu et al. [25] introduced PA-Net, a phase attention network that fuses venous and arterial phase features of CT images for liver tumor segmentation, effectively leveraging phase-specific information to enhance segmentation performance. CNN-based approaches have shown powerful representation abilities together with resilience to different image appearances. However, CNNs are inherently limited in modeling long-range dependencies, which can lead to suboptimal segmentation outcomes. Specifically, the localized receptive fields of convolutional operations restrict the network's focus to local context rather than global context [26].

Transformers which were initially created for sequence-to-sequence prediction tasks now play a primary role in computer vision tasks. Transformers demonstrate outstanding performance across multiple computer vision tasks including image classification [27], object detection [28], semantic segmentation [29] and generative tasks like text-to-image synthesis [25]. Transformers achieve success because their self-attention mechanism provides large receptive fields and long-range dependency

 doi:10.20944/preprints202503.0955.v1

capturing abilities. Medical image segmentation tasks have seen multiple proposals for hybrid methods that integrate both CNNs and Transformers. For instance, Balasubramanian et al. [30] proposed APESTNet, a Mask R-CNN-based Enhanced Swin Transformer Network for tumor segmentation and classification. This method combines the strengths of Mask R-CNN with the attention mechanisms of the Swin Transformer to improve segmentation accuracy. Chen et al. [31] introduced TransUNet, a cascaded architecture that integrates CNN and Transformer modules to enhance segmentation performance. Ni et al. [32] presented DA-Tran, a domain-adaptive transformer network for multiphase liver tumor segmentation. DA-Tran leverages domain adaptation techniques to effectively integrate multiphase CT images, improving segmentation accuracy and robustness across varying imaging conditions.

Despite their accuracy, existing models require substantial memory and computational power, which are unsuitable for edge devices like Jetson Nano. These models operate on server systems, requiring sensitive patient data transmission over the internet and exposing data to privacy and security risks. High server energy consumption limits feasibility in resource-constrained environments. This problem is now actively discussed in various domains: recent work in predictive model optimization demonstrates how tailored loss functions that balance accuracy and hardware constraints enable edge deployment [33], while hybrid training frameworks that integrate domain-specific priors show improved convergence efficiency [34**?**]. Inspired by these advances, there is a need to optimize segmentation models to reduce their size and power use while preserving accuracy, enabling deployment on edge devices for real-time, secure, and energy-efficient tumor segmentation. This study proposes a novel approach to optimize the Swin-UNet model for efficient liver cancer segmentation on edge devices, balancing model size and Area Under the Curve (AUC). Contributions include:

- **Model Size Optimization:** The discrete design space of Swin-UNet achieves a balance between model size and accuracy, enabling deployment on memory-constrained devices like Jetson Nano.
- **Quadratic Penalty Objective Function:** A quadratic penalty-based objective function balances model size and AUC, encouraging compact, accurate models.
- **Search and Rescue Algorithm:** The Search and Rescue algorithm identifies optimal configurations, yielding an optimized model termed SAR-Swin-UNet.
- **Focal AUC Loss Function:** The Focal AUC loss function addresses class imbalance during training, enhancing the model's ability to segment minority class pixels.

This approach facilitates accurate tumor segmentation on edge devices, ensuring real-time analysis with data security and energy efficiency.

2. Preliminary Knowledge

This section covers key computer vision architectures: Transformers for feature extraction and U-Net for segmentation. Transformers excel at feature extraction, particularly in image classification; U-Net optimizes image segmentation. The section details both architectures with key equations and variable explanations.

2.1. The Transformer Model

The Transformer architecture, introduced by Vaswani et al. [35], revolutionizes sequence-based models by replacing recurrent mechanisms with self-attention. This enables parallel input processing and improves computational efficiency. The core operation is scaled dot-product attention, defined as follows,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

 doi:10.20944/preprints202503.0955.v1

where, Q, K and V are matrices representing the query, key and value vectors, respectively. The dimensions of these matrices are as follows:

$$Q, K \in \mathbb{R}^{n \times d_k}, \quad V \in \mathbb{R}^{n \times d_v}$$

where n is the sequence length (number of tokens), d_k is the dimensionality of the query and key vectors and d_v is the dimensionality of the value vectors. The attention mechanism computes a sum of the value vectors, where the weights are determined by the similarity between the query and key vectors. This self-attention mechanism is applied to model dependencies across the entire sequence. In the context of image-based tasks, this model is typically applied after embedding image patches into tokens, as in the Vision Transformer (ViT), which we discuss in the next subsection.

2.2. The Vision Transformer (ViT)

The Vision Transformer (ViT) [36], adapts the Transformer architecture to image classification tasks by treating image patches as tokens. Given an input image $X \in \mathbb{R}^{H \times W \times C}$, where H is the height, W is the width and C is the number of channels, the image is divided into non-overlapping patches of size $P \times P$. These patches are then flattened and projected into an embedding space. The projection of patch x_i can be formulated as:

$$z_0 = [x_1, x_2, \ldots, x_N]W_e + b_e \tag{2}$$

where $W_e \in \mathbb{R}^{(P^2C) \times d}$ is the embedding matrix, b_e is the embedding bias and $N = \frac{H}{P} \times \frac{W}{P}$ is the number of patches. Each embedded patch is treated as a token and passed through the Transformer layers. The model utilizes multi-head self-attention and feed-forward networks, as described previously. However, ViT relies heavily on large datasets for training and performs well when pre-trained on large-scale data and fine-tuned on specific tasks. The advantage of ViT lies in its ability to capture global dependencies across image patches, which is a limitation in conventional CNNs that rely on localized receptive fields. However, ViT requires significant computational resources and large datasets to perform optimally.

2.3. The Swin Transformer

The Swin Transformer [37] presents a modification to the traditional Vision Transformer (ViT) architecture by addressing the challenges associated with computational complexity when working with high-resolution images. Unlike ViT's global self-attention across all image patches, Swin Transformer employs local window-based attention. This reduces computational costs, enhancing scalability for large images. It also uses a hierarchical structure that progressively enlarges the receptive field, enabling capture of local and global features across scales.

Swin Transformer applies attention within non-overlapping local windows. The model computes self-attention within each window independently and aggregates outputs. Its key innovation—a shifting window mechanism applied to successive transformer layers—captures long-range dependencies between neighboring regions, overcoming limitations of strictly local attention. The shifted-window attention mechanism is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \Delta\right)V \tag{3}$$

where, Q, K and V are the query, key and value matrices, respectively and d_k is the dimensionality of the key vectors. The term Δ represents the shift between windows that occurs at successive layers of the model. This shift is critical for allowing information to propagate between neighboring windows, ensuring that global dependencies can still be captured, despite the local nature of the attention in the first few layers. The introduction of the shift operation helps the Swin Transformer overcome the limitations of purely local attention, providing the benefits of both local and global feature learning.

 doi:10.20944/preprints202503.0955.v1

To clarify the role of each variable in the equation: - $Q \in \mathbb{R}^{n \times d_k}$: The query matrix, where n is the number of patches in the window and d_k is the dimensionality of the query vector. - $K \in \mathbb{R}^{n \times d_k}$: The key matrix, which has the same shape as the query matrix. - $V \in \mathbb{R}^{n \times d_v}$: The value matrix, used to compute the sum of the values based on the attention scores. - Δ: The shift term introduced between successive layers to enable the exchange of information between neighboring windows.

Unlike the original Transformer where attention is computed globally across the entire image, the Swin Transformer confines attention within local windows at the early layers of the model, dramatically reducing computational complexity. The local-to-global attention mechanism is achieved by shifting the window between layers, gradually increasing the receptive field and allowing the model to capture long-range dependencies. This hierarchical design makes the Swin Transformer particularly well-suited for vision tasks such as image classification and object detection, where both fine-grained details and global context are important.

In contrast to the Vision Transformer, where the attention mechanism is applied globally across the entire input sequence (i.e., all patches in the image), the Swin Transformer introduces a more efficient mechanism by limiting attention to small, local windows. However, the shift in windows at each layer ensures that information is shared across windows, thus enabling the model to learn long-range dependencies. The computational complexity of the self-attention operation in the Swin Transformer is reduced to $O(N \cdot W^2)$, where N is the number of patches and W is the window size, as opposed to the $O(N^2)$ complexity in ViT, which computes attention globally for all image patches. This reduction in complexity allows Swin Transformer to scale to much larger images without sacrificing performance.

ViT and Swin Transformer differ in attention handling, ViT applies global attention, incurring high computational costs, while Swin restricts attention to local regions and uses shifting windows to propagate information. This strategy improves efficiency and scalability for high-resolution images. Swin Transformer thus combines local window-based attention with a hierarchical structure, reducing computational complexity while capturing local and global dependencies, improving vision task efficiency.

2.4. U-Net for Segmentation

U-Net, introduced by Ronneberger et al. [38], is a convolutional neural network designed for semantic segmentation, notably in medical imaging. Its encoder-decoder structure uses skip connections to recover spatial information lost during encoder downsampling. The encoder comprises convolutional layers that progressively reduce input resolution, extracting abstract features. At layer l,

$$f_l = \sigma(W_l * f_{l-1} + b_l) \tag{4}$$

where W_l denotes the convolutional filter at layer l, $*$ is the convolution operation, b_l is the bias term, and σ is the activation function (typically ReLU). This process reduces spatial resolution and increases feature map depth, enabling capture of higher-level abstract features representing input content. The decoder reverses the encoder's downsampling using upsampling layers, often transposed convolutions, to restore feature map resolution. The upsampling operation at decoder layer l is:

$$f'_l = \sigma(W'_l * f'_{l-1} + b'_l) \tag{5}$$

where W'_l denotes transposed convolution filters and f'_l is the upsampled feature map. Skip connections between encoder and decoder layers transfer fine-grained spatial information from encoder to decoder, preserving precise localization of segmentation boundaries. This U-Net design ensures accurate and spatially precise segmentation masks. The decoder's feature map passes through an activation function: softmax for multi-class segmentation or sigmoid for binary segmentation. The network computes the final segmentation mask as:

 doi:10.20944/preprints202503.0955.v1

$$\hat{Y} = \text{softmax}(W_{\text{final}} * f_{\text{final}} + b_{\text{final}}) \tag{6}$$

In this equation, W_{final} and b_{final} are the final weights and biases and f_{final} is the last feature map produced by the decoder. The architecture of U-Net, with its careful design of encoder-decoder structures and skip connections, is particularly well-suited for applications requiring precise, pixel-level segmentation, such as in medical imaging tasks.

U-Net addresses a distinct task compared to Transformer-based models like Vision Transformer (ViT) or Swin Transformer. Transformers focus on feature extraction, particularly image classification, where they extract semantic features and classify images using global contexts. While effective for classification, their self-attention mechanisms and reliance on large training datasets increase computational costs. They also struggle with pixel-level precision tasks like segmentation. U-Net's encoder-decoder structure with skip connections captures high-level features and spatial details, enabling effective medical image segmentation where pixel accuracy matters. The architecture preserves spatial information lost in deeper networks, making it ideal for such tasks. Hence, Transformers suit tasks needing global context, like classification, U-Net excels in segmentation requiring precise localization for accurate predictions. Whereas, the overall Swin-Unet model outperforms other architectures in both aspects.

3. Proposed Methodology

This section details the methodology for optimizing Swin-UNet for liver cancer segmentation on memory-constrained edge devices. The approach combines four components ensuring high accuracy under edge device constraints.

The methodology integrates Swin-UNet, merging Swin Transformer and U-Net for medical image segmentation. Swin Transformer captures local and global features critical for segmentation accuracy. The Search and Rescue (SAR) algorithm adjusts hyperparameters to minimize an objective function balancing accuracy and model size. The objective function optimizes the Area Under the Curve (AUC) for classification performance and model size for edge deployment. AUC focal loss prioritizes hard examples (e.g., complex lesions), improving segmentation robustness. Mathematical formulations underpin the optimization process, justifying its effectiveness. Figure 1 illustrates the workflow and component interactions.

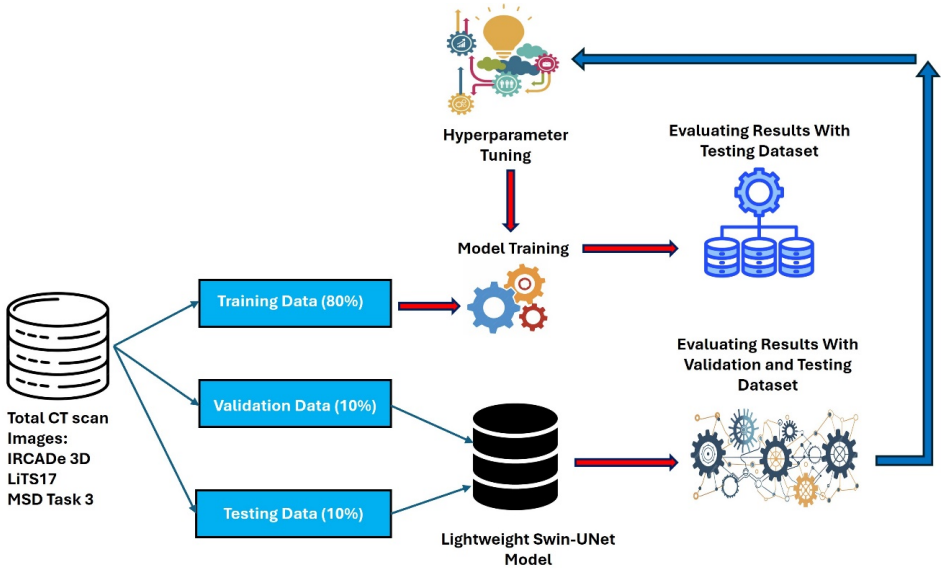


Figure 1. Block diagram of the proposed methodology for optimizing Swin-UNet for liver cancer segmentation.

 doi:10.20944/preprints202503.0955.v1

The training dataset splits into training, validation, and test sets. The SAR algorithm iteratively reduces the objective function, enhancing accuracy while constraining model size. Post-optimization, the final model trains and computes results. Subsequent subsections elaborate on each component.

3.1. Swin-UNet and Hyperparameter Optimization

The Swin-UNet model integrates the hierarchical structure and shifted window attention mechanism of the Swin Transformer into the encoder-decoder framework of UNet as shown in the Figure 2. This subsection overviews the Swin-UNet architecture and details the hyperparameters that significantly impact model performance and memory efficiency.

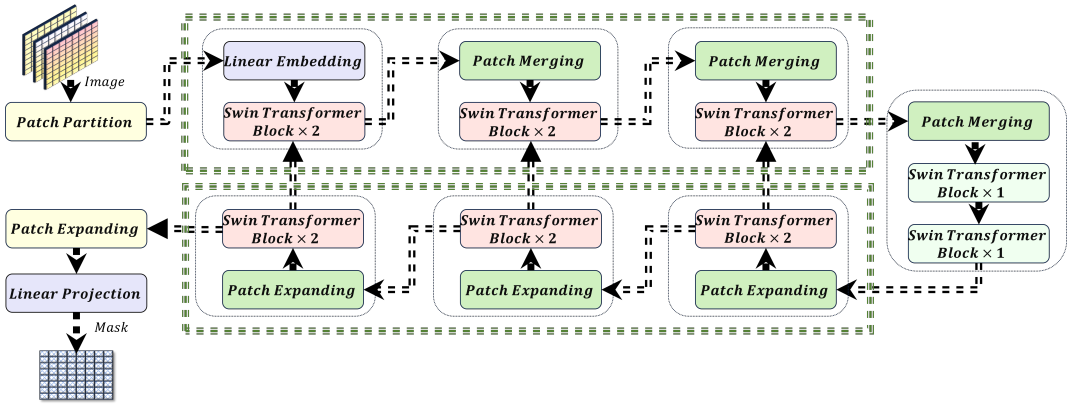


Figure 2. Block Diagram of Swin-Unet Architecture.

3.1.1. Multi-Head Self-Attention (MHSA) in Swin-UNet

Swin-UNet employs a localized attention mechanism within non-overlapping windows, which differs from the global self-attention used in traditional Vision Transformers (ViTs). The attention mechanism for each window is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

where, Q, K and V are the query, key and value matrices, respectively and d_k is the dimensionality of the keys. This localized approach reduces the computational complexity compared to global attention mechanisms, making it more suitable for deployment on edge devices.

To facilitate spatial information exchange between windows, the Swin Transformer employs a shifted windowing mechanism, mathematically expressed as:

$$\text{Shifted Window}_i = \text{Window}_i + \Delta \tag{8}$$

where, Δ represents the offset applied to window positions. This mechanism allows the model to capture inter-window dependencies, enhancing the overall segmentation accuracy of liver tumors.

3.1.2. Hyperparameter Space and Constraints

The efficiency and performance of the Swin-UNet model are influenced by several key hyperparameters, each subject to specific constraints. Table 1 summarizes these hyperparameters along with their optimization ranges:

 doi:10.20944/preprints202503.0955.v1

Table 1. Hyperparameter constraints for the Swin-UNet model and AUC focal loss.

Hyperparameter	Symbol	Range
Depth	D	[2, 8]
Initial Filter Number	F_0	[32, 256]
Patch Size	P	[2, 16]
Number of Attention Heads	H	[1, 16]
Window Size	W	[1, 8]
MLP Size	M	[32, 512]
AUC Focal Loss Alpha	α	[0, 5]
AUC Focal Loss Gamma	γ	[0, 5]

These hyperparameters control various aspects of the model, such as its depth, capacity and the size of the input data it processes. Optimizing these parameters within their specified ranges ensures a balance between the model's segmentation accuracy and its memory footprint.

3.2. Search and Rescue (SAR) Algorithm

The Search and Rescue (SAR) algorithm [39] solves optimization problems by balancing global exploration and local refinement. Inspired by rescue missions, SAR explores high-dimensional search spaces efficiently while focusing on promising regions, avoiding local minima. In machine learning, hyperparameter selection critically affects model performance. Algorithms like neural networks, decision trees, and support vector machines depend on hyperparameters (e.g., learning rate, depth, regularization) for accuracy, training time, and generalization. SAR navigates large hyperparameter spaces efficiently, exploiting promising configurations while balancing performance and computational cost.

This study applies SAR to hyperparameter optimization in medical image segmentation for the first time. SAR tunes Swin-UNet for liver tumor segmentation, targeting high accuracy and computational efficiency. Figure 3 illustrates the SAR-based optimization workflow.

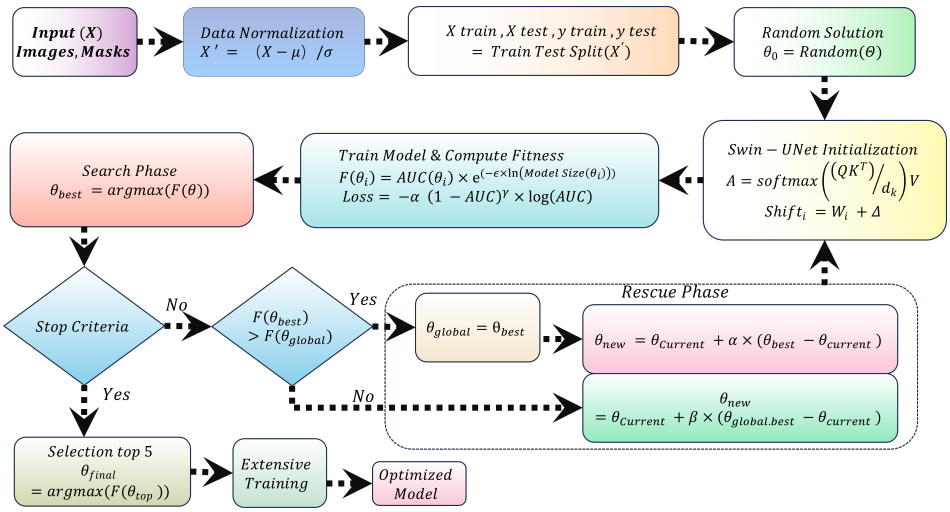


Figure 3. Block diagram of SAR-based hyperparameter optimization and training with AUC focal loss.

3.2.1. SAR Optimization Phases

SAR operates in three key phases: Initialization, Search and Rescue. These phases integrate both global search strategies and local refinement to ensure the algorithm explores the parameter space widely and then fine-tunes the promising regions. The following sections provide a detailed breakdown of each phase, followed by the mathematical formulations that guide the optimization process.

Initialization Phase:

 doi:10.20944/preprints202503.0955.v1

The algorithm starts by generating an initial population of candidate solutions θ_i, where each solution represents a potential set of hyperparameters. These candidate solutions are randomly sampled within predefined bounds, which are set based on prior knowledge or expert intuition about the parameter space. The initialization phase plays a critical role in setting the starting point of the optimization and the quality of the initial candidates can significantly impact the subsequent search process.

The fitness of each candidate solution is evaluated using a fitness function that considers two key aspects: model performance (measured by AUC) and model size (which affects computational efficiency). The fitness function is formulated as follows:

$$\text{Fitness}(\theta_i) = \text{AUC}(\theta_i) \times \exp(-\epsilon \times \log(\text{Model Size}(\theta_i))) \tag{9}$$

where θ_i represents a set of hyperparameters for candidate solution i, ϵ is a scaling factor that controls the influence of model size and AUC denotes the area under the receiver operating characteristic (ROC) curve. This formulation ensures that the fitness function rewards configurations that achieve high accuracy while keeping the model size manageable.

Search Phase:

Once the initial population is generated and evaluated, the SAR algorithm enters the Search phase. During this phase, each candidate solution is assessed based on its fitness score and the algorithm iteratively updates the positions of the solutions. The goal is to explore the search space and identify regions that are most promising, based on the fitness function. The search process is carried out using the following formula for updating the positions of each candidate:

$$\theta_i^{\text{new}} = \theta_i^{\text{current}} + \alpha \times (\theta_{\text{best neighbor}} - \theta_i^{\text{current}}) \tag{10}$$

where α is a learning rate parameter and $\theta_{\text{best neighbor}}$ refers to the solution with the highest fitness score among the neighboring solutions. By updating the position of each candidate solution iteratively, SAR ensures that the search is directed towards regions of the search space that hold the potential for better performance.

Rescue Phase:

After the Search phase, the algorithm proceeds to the Rescue phase, where further refinement of the solutions is performed. During this phase, the candidate solutions are adjusted based on their proximity to the best solutions found during the Search phase. The goal of the Rescue phase is to focus the search on the most promising regions, fine-tuning the hyperparameters in order to improve model performance. To adjust the candidate solutions, the algorithm uses two strategies. First, if a suitable neighboring solution is found, the candidate solution is updated by moving it toward the best-performing neighbor. The adjustment is calculated as given in equation 10. If no suitable neighbors are found, the candidate solution is adjusted towards the globally best solution identified during the Search phase. The update for this adjustment is given by:

$$\theta_i^{\text{new}} = \theta_i^{\text{current}} + \beta \times (\theta_{\text{global best}} - \theta_i^{\text{current}}) \tag{11}$$

where α and β are scaling factors that control the magnitude of the adjustments. These strategies ensure that the algorithm refines the solutions towards the most optimal configurations while maintaining a balance between exploration and exploitation. The iterative process of the Rescue phase continues until the algorithm converges, yielding an optimal or near-optimal set of hyperparameters that balances both model performance (AUC) and model size, ensuring efficient training and accurate predictions.

3.3. AUC Focal Loss for Class Imbalance

Liver tumor segmentation faces class imbalance between tumor and healthy pixels. This work employs AUC focal loss to prioritize minority class learning by modulating easy and hard example contributions. The loss function focuses on difficult tumor pixels:

 doi:10.20944/preprints202503.0955.v1

$$\text{Loss} = -\alpha(1 - \text{AUC})^{\gamma} \log(\text{AUC}) \tag{12}$$

Here, α weights the minority class and γ adjusts focus on hard examples. This work optimizes $\alpha \in [0, 5]$ and $\gamma \in [0, 5]$ to handle imbalance effectively. Combining SAR-based hyperparameter optimization with AUC focal loss enables efficient Swin-UNet models for accurate segmentation on edge devices. Algorithm 1 provides implementation details.

Algorithm 1 Search and Rescue (SAR) Algorithm for Swin-UNet Optimization

Initialize Swin-UNet Architecture
  Input: Hyperparameters - Depth (D), Initial Filter Number (F_0), Patch Size (P),
  Number of Attention Heads (H), Window Size (W), MLP Size (M)
  Output: Swin-UNet Model
Define the SAR Algorithm:
  Input: Hyperparameter ranges
  Output: Optimized hyperparameters
Initialization Phase:
  Generate initial population of candidate solutions $\{\theta_i\}$
  For each candidate θ_i, randomly sample hyperparameters within the predefined ranges:
    $D \in [2, 8], F_0 \in [32, 256], P \in [2, 16], H \in [1, 16], W \in [1, 8], M \in [32, 512]$
Search Phase:
  For each candidate θ_i, calculate Fitness using the following formula:
    $\text{Fitness}(\theta_i) = \text{AUC}(\theta_i) \times \exp(-\epsilon \times \log(\text{Model Size}(\theta_i)))$
  Select candidates with highest fitness for further exploration
Rescue Phase:
  For each selected candidate θ_i:
    If a suitable neighbor $\theta_{\text{best neighbor}}$ is found:
      Adjust θ_i towards neighbor:
        $\theta_i^{\text{new}} = \theta_i^{\text{current}} + \alpha \times (\theta_{\text{best neighbor}} - \theta_i^{\text{current}})$
    Else, adjust θ_i towards global best $\theta_{\text{global best}}$:
      $\theta_i^{\text{new}} = \theta_i^{\text{current}} + \beta \times (\theta_{\text{global best}} - \theta_i^{\text{current}})$
AUC Focal Loss:
  Define the AUC Focal Loss function:
  $\mathcal{L}_{\text{AUC}} = -\alpha \times (1 - \text{AUC})^{\gamma} \times \log(\text{AUC})$
Train Swin-UNet:
  Train the Swin-UNet model using the optimized hyperparameters and AUC focal loss.
Output: Final optimized Swin-UNet model

3.4. Optimality Analysis of the Objective Function with SAR

The objective function used in this study is given by equation 9. Where θ_i is a set of hyperparameters that includes both discrete and continuous variables. This function is designed to balance model accuracy (AUC) with model size, where ϵ is a positive constant that controls the trade-off. Given that θ_i includes a combination of discrete and continuous variables, the optimization process operates in a mixed space. Thus, traditional convexity concepts apply only to the continuous subspace.

Convexity in the Continuous Subspace:

For a fixed discrete setting of θ_i, consider the continuous component $\theta_i^{(c)}$. The objective function within this continuous subspace can be simplified as:

$$f(\theta_i^{(c)}) = x \times y^{-\epsilon}, \tag{13}$$

where $x = \text{AUC}$ and $y = \text{Model Size}$.

The convexity analysis in this continuous domain reveals that the Hessian matrix H of this function is given by:

$$H = \begin{bmatrix} 0 & -\epsilon y^{-\epsilon-1} \\ -\epsilon y^{-\epsilon-1} & \epsilon x(\epsilon+1) y^{-\epsilon-2} \end{bmatrix}. \tag{14}$$

 doi:10.20944/preprints202503.0955.v1

The determinant of this Hessian matrix is:

$$\text{Det}(H) = \epsilon^2 y^{-2\epsilon-2}. \tag{15}$$

Since $\epsilon > 0$ and $y > 0$, $\text{Det}(H) \geq 0$. However, since $\frac{\partial^2 f}{\partial x^2} = 0$, the function is not strictly convex but exhibits convexity in a weaker sense. This suggests that the function has flat regions in the continuous domain, leading to multiple suboptimal solutions rather than a unique global optimum.

Optimality in the Discrete Space:

In the context of discrete variables, traditional convexity does not directly apply. However, the concept of piecewise convexity and discrete optimization can be leveraged. For the discrete components of θ_i, the objective function can be viewed as a set of piecewise convex functions:

$$\text{Fitness}(\theta_i^{(d)}, \theta_i^{(c)}) = \left\{ f_j(\theta_i^{(c)}) \mid \theta_i^{(d)} = j \right\}, \tag{16}$$

where each $f_j(\theta_i^{(c)})$ represents the objective function in the continuous subspace for a fixed discrete setting j. Each function $f_j(\theta_i^{(c)})$ is convex, as previously proven.

Although the discrete space lacks a gradient, the SAR algorithm explores it by evaluating a finite set of configurations. SAR employs a combination of local search and global adjustment strategies to navigate this space, effectively finding a configuration that minimizes the objective function.

Suboptimality and Convergence with SAR:

SAR ensures that the search process converges to a suboptimal solution in the mixed space. Given that the continuous subspace is convex for fixed discrete settings, SAR optimizes locally in these regions as given by equation 10. In cases where discrete changes are necessary, SAR adjusts the discrete variables, re-evaluating the objective function. This iterative process guarantees convergence to a suboptimal or near-optimal solution due to the following:

- **Local Optimality in Continuous Subspace** Each continuous optimization step finds a locally optimal solution within the convex region.
- **Global Exploration in Discrete Space** By systematically exploring different discrete configurations, SAR ensures broad coverage of the search space.

Thus, while strict convexity in the discrete space is not mathematically provable, the combination of convexity in the continuous space and SAR's exploration mechanism ensures effective navigation towards at least a suboptimal point.

4. Results and Discussion

This section discusses the results and comparisons for the proposed scheme. It begins with an overview of the datasets and comparison metrics used to evaluate the scheme. Following this, the optimal hyperparameters are presented, along with an analysis of the model's performance on various datasets. Finally, the section compares the proposed scheme with recently introduced schemes that have used the same datasets.

4.1. Datasets and Preprocessing

This study uses three datasets for liver tumor segmentation: the LiTS17 dataset, the IRCADe 3D dataset and the MSD Task 3 dataset. The LiTS17 dataset contains 131 images with labels for liver tumor segmentation. These images provide quality data for training and evaluating models. The IRCADe 3D dataset includes 20 images of patients with liver tumors. These images offer layers of CT scans that show liver and nearby structures. Differences in slice thickness and tumor features make this dataset useful for testing models under practical conditions. The MSD Task 3 dataset has 130 images often used for segmentation tasks in medical imaging. These images show a range of liver tumor cases, helping evaluate models in many scenarios. The datasets split into training, validation and test sets in 80%, 10% and 10% proportions, as shown in Table 2.

 doi:10.20944/preprints202503.0955.v1

Table 2. Dataset Distribution for Liver Tumor Segmentation.

Dataset	Total Images	Training (80%)	Validation (10%)	Test (10%)
LiTS17[40]	131	104	13	13
IRCADe 3D [41]	20	16	4	4
MSD Task-3 [42]	131	104	13	13
Total	**282**	**224**	**30**	**30**

Preprocessing helps expand the training dataset and improve model performance. Transformations like rotation, flipping, scaling, gamma correction and logarithmic scaling apply to training images while keeping segmentation masks aligned. These transformations create new versions of images to make models learn from more examples.

Rotation randomly changes image angles within $\pm 15^{\circ}$. The transformation for rotation is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{17}$$

where (x, y) are original coordinates and (x', y') are rotated coordinates. This transformation applies to both images and masks to keep them aligned.

Flipping creates variations by horizontally or vertically flipping images. A horizontal flip uses:

$$(x', y') = (-x, y) \tag{18}$$

A vertical flip uses:

$$(x', y') = (x, -y) \tag{19}$$

Scaling simulates zooming by randomly resizing images by $\pm 10\%$. Scaling uses the formula:

$$(x', y') = (s \cdot x, s \cdot y) \tag{20}$$

where s is the scale factor. Gamma correction adjusts brightness and contrast to help models detect regions under different lighting. Gamma correction follows:

$$I' = c \cdot I^{\gamma} \tag{21}$$

where I is the intensity and c and γ are constants. Logarithmic scaling improves visibility in areas with low contrast. This uses:

$$I' = \alpha \cdot \log(1 + I) \tag{22}$$

where I is the intensity and α is a constant. Both gamma and logarithmic scaling enhance features but leave masks unchanged. These steps increase the variety of the data sets and help the models generalize better to new data.

4.2. Experimental Setup

This work optimizes the Swin-UNet model for liver tumor segmentation to improve accuracy and computational efficiency. The Search and Rescue algorithm tunes hyperparameters using an Nvidia 3090 Ti GPU, accelerating the search process. This work deploys the optimized model on a Jetson Nano to evaluate edge-device performance under memory constraints. The Jetson Nano provides a practical balance for real-time medical image segmentation. All models use a learning rate of 0.0001, empirically shown to balance convergence speed and stability. Training runs for 2000 epochs with early stopping (patience: 10 epochs) to prevent overfitting. A batch size of 64 balances memory usage and convergence. The Adam optimizer enhances training through adaptive learning

 doi:10.20944/preprints202503.0955.v1

rates and momentum. AUC focal loss addresses class imbalance by prioritizing challenging tumor regions. Experiments demonstrate that the optimized model achieves high segmentation accuracy while maintaining computational efficiency on edge devices like the Jetson Nano.

Table 3. Summary of Model Hyperparameters and Setup.

Hyperparameter	Value
Learning Rate	0.0001
Epochs	2000
Batch Size	64
Optimizer	Adam
Patience (Early Stopping)	10 epochs
Device for Optimization	Nvidia 3090 Ti
Device for Deployment	Jetson Nano

4.3. Comparison Metrics

This section introduces metrics to evaluate segmentation algorithms, assessing model performance in distinguishing positive and negative cases and spatial accuracy against ground truth. Metrics fall into two main categories, classification and overlap-based matrics. Classification metrics evaluate model classification performance. Precision represents the proportion of true positives among positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{23}$$

Recall indicates the proportion of true positives identified:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{24}$$

The F1 score combines precision and recall through harmonic mean:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{25}$$

Accuracy measures overall correctness:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

Specificity quantifies true negative identification:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{27}$$

Overlap-based metrics assess spatial accuracy between predicted (P) and ground truth (G) segmentations. The Dice Similarity Coefficient measures overlap:

$$\text{Dice} = \frac{2|P \cap G|}{|P| + |G|} \tag{28}$$

where $|P|$ and $|G|$ denote pixel counts in predicted and ground truth regions, and $|P \cap G|$ counts overlapping pixels. Volume Overlap Error (VOE) computes union-based error:

$$\text{VOE} = \frac{|P \cup G| - |P \cap G|}{|P \cup G|} \tag{29}$$

Relative Volume Difference (RVD) quantifies volume discrepancy:

 doi:10.20944/preprints202503.0955.v1

$$\text{RVD} = \frac{|V_{\text{seg}} - V_{\text{gt}}|}{V_{\text{gt}}} \times 100 \tag{30}$$

where V_{seg} and V_{gt} represent segmented and ground truth volumes. Lower VOE and RVD indicate better spatial and volumetric accuracy.

4.4. Hyperparameter Optimization and Model Performance

This work evaluates hyperparameter settings for optimizing the Swin-UNet model using the IRCADe 3D dataset. Four experiments analyze the trade-offs between model complexity, segmentation performance, and size: one unoptimized baseline and three configurations with distinct ϵ values (0.0486, 0.1172, 0.2344). Table 4 summarizes the parameters.

Table 4. Hyperparameter Settings for Swin-UNet Model with Different ϵ Values.

Hyperparameter	**Unoptimized**	$\epsilon = 0.2344$	$\epsilon = 0.1172$	$\epsilon = 0.0486$
Filter Number Begin	128	32	32	64
Depth	4	4	4	4
Stack Num Down	2	2	2	2
Stack Num Up	2	2	2	2
Patch Size	4	16	16	16
Number of Heads	4, 8, 8, 8	4, 2, 8, 2	4, 1, 4, 2	8, 4, 2, 4
Window Size	4, 2, 2, 2	1, 1, 2, 2	8, 1, 2, 2	8, 1, 4, 1
Num MLP	512	46	158	46
Gamma	2	2.6326	1.7471	1.4890
Alpha	0.5	4.9448	4.9407	3.7244

The unoptimized model uses 128 filters and patch size 4. Optimized configurations adjust these parameters: $\epsilon = 0.0486$ minimizes model size through reduced filters and simpler attention mechanisms; $\epsilon = 0.1172$ balances complexity and performance; $\epsilon = 0.2344$ maximizes AUC via larger filters and complex attention. This work enables selection of ϵ based on deployment requirements for memory efficiency versus segmentation accuracy.

4.5. Model Performance on Liver Tumor Segmentation

This work assesses segmentation performance across 3DIRCADb, LiTS, and MSD Task03 datasets using accuracy, precision, recall, specificity, Dice score, VOE, and RVD metrics. It identifies optimal ϵ values for each dataset by analyzing model size-performance trade-offs. Table 5 summarizes results, highlighting configurations that balance compression and accuracy for specific clinical applications.

Table 5. Performance Metrics Across Configurations and Datasets.

Dataset	ϵ	**Size (MB)**	**Acc.**	**Prec.**	**Rec.**	**Spec.**	**Dice**	**VOE**	**RVD(%)**
3DIRCADb	unoptimized	324.91	0.9985	0.8272	0.8488	0.9999	0.8340	0.1801	0.30
3DIRCADb	0.2344	64.16	0.9998	0.9297	0.9915	1.0000	0.9478	0.0783	0.23
3DIRCADb	0.1172	30.88	0.9976	0.7423	0.8401	0.9994	0.7891	0.2661	0.89
3DIRCADb	0.0486	17.22	0.9962	0.7400	0.8329	0.9962	0.7644	0.2634	4.82
LiTS	unoptimized	324.91	0.9998	0.9797	0.9709	1.0000	0.8753	0.0291	2.89
LiTS	0.2344	64.16	0.9999	0.9923	0.9910	1.0000	0.8906	0.0166	2.51
LiTS	0.1172	30.88	0.9996	0.9288	0.9817	0.9998	0.8405	0.0791	3.17
LiTS	0.0486	17.22	0.9987	0.8724	0.9925	0.9988	0.7998	0.1345	16.8
MSD Task03	unoptimized	324.91	0.9999	0.9712	0.9921	0.9999	0.8758	0.0366	19.33
MSD Task03	0.2344	64.16	0.9999	0.9906	0.9920	1.0000	0.8895	0.0173	4.63
MSD Task03	0.1172	30.88	0.9999	0.9569	0.9921	0.9999	0.8654	0.0508	27.39
MSD Task03	0.0486	17.22	0.9964	0.8124	0.9933	0.9965	0.8363	0.1938	48.04

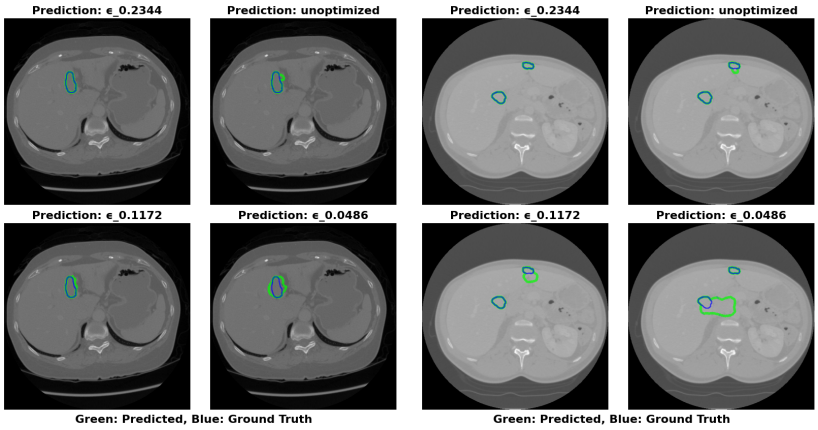
 doi:10.20944/preprints202503.0955.v1

4.5.1. 3DIRCADb Dataset Analysis

The unoptimized model on the 3DIRCADb dataset achieves a high Dice score of 83.40% and an RVD of 0.23%. Optimizing the model with ϵ=0.2344 reduces the model size by 80.25% to 64.16 MB, with a slight increase in Dice to 94.78% and a marginal reduction in RVD to 0.23%. This shows that a smaller model size can be maintained without a significant degradation in performance, even with substantial model compression. Further reducing ϵ to 0.0486 yields a smaller model size of 17.22 MB but leads to a marked drop in Dice to 76.44%, despite achieving a recall of 83.29%. This highlights the trade-off between model size and segmentation performance, where a smaller model may sacrifice precision in favor of capturing a broader range of true positives, as reflected in the inflated RVD.

In terms of liver tumor segmentation, the model's recall increases as ϵ decreases, suggesting better sensitivity to true positives. However, this comes at the cost of a reduced Dice score, indicating poorer localization of tumor boundaries. These observations align with the concept that recall improvement does not necessarily guarantee better overall model performance, as it may increase false positives and degrade precision, leading to higher volume estimation errors (RVD).

The visual segmentation results presented in Figure 4 support these findings. The configuration with ϵ=0.2344 exhibits the most accurate tumor boundary predictions, with minimal deviation from the true boundaries. In contrast, the unoptimized model, though effective, demonstrates less precise boundary delineation, particularly in complex tumor regions. These visual results confirm the quantitative findings, reinforcing that optimizing ϵ significantly enhances segmentation accuracy and computational efficiency.



(a) (b)

Figure 4. Visual Results of Segmentation with Different ϵ Values on IRCADe 3D Dataset

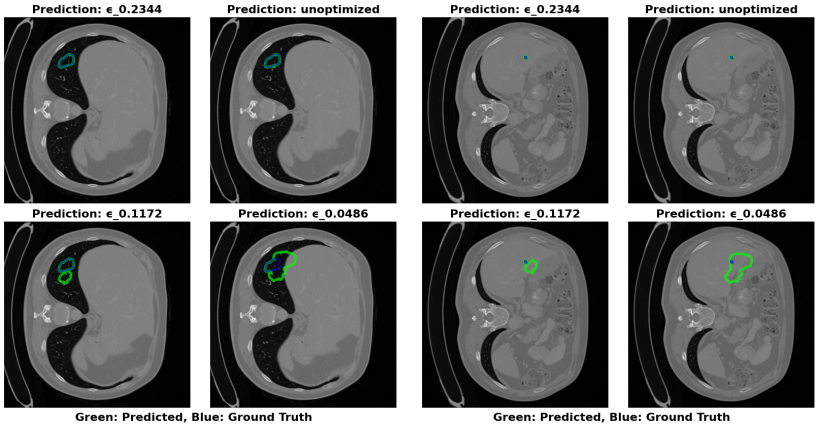
4.5.2. LiTS Dataset Analysis

For the LiTS dataset, the unoptimized model achieves a high Dice score of 87.53% and an RVD of 2.89%. After optimizing ϵ to 0.2344, the model size is reduced to 64.16 MB with a slight decrease in Dice to 89.06% and an improvement in RVD to 2.51%. This demonstrates that optimizing the model with a smaller ϵ leads to a good balance between performance and model size, with minimal sacrifice in segmentation accuracy. However, further reduction of ϵ to 0.0486 significantly reduces the model size to 17.22 MB but causes a large drop in Dice to 79.98% and a marked increase in RVD to 16.8%, indicating that excessive model compression leads to a significant performance trade-off.

As with the 3DIRCADb dataset, reducing ϵ enhances recall but at the expense of precision, resulting in larger errors in volume estimation. The model becomes more sensitive to liver tumor pixels, but this increased sensitivity also leads to a greater number of false positives. Figure 5 shows that the optimal configuration at ϵ=0.2344 offers the most accurate segmentation boundaries, with slight deviations from the true tumor boundaries. The lower ϵ configuration (0.0486) results in

 doi:10.20944/preprints202503.0955.v1

larger inaccuracies in boundary delineation, supporting the need for careful selection of ϵ to balance performance and model size.



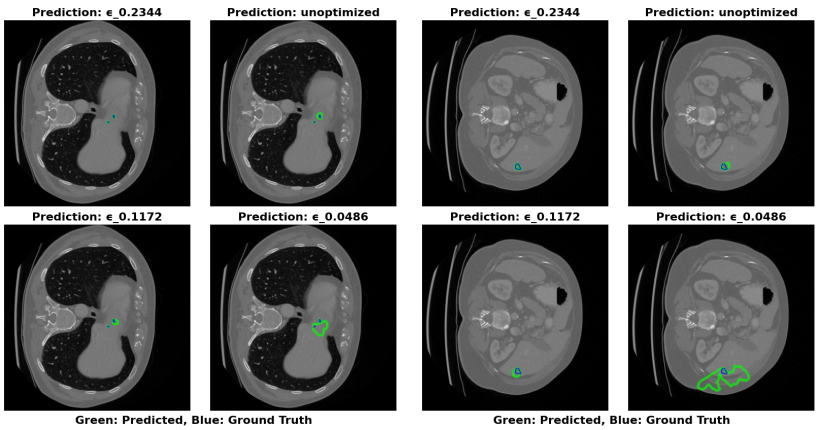
(a) (b)

Figure 5. Visual Results of Segmentation with Different ϵ Values on LiTS17 Dataset

4.5.3. MSD Task03 Dataset Analysis

On the MSD Task03 dataset, the unoptimized model achieves a Dice score of 87.58% and an RVD of 19.33%. After optimizing ϵ to 0.2344, the model size is reduced to 64.16 MB, with an improvement in Dice to 88.95% and a significant reduction in RVD to 4.63%. This indicates that ϵ=0.2344 strikes the optimal balance between model size and performance, offering enhanced tumor segmentation while maintaining a compact model size. However, for the extreme compression setting of ϵ=0.0486, the model size is reduced to 17.22 MB, but the Dice score drops to 83.63% and RVD increases drastically to 1233.04%, showing the detrimental effects of extreme compression on model performance.

The recall shows improvement as ϵ decreases, but this is not accompanied by better performance in terms of precision, as indicated by the reduced Dice and the inflated RVD. The increased false positives lead to higher volume estimation errors, particularly when ϵ is set to the lowest value. Visual results in Figure 6 further demonstrate the superior segmentation performance of the ϵ=0.2344 configuration, which closely matches the true tumor boundaries. The unoptimized model, while still effective, demonstrates less accurate boundary delineation, underscoring the advantage of optimizing ϵ for both segmentation accuracy and computational efficiency.



(a) (b)

Figure 6. Visual Results of Segmentation with Different ϵ Values on MSD Dataset

 doi:10.20944/preprints202503.0955.v1

4.5.4. Optimal Configuration Selection

The ϵ parameter plays a critical role in adjusting the size-performance trade-off for liver tumor segmentation. The unoptimized configuration offers maximal performance with high Dice and specificity but requires substantial storage. The ϵ=0.2344 configuration strikes an optimal balance, offering good performance with a significant reduction in size, maintaining a Dice score above 83% and RVD less than 25% for all datasets. Meanwhile, the ϵ=0.0486 configuration provides maximum compression, reducing the model size dramatically, but at the expense of substantial degradation in both Dice and RVD, highlighting the trade-offs involved in model compression.

In liver tumor pixel segmentation, precision-recall balance is paramount. A high recall value indicates good detection of true positives, but it may also increase false positives, resulting in decreased precision and inflated volume estimation errors. As ϵ decreases, the model's recall increases, but this comes at the expense of precision, as seen in the drop in Dice and the increase in RVD. This demonstrates the importance of achieving a balance between recall and precision for accurate liver volume estimation.

- **Maximal Compression**: ϵ=0.0486 (17.22 MB) for storage-constrained deployments.
- **Optimal Balance**: ϵ=0.2344 (64.16 MB) maintains greater than 83% Dice with RVD less than 25% across datasets.
- **Maximal Accuracy**: Unoptimized (324.91 MB) for non-constrained environments.

The ϵ=0.2344 configuration reduces model size by 80.25% while maintaining average Dice scores within 3.25% of the baseline across all datasets. This 64.16 MB model provides clinically acceptable RVD values (less than 5% for LiTS and MSD Task03) while requiring only 19.8% of the original storage capacity.

4.6. Optimized Model Performance Comparison with SOTA

To evaluate the performance of the proposed Swin-UNet scheme, we compare it against several state-of-the-art liver tumor segmentation methods across three datasets: 3DIRCADb, LiTS and MSD Task03. Table 6 summarizes the results using three metrics: Dice Coefficient (%), Volume Overlap Error (VOE %) and Relative Volume Difference (RVD %).

Table 6. Comparison of Liver Tumor Segmentation Methods across Datasets in terms of Dice, VOE and RVD.

Dataset	Method/Scheme	Dice (%)	VOE (%)	RVD (%)
3DIRCADb	DefED-Net [43]	66.2	34.3	0.8
	X-net [44]	69.1	36.1	0.7
	TD-Net [24]	68.2	40.8	8.4
	MS-FANet [21]	78.0	31.3	15.5
	Lgma-net [45]	83.2	24.3	0.76
	MS-UNet [46]	84.1	27.3	0.22
	MAPFUNet [47]	85.9	23.7	0.22
	Proposed Scheme	94.78	7.83	0.23
LiTS	TD-Net [24]	70.9	39.6	11.7
	MS-FANet [21]	74.2	36.7	10.7
	X-net [44]	76.4	-	-
	MAPFUNet [47]	85.8	22.0	11.02
	Lgma-net [45]	87.4	23.1	5.72
	DefED-Net [43]	87.52	23.85	5.22
	Proposed Scheme	89.06	1.66	2.51
MSD Task03	S. Muhammad *et al.* [48]	87.0	12.09	6.39
	Proposed Scheme	88.95	1.73	4.63

 doi:10.20944/preprints202503.0955.v1

For the 3DIRCADb dataset, the proposed scheme achieves a Dice score of 94.78%, significantly surpassing existing methods, such as DefED-Net (66.2%), X-net (69.1%) and TD-Net (68.2%). It also outperforms MS-FANet (78.0%), Lgma-net (83.2%) and MAPFUNet (85.9%). The proposed model achieves the highest Dice score among the methods compared, demonstrating superior segmentation accuracy. Additionally, the proposed scheme achieves a remarkably low VOE of 7.83%, which is a notable improvement over all compared methods, including MAPFUNet (23.7%) and MS-FANet (31.3%). The RVD of 0.23% is also the best among all methods, closely matching MS-UNet (0.22%) and MAPFUNet (0.22%) and providing an exceptional level of volume estimation accuracy.

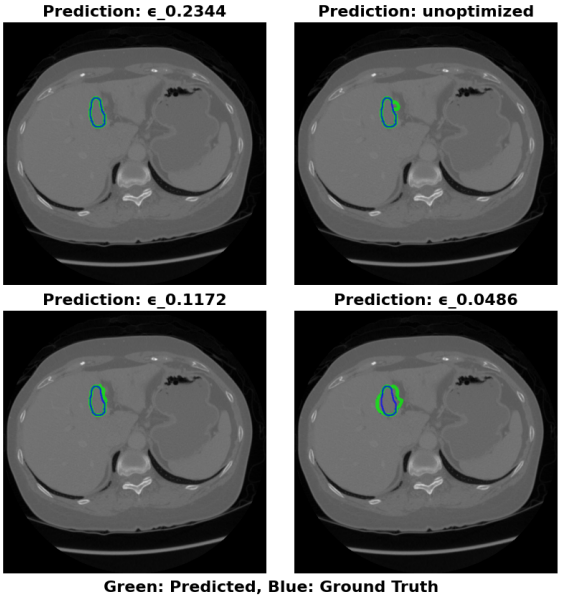


Figure 7. Graphical representation of segmentation results on the 3DIRCADb dataset.

On the LiTS dataset, the proposed scheme achieves a Dice score of 89.06%, which is substantially higher than other state-of-the-art methods, including DefED-Net (87.52%) and MAPFUNet (85.8%). This represents a significant improvement in segmentation accuracy. The VOE is drastically reduced to 1.66%, outperforming the next best value of 22.0% achieved by MAPFUNet and is much lower than values observed in other methods like TD-Net (39.6%) and MS-FANet (36.7%). Similarly, the RVD of 2.51% represents a considerable improvement over previous methods such as DefED-Net (5.22%) and Lgma-net (5.72%). These results demonstrate that the proposed Swin-UNet achieves highly accurate segmentation with minimal overlap and volume estimation errors.

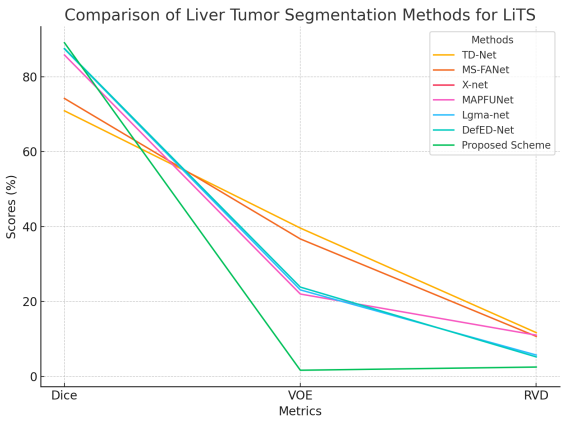


Figure 8. Graphical representation of segmentation results on the LiTS dataset.

 doi:10.20944/preprints202503.0955.v1

For the MSD Task03 dataset, the proposed Swin-UNet achieves an impressive Dice score of 88.95%, surpassing the existing method by S. Muhammad et al. (87.0%). The VOE is reduced to 1.73%, compared to 12.09% for Muhammad et al., demonstrating a substantial improvement in segmentation precision. The RVD of 4.63% also shows a noticeable improvement over the previous result of 6.39%, further reinforcing the accuracy of the proposed model.

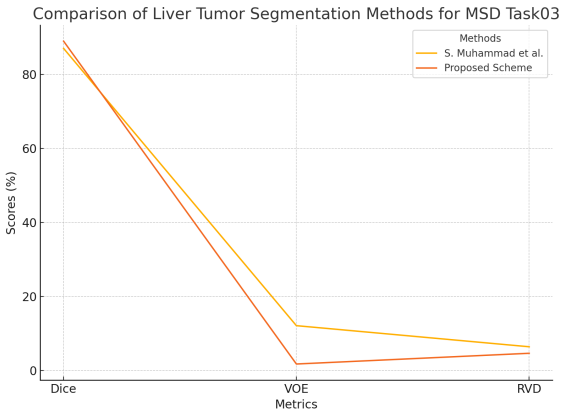


Figure 9. Graphical representation of segmentation results on the MSD Task03 dataset.

The proposed Swin-UNet model establishes new state-of-the-art results on the LiTS and MSD Task03 datasets, excelling in all performance metrics. For the 3DIRCADb dataset, it delivers the lowest VOE while maintaining competitive Dice and RVD values. These results highlight the proposed model's ability to segment liver tumors with high accuracy, minimal overlap errors and precise volume estimation, establishing it as a highly effective solution for medical imaging tasks.

5. Conclusions

This work presented an optimized Swin-UNet framework for liver tumor segmentation, specifically designed for memory-constrained edge devices. By integrating the Search and Rescue (SAR) algorithm with a quadratic penalty objective function, the model balanced segmentation accuracy (e.g., Dice scores of 94.78% on 3DIRCADb) and computational efficiency, achieving an 80.3% reduction in parameters compared to baseline architectures. The inclusion of AUC focal loss effectively addressed class imbalance, improving minority-class pixel detection in CT scans. Evaluations across three benchmark datasets (3DIRCADb, LiTS, MSD) demonstrated that the optimized framework outperformed state-of-the-art methods in key metrics, including Volume Overlap Error (1.73% on MSD) and Relative Volume Difference (0.23% on 3DIRCADb). The solution enabled energy-efficient, real-time inference on edge platforms like the Jetson Nano while maintaining data privacy—a critical requirement for clinical deployment. Although validated for liver tumors, the framework's principles of model compression and hybrid optimization showed promise for broader medical imaging applications. This advancement bridged the gap between high-accuracy segmentation and practical deployment in low-resource healthcare settings.

Author Contributions: Author Contributions: Conceptualization, W.M.I.; methodology, W.M.I.; software, W.M.I. and L.A.; validation, W.M.I., S.S.H. and M.ELA.; formal analysis, W.M.I., S.S.H. and L.A.; investigation, W.M.I., S.S. and Y.Q.Z.; data curation, W.M.I. and M.A.; writing—original draft preparation, W.M.I. and Y.Q.Z.; writing—review and editing, W.M.I., Y.Q.Z., S.S.H., L.A. and M.ELA.; visualization, W.M.I.; supervision, Y.Q.Z. and S.S.H.; project administration, Y.Q.Z.; funding acquisition, L.A. and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank Prince Sultan University for paying the APC of this article.Moreover this work was also supported by the Joint Funds of the National Natural Science Foundation of China (Grant No. U23B2063) and the National Natural Science Foundation of China (Grant No. 62076256)

Data Availability Statement: Data used in this study are public datasets and are available.

 doi:10.20944/preprints202503.0955.v1

Acknowledgments: This work was supported by EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University. The authors would like to thanks Prince Sultan University for their support.

Conflicts of Interest: "The authors declare no conflicts of interest.

References

1. Sung, H.; Ferlay, J.; Siegel, R.; Laversanne, M.; Soerjomataram, I.; Jemal, A.; others. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians* **2021**, *71*, 209–249.
2. Kazi, I.A.; Jahagirdar, V.; Kabir, B.W.; Syed, A.K.; Kabir, A.W.; Perisetti, A. Role of Imaging in Screening for Hepatocellular Carcinoma. *Cancers* **2024**, *16*, 3400.
3. Dharaneswar, S.; Kumar, B.S. Elucidating the novel framework of liver tumour segmentation and classification using improved Optimization-assisted EfficientNet B7 learning model. *Biomedical Signal Processing and Control* **2025**, *100*, 107045.
4. Ghobadi, V.; Ismail, L.I.; Hasan, W.Z.W.; Ahmad, H.; Ramli, H.R.; Norsahperi, N.M.H.; Tharek, A.; Hanapiah, F.A. Challenges and solutions of deep learning-based automated liver segmentation: A systematic review. *Computers in Biology and Medicine* **2025**, *185*, 109459.
5. Rahman, H.; Aoun, N.B.; Bukht, T.F.N.; Ahmad, S.; Tadeusiewicz, R.; Pławiak, P.; Hammad, M. Automatic Liver Tumor Segmentation of CT and MRI Volumes Using Ensemble ResUNet-InceptionV4 Model. *Information Sciences* **2025**, p. 121966.
6. Hammad, M.; ElAffendi, M.; Asim, M.; Abd El-Latif, A.A.; Hashiesh, R. Automated lung cancer detection using novel genetic TPOT feature optimization with deep learning techniques. *Results in Engineering* **2024**, *24*, 103448.
7. Rehman, A.; Mujahid, M.; Damasevicius, R.; Alamri, F.S.; Saba, T. Densely convolutional BU-NET framework for breast multi-organ cancer nuclei segmentation through histopathological slides and classification using optimized features. *CMES-Computer modeling In engineering and sciences.* **2024**, *141*, 2375–2397.
8. Hussain, S.S.; Degang, X.; Shah, P.M.; Islam, S.U.; Alam, M.; Khan, I.A.; Awwad, F.A.; Ismail, E.A. Classification of Parkinson's disease in patch-based MRI of substantia nigra. *Diagnostics* **2023**, *13*, 2827.
9. Javed, R.; Saba, T.; Alahmadi, T.J.; Al-Otaibi, S.; AlGhofaily, B.; Rehman, A. EfficientNetB1 Deep Learning Model for Microscopic Lung Cancer Lesion Detection and Classification Using Histopathological Images. *Computers, Materials & Continua* **2024**, *81*.
10. Gul, S.; Khan, M.S.; Bibi, A.; Khandakar, A.; Ayari, M.A.; Chowdhury, M.E. Deep learning techniques for liver and liver tumour segmentation: A review. *Computers in Biology and Medicine* **2022**, *147*, 105620.
11. Moghe, A.A.; Singhai, J.; Shrivastava, S. Automatic threshold based liver lesion segmentation in abdominal 2D-CT images. *International Journal of Image Processing (IJIP)* **2011**, *5*, 166.
12. Peng, W.; Zhao, Y. Liver CT image segmentation based on modified Canny algorithm. 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). IEEE, 2019, pp. 1–5.
13. Anter, A.; Hassenian, A. CT liver tumor segmentation hybrid approach using neutrosophic sets, fast fuzzy c-means and adaptive watershed algorithm. *Artif. Intell. Med.* **2019**, *97*, 105–117.
14. Xu, Y.; Quan, R.; Xu, W.; Huang, Y.; Chen, X.; Liu, F. Advances in medical image segmentation: A comprehensive review of traditional, deep learning and hybrid approaches. *Bioengineering* **2024**, *11*, 1034.
15. Al-Kofahi, Y.; Lassoued, W.; Lee, W.; Roysam, B. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Trans Biomed Eng* **2010**, *57*, 841–852.
16. Kong, H.; Akakin, H.; Sarma, S. A generalized Laplacian of Gaussian filter for blob detection and its applications. *IEEE Trans Cybern* **2013**, *43*, 1719–1733.
17. Basu, M. Gaussian-based edge-detection methods-a survey. *IEEE Trans Syst Man Cybern Part C (appl Rev)* **2002**, *32*, 252–260.
18. Chan, T.; Vese, L. Active contours without edges. *IEEE Trans Image Process* **2001**, *10*, 266–277.
19. Moga, A.; Gabbouj, M. Parallel marker-based image segmentation with watershed transformation. *J Parallel Distrib Comput* **1998**, *51*, 27–45.

 doi:10.20944/preprints202503.0955.v1

20. Saha Roy, S.; Roy, S.; Mukherjee, P.; Roy, A. An automated liver tumour segmentation and classification model by deep learning based approaches. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2022**, pp. 1–13.
21. Chen, Y.; others. MS-FANet: multi-scale feature attention network for liver tumor segmentation. *Computers in Biology and Medicine* **2023**, *163*, 107208.
22. Lakshmi, P.; Sampurna, P.; others. Deploying the model of improved heuristic-assisted adaptive SegUnet++ and multi-scale deep learning network for liver tumor segmentation and classification. *J. Real-Time Image Process.* **2025**, *22*, 8.
23. Reyad, M.; others. Architecture optimization for hybrid deep residual networks in liver tumor segmentation using a GA. *Int. J. Comput. Intell. Syst.* **2024**, *17*, 209.
24. Di, S.; others. TD-Net: A hybrid end-to-end network for automatic liver tumor segmentation from CT images. *IEEE Journal of Biomedical and Health Informatics* **2022**, *27*, 1163–1172.
25. Liu, Z.; others. PA-Net: A phase attention network fusing venous and arterial phase features of CT images for liver tumor segmentation. *Comput. Methods Programs Biomed.* **2024**, *244*, 107997.
26. Valanarasu, J.; Oza, P.; Hacihaliloglu, I.; Patel, V. Medical transformer: Gated axial-attention for medical image segmentation. Proc. Med. Image Comput. Comput. Assist. Interv., 2021, pp. 36–46.
27. Dosovitskiy, A.; others. An image is worth 16×16 words: Transformers for image recognition at scale. Proc. 9th Int. Conf. Learn. Representations, 2021, pp. 1–22.
28. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable transformers for end-to-end object detection. Proc. Int. Conf. Learn. Representations, 2021, pp. 1–16.
29. Liang, J.; Homayounfar, N.; Ma, W.C.; Xiong, Y.; Hu, R.; Urtasun, R. PolyTransform: Deep polygon transformer for instance segmentation. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 9128–9137.
30. Balasubramanian, P.; Lai, W.C.; Seng, G.; Selvaraj, J. APESTNet with Mask R-CNN for liver tumour segmentation and classification. *Cancers* **2023**, *15*, 330.
31. Chen, J.; others. TransUNet: Transformers make strong encoders for medical image segmentation. *arXiv:2102.04306* **2021**.
32. Ni, Y.; others. DA-Tran: Multiphase liver tumor segmentation with a domain-adaptive transformer network. *Pattern Recognition* **2024**, *149*, 110233.
33. Aslam, L.; Zou, R.; Awan, E.S.; Hussain, S.S.; Shaki, K.A.; Wani, M.A.; Asim, M. Hardware-Centric Exploration of the Discrete Design Space in Transformer-LSTM Models for Wind Speed Prediction on Memory-Constrained Devices **2025**.
34. Aslam, L.; Zou, R.; Awan, E.; Butt, S.A. Integrating Physics-Informed Vectors for Improved Wind Speed Forecasting with Neural Networks. 2024 14th Asian Control Conference (ASCC). IEEE, 2024, pp. 1902–1907.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser. Attention is all you need. *Advances in Neural Information Processing Systems* **2017**, *30*.
36. Alexey, D. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929* **2020**.
37. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Xie, Z.; Lin, S.; Li, H. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* **2021**, pp. 10012–10022.
38. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* **2015**, pp. 234–241.
39. Shabani, A.; Asgarian, B.; Salido, M.; Gharebaghi, S.A. Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems. *Expert Systems with Applications* **2020**, *161*, 113698.
40. Bilic, P.; Christ, P.; Li, H.B.; Vorontsov, E.; Ben-Cohen, A.; Kaissis, G.; Menze, B.H. The liver tumor segmentation benchmark (LiTS). *Medical Image Analysis* **2023**, *84*, 10268.
41. Soler, L.; Hostettler, A.; Agnus, V.; Charnoz, A.; Fasquel, J.B.; Moreau, J.; Marescaux, J. 3D image reconstruction for comparison of algorithm database, 2010.
42. Antonelli, M.; Reinke, A.; Bakas, S.; Farahani, K.; Kopp-Schneider, A.; Landman, B.A.; Cardoso, M.J. The medical segmentation decathlon. *Nature Communications* **2022**, *13*, 4128.

 doi:10.20944/preprints202503.0955.v1

43. Lei, T.; others. DefED-Net: Deformable encoder-decoder network for liver and liver tumor segmentation. *IEEE Transactions on Radiation and Plasma Medical Sciences* **2021**, *6*, 68–78.
44. Chi, J.; others. X-Net: Multi-branch UNet-like network for liver and tumor segmentation from 3D abdominal CT scans. *Neurocomputing* **2021**, *459*, 81–96.
45. Ren, W.; others. Lgma-net: liver and tumor segmentation methods based on local–global feature mergence and attention mechanisms. *Signal, Image and Video Processing* **2025**, *19*, 1–11.
46. Kushnure, D.T.; Talbar, S.N. MS-UNet: A multi-scale UNet with feature recalibration approach for automatic liver and tumor segmentation in CT images. *Computerized Medical Imaging and Graphics* **2021**, *89*, 101885.
47. Sun, J.; others. MAPFUNet: Multi-attention Perception-Fusion U-Net for Liver Tumor Segmentation. *Journal of Bionic Engineering* **2024**, pp. 1–25.
48. Muhammad, S.; Zhang, J. Segmentation of Liver Tumors by Monai and PyTorch in CT Images with Deep Learning Techniques. *Applied Sciences* **2024**, *14*, 5144.