QTSeg: A Query Token-Based Architecture for Efficient 2D Medical Image Segmentation

Phuong-Nam Tran[1], Nhat Truong Pham[2], Duc Ngoc Minh Dang[3], Eui-Nam Huh[4] and Choong Seon Hong[4,*]

[1] Department of Artificial Intelligence Kyung Hee University, Yongin, Republic of Korea
[2] Department of Integrative Biotechnology Sungkyunkwan University, Suwon, Gyeonggi-do, Republic of Korea
[3] Department of Computing Fundamental, FPT University, Ho Chi Minh Campus, Vietnam
[4] Department of Computer Science and Engineering Kyung Hee University, Yongin, Republic of Korea
tpnam0901@khu.ac.kr, truongpham96@skku.edu, ducdnm2@fe.edu.vn, {johnhuh,cshong}@khu.ac.kr

***Abstract*—Medical image segmentation is crucial in assisting medical doctors in making diagnoses and enabling accurate automatic diagnosis. While advanced convolutional neural networks (CNNs) excel in segmenting regions of interest with pixel-level precision, they often struggle with long-range dependencies, which is crucial for enhancing model performance. Conversely, transformer architectures leverage attention mechanisms to excel in handling long-range dependencies. However, the computational complexity of transformers grows quadratically, posing resource-intensive challenges, especially with high-resolution medical images. Recent research aims to combine CNN and transformer architectures to mitigate their drawbacks and enhance performance while keeping resource demands low. Nevertheless, existing approaches have not fully leveraged the strengths of both architectures to achieve high accuracy with low computational requirements. To address this gap, we propose a novel architecture for 2D medical image segmentation (QTSeg) that leverages a feature pyramid network (FPN) as the image encoder, a multi-level feature fusion (MLFF) as the adaptive module between encoder and decoder and a multi-query mask decoder (MQM Decoder) as the mask decoder. In the first step, an FPN model extracts pyramid features from the input image. Next, MLFF is incorporated between the encoder and decoder to adapt features from different encoder stages to the decoder. Finally, an MQM Decoder is employed to improve mask generation by integrating query tokens with pyramid features at all stages of the mask decoder. Our experimental results show that QTSeg outperforms state-of-the-art methods across all metrics with lower computational demands than the baseline and the existing methods.**

***Index Terms*—Convolutional neural networks, feature pyramid network, medical image segmentation, self-attention mechanism, transformer.**

I. Introduction

MEDICAL image segmentation is increasingly garnering interest within the scientific community due to its promising applications in the medical domain. Developing a precise medical image segmentation model holds the potential to aid healthcare professionals in diagnosing diseases, tailoring individualized treatment strategies for patients, and even automating analyses to predict disease outcomes.

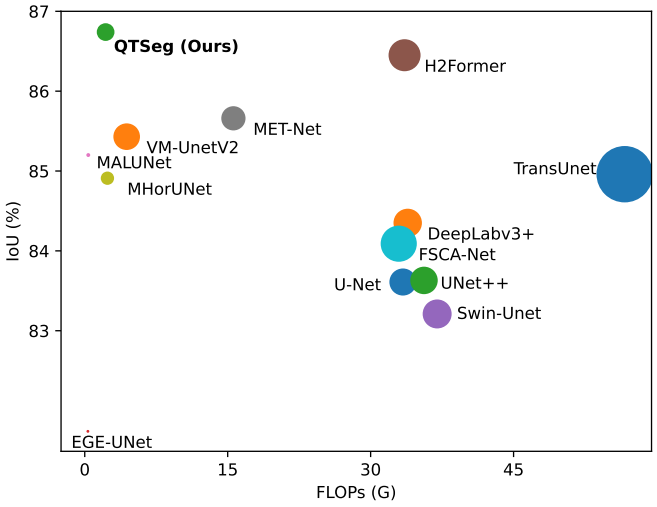


Fig. 1. The comparison of Dice score and FLOPs on the ISIC2016 dataset between QTSeg and other methods. It shows that our proposed QTSeg outperforms all the other methods in IoU score with small FLOPs. Larger circles indicate higher parameter sizes.

In computer vision, the convolutional neural network (CNN) stands as a cornerstone, driving significant advancements in the medical field. Various CNN architectures have been successfully applied for tumor or lesion segmentation in medical images [1], [2], reflecting the effectiveness of this architecture. Despite its merits, CNN encounters challenges in capturing long-range dependencies within medical images, a crucial aspect that can offer valuable insights for enhancing model accuracy. While approaches such as skip connection can mitigate this limitation, CNN still struggles to fully grasp the intricate relationships among all pixels or features in an image. Moreover, CNN's scalability is hindered by its tendency to

* Corresponding author.

demand substantial computational resources for large models without commensurate performance enhancements.

On the contrary, the transformer architecture emerges as a solution for capturing long-range dependencies, a task that proves challenging for CNNs. Originally designed for natural language processing, transformers exhibit potential for application across diverse fields. The pioneering integration of transformers into the computer vision domain was realized through architectures such as Vision Transformer (ViT) [3] and Swin Transformer [4], which treat individual pixel regions as tokens for processing. Leveraging attention mechanisms, transformer models can effectively learn intricate long-range dependencies within image pixel areas. Notably, several studies [5], [6] have showcased the efficiency of pure transformer models in medical image segmentation, surpassing the performance of traditional CNN architectures. However, transformers require high computational costs, substantial computational power, and memory resources. Training an efficient transformer model requires vast amounts of data or initialization from a well-trained model in a related domain. Additionally, transformers require sufficiently large models with quadratic computational complexity to attain the desired accuracy, leading to time-consuming training and inference processes.

Another approach in this domain involves combining CNN and transformer architectures to create a hybrid model that leverages both strengths of these two networks. Hybrid models offer a promising solution by integrating spatial information from CNNs and addressing weak long-range dependencies through the attention mechanism. Recent research has showcased the potential of this approach through various architectural designs [7]–[9]. These architectures enable models to handle diverse data types and complex multimodal tasks, such as integrating text and image data or prompt hint segmentation. However, this combination often introduces increased complexity, posing challenges for training and deploying models in real-life applications. Moreover, this architecture requires substantial computational resources in terms of both memory and processing power, particularly in the medical imaging domain, where images are typically of very high resolution. Downsampling images to fit the model input size may lead to information loss crucial for accurate predictions and interpretations, which requires a smaller and more efficient model in this field. Given the constraints and challenges associated with integrating CNN and transformer architectures, an important question arises: How can we effectively combine the strengths of both models while mitigating their limitations and achieving reduced computational complexity?

To answer this question, we propose the query token-based hybrid architecture for 2D medical image segmentation (QTSeg)[1] by integrating a CNN model as the image encoder and a transformer decoder with a mask query mechanism. Drawing inspiration from how CNN addresses local context by skip connection and the efficiency of the feature pyramid network (FPN) [10], we have designed an efficient CNN encoder based on the architecture of YOLOv8 [11] to extract multi-scale features. A multi-query mask decoder (MQM Decoder) is attached for each feature embedding to learn the relationship among feature embeddings at each level. Inspired by the Segment Anything Model (SAM) [12], we have designed the MQM Decoder incorporating query tokens to extract the target mask from the feature embeddings generated by the encoder. Unlike conventional approaches that employ a simple multi-perceptron layer (MLP) for mask extraction, our MQM Decoder aligns query tokens with feature embeddings from low-level to high-level feature embeddings using cross-attention mechanisms [13]. Additionally, the MQM Decoder leverages attention mechanisms to learn feature embeddings, facilitating enhanced relationship understanding among all features in the image and addressing the weak long-range dependency limitation of CNNs at the feature level. Experimental results have demonstrated the efficiency of our proposed QTSeg method with a competitive parameter count and low floating-point operations (FLOPs), as shown in Fig. 1.

Our main contribution can be summary as follows:

1) We proposed a query token-based hybrid architecture for 2D medical image segmentation that integrates spatial information utilizing CNN with FPN architecture while harnessing the attention mechanism in transformers to address the weak long-range dependencies inherent in CNN models at the feature level.
2) We presented a multi-query mask decoder inspired by SAM, which uses query tokens to extract the target mask at multi-level features, effectively enhancing performance.
3) We introduced a multi-level feature fusion technique to merge features from all encoder stages utilizing a simple CNN module, boosting the performance of the decoder.
4) Finally, we conducted extensive experiments to showcase the efficiency of our proposed method, surpassing other state-of-the-art models in 2D medical image segmentation tasks.

The structure of the remaining sections in this paper is outlined as follows. Section II presents the methodologies employed in previous studies and outlines the motivation behind our proposed approach. Section III provides a detailed explanation of our proposed method. Section IV presents the experimental results obtained from different datasets and highlights the significance of each module in our architecture. Finally, in Section V, we draw conclusions based on the findings presented in this paper.

II. Related Work

A. Convolutional Neural Networks in Medical Image Segmentation

Various CNN architectures have been developed for medical image segmentation in recent years, leveraging their effectiveness in capturing spatial features. One of the most common models is U-Net [1], renowned for its U-shaped architecture design. The incorporation of skip connections from previous stages enables U-Net to preserve crucial information throughout the network, contributing to its success. Building upon the achievements of U-Net, several subsequent models have emerged following a similar architecture, including

[1]Code is available at https://github.com/tpnam0901/QTSeg (v0.1.0)

UNet++ [2], Dense-UNet [14], nnUnet [15], and Attention Unet [16]. These models have significantly advanced both general image segmentation and, specifically, medical image segmentation, underscoring the potential of CNN models in the realm of medical computer vision.

B. Transformers in Medical Image Segmentation

Recently, the transformer architecture has demonstrated its potential in various computer vision tasks [17]–[21] by addressing the challenge of long-range dependencies in CNN networks. The emergence of the ViT [3] architecture has been a significant milestone in integrating attention mechanisms into computer vision applications. By dividing the input into a sequence of image patches and applying attention mechanisms to these features, the ViT architecture has effectively enhanced model performance in computer vision tasks. Subsequently, the Swin Transformer [4] introduced a window-based approach to implement self-attention using local windows, thereby reducing the model's computational complexity and enhancing overall performance. With the demonstrated effectiveness of the transformer architecture, numerous studies have integrated this framework to enhance their performance in medical image segmentation tasks. A common approach involves combining CNN and transformer models to capitalize on their respective strengths. An early adopter of this hybrid approach is TransUNet [6], which pioneers the fusion of CNN and transformer architectures for medical image segmentation. Rather than replacing the CNN model, TransUnet leverages the transformer's capabilities to enhance the existing CNN architecture. Building upon this concept, Swin-Unet [5] introduced a pure transformer architecture to extract target images in the medical image segmentation domain. However, it is worth noting that due to the quadratic complexity of the transformer architecture, these models still demand significant computational resources to generate outputs.

C. Analysis of Previous Work

In recent years, researchers have explored various architectures that combine CNN and transformer components to enhance performance. These diverse architectural approaches are summarized in Fig. 2, divided into five approaches. For the initial architectures, the CNN U-shaped design is commonly employed due to its simplicity and effectiveness. Illustrated in Fig. 2a, this architecture comprises both CNN encoder and decoder with skip connections between them. To further improve the performance of this architecture, a transformer block is inserted between the encoder and decoder for learning feature attention at a low level, as shown in Fig. 2b.

An alternative approach involves eliminating the incorporation of CNN in architecture and utilizing pure transformer, as illustrated in Fig. 2c. However, opting for pure transformers introduces challenges related to transformer architecture in computer vision, particularly in terms of computational complexity. To address this complexity issue, researchers have focused on reducing parameters by developing hybrid CNN-transformer encoder and decoder architectures. As depicted in Fig. 2d, the fusion of CNN and transformer components

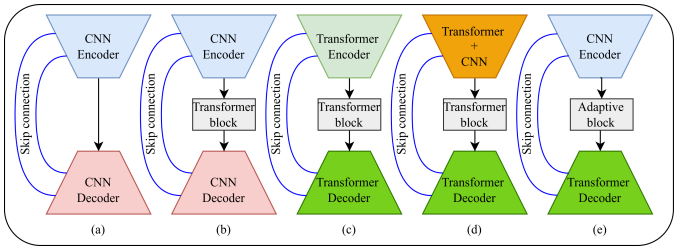


Fig. 2. Comparative conceptual of architectures for medical image segmentation. (a): The vanilla technique using CNN U-Shaped (e.g., UNet [1]). (b): The cascaded architecture of CNN and transformer module utilizing in TransUnet [6]. (c): The pure transformer architecture for image segmentation in SwinUnet [5]. (d): The efficient architecture in encoder image feature using a hybrid transformer in H2Former [7]. (e): Our proposed efficient QTSeg architecture.

has demonstrated significant potential, offering higher accuracy while maintaining competitive parameter requirements, as evidenced in [7]. However, this architecture fails to fully capitalize on the strengths of both CNN and transformer models while still exhibiting high parameter counts and FLOPs.

CNN architecture excels at extracting local spatial information effectively, but its ability to capture long-range dependencies is minimal. In contrast, transformers are good at capturing long-range dependencies, but their quadratic computational complexity poses a challenge for deploying in low-facility hospitals. To maximize the advantages offered by both CNN and transformer architectures, hybrid models have been introduced that merge the convolutional layer with an attention mechanism. However, these hybrid models tend to be more complex with high FLOPs than models that rely solely on CNN components. To tackle these challenges, our proposed method adopts a unique approach by segregating the CNN and transformer architectures instead of integrating them directly with an adaptive block between them, as shown in Fig. 2e. This architecture offers the flexibility to interchangeably replace the CNN or transformer components with pre-trained models, enabling scalability and adaptability in our network design. In addition, our proposed method effectively establishes long-range dependencies at the feature level by considering diverse feature relationships extracted from the image utilizing the attention mechanism [13], [22]. This approach helps mitigate the inherent challenge of weak long-range dependencies in CNN architectures. Moreover, the decoder in our model can comprehensively analyze all image features, often referred to as feature embeddings, by utilizing the query token mechanism introduced in the mask decoder of SAM [12]. Previous works typically employ an MLP layer or a convolutional with a kernel size of 1 at the end of the network to predict the target mask from extracted features. The performance of the model depends on how effectively the extracted feature information contributes to the last layer. Therefore, enhancing the information provided to the last layer can significantly boost the performance. Inspired by SAM [12], our proposed model integrates query tokens to predict the target mask from the extracted features. Differing from previous methods, query tokens can extract information from high-level to low-level

features by leveraging cross-attention, empowering them with enhanced capabilities for improved target mask prediction. The effectiveness of this innovative approach is demonstrated in Section IV-C and Section IV-D.

III. Methodology

A. Overall Architecture

The overall pipeline of our proposed method is depicted in Fig. 3, where the encoder is designed based on the principles of YOLOv8 [11], FPN [10], and the decoder is a stacked query mask decoder inspired by the lightweight mask decoder in SAM [12] and the U-shaped architecture. The encoder employs a CNN architecture with skip connections following the FPN design to generate pyramid features. A multi-level feature fusion (MLFF) module is then used to distribute these features across all levels before they are passed to the decoder. In the decoder stage, the query tokens are aligned with the feature embeddings to enhance mask-predicted performance in the final stages using cross-attention mechanism [13]. Furthermore, skip connections are utilized to provide additional information in subsequent phases. Finally, a dot product is applied to feature embeddings and query tokens in the final layer to generate the final mask output. The key innovation of our proposal lies in the MQM Decoder, which enables the extraction of target masks through an attention mechanism at every feature level. Within our architecture, query tokens play a pivotal role similar to that of an MLP layer, facilitating the extraction of predicted masks from the features. Differing from the conventional practice of simply adding an MLP layer at the end of the network, our unique approach involves integrating query tokens to gather additional information spanning low-level to high-level features. This strategic use of query tokens enriches the mask-generation process by harnessing a more diverse and comprehensive range of information sources for the final prediction.

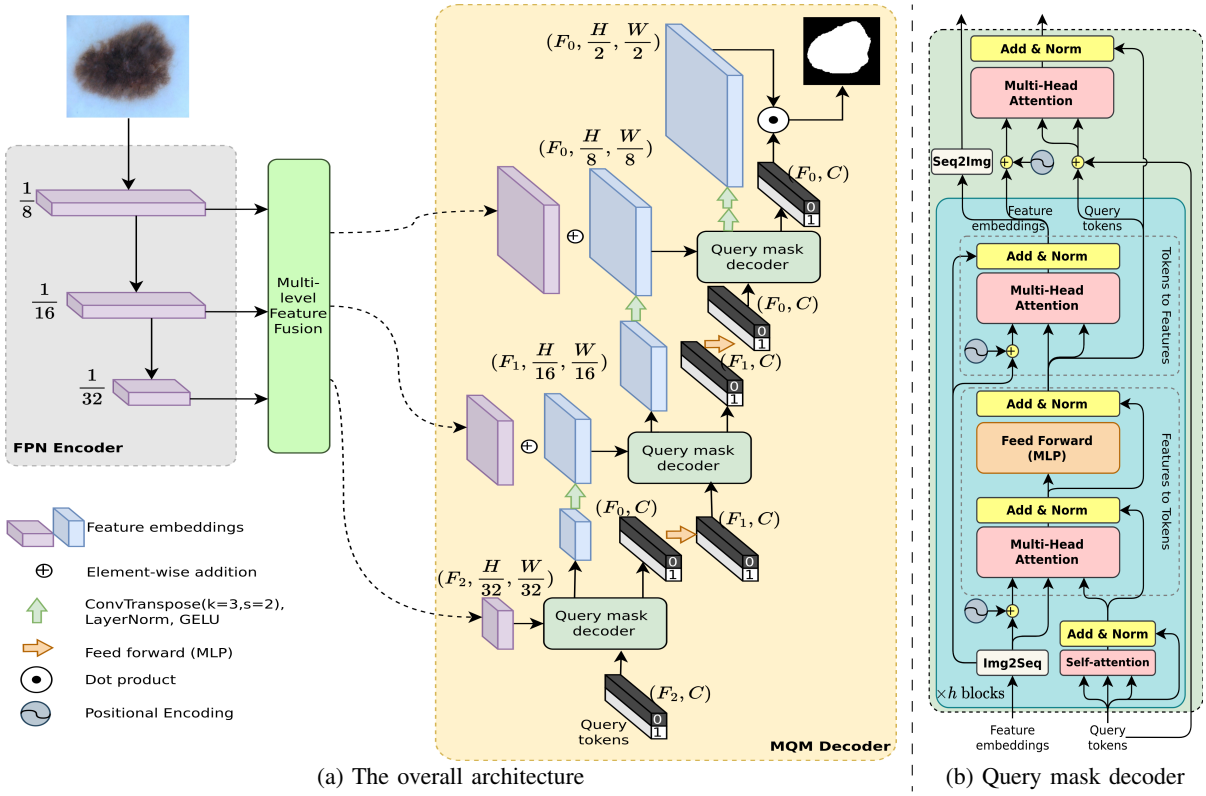
B. Feature Pyramid Network Encoder

The feature pyramid network encoder (FPN Encoder) is designed based on the YOLOv8 [11] architecture, incorporating the FPN [10] to enhance the diversity of image features. The comprehensive layout of the encoder block is depicted in Fig. 4. Initially, consider an input image $\mathcal{I} \in \mathbb{R}^{C\times H\times W}$ where C represents the number of channels of the image and H and W denote the height and width respectively. The image $\mathcal{I}$ undergoes three encoder stages to generate a feature pyramid with spatial resolutions of $\{\frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of the original input $\mathcal{I}$ denoted as $\mathcal{S}_0, \mathcal{S}_1$ and $\mathcal{S}_2$ respectively. In the initial stage, the model acquires both high-level and low-level features through the utilization of the downsample (ConvBlock) and convolutional feature (C2F) blocks. The ConvBlock block comprises a convolutional layer followed by Batch Normalization and SILU activation, enabling the layer to locally learn features within the image while simultaneously downsampling the resolution feature by half. On the contrary, the C2F block is responsible for understanding the current feature resolution without altering its dimensions. This is achieved by employing a convolutional operation with a kernel size of 1×1 and incorporating a skip connection within the block. Additionally, this layer includes a bottleneck component that executes multiple ConvBlock operations with a kernel size of 3×3. The padding operation is applied to these layers to ensure feature resolution remains consistent throughout processing.

In the second stage, the model learns features progressively from low-level to high-level features by employing upsample and concatenate operations, mirroring the architectural principles of FPN [10]. This stage utilizes the spatial pyramid pooling fast (SPPF) block [11], [23], which serves as a pooling layer that alleviates the fixed-size constraint within the network. This layer conducts information aggregation at a deeper network stage, eliminating the necessity for cropping or warping at the initial stages of processing. Subsequently, after traversing through the SPPF block, the feature embeddings undergo upscaling and concatenation with the features from the previous stage before being forwarded through another C2F block. This design facilitates backward learning from the lower to the higher stage via the Upsample, Concat, and C2F blocks, which maintain the same feature resolution at each stage. Lastly, in the final stage, the feature embeddings undergo further refinement through downsampling, transitioning from high-resolution to low-resolution features. The outputs of C2F_{15}, C2F_{18}, and C2F_{21}, as shown in Fig. 4, encapsulate the most crucial features necessary for predicting the mask at each stage. The outputs of these features are subsequently fed into the MQM Decoder to decode the features and predict the masks accurately.

C. Multi-Query Mask Decoder

Based on the lightweight decoder module introduced in SAM [12], we introduce the MQM Decoder, which is a stack of multiple query mask decoders (QM Decoder). QM Decoder utilizes the query tokens to serve as the MLP layer responsible for extracting the target mask from low to high-level features. The QM Decoder is fed with two inputs: query tokens $Q \in \mathbb{R}^{F_i\times N}$ and image features $S_i \in \mathbb{R}^{F_i\times H_i\times W_i}, i \in (0, 1, 2)$ extracted from the encoder where N is the number of classes in the dataset. In the initial stage, the decoder focuses on merging information from feature embeddings into the query tokens utilizing self-attention [22] and cross-attention [13] as shown in Fig. 3. Initially, self-attention is applied to the query tokens to refine them. Subsequently, cross-attention is employed to incorporate the details from feature embeddings into the query tokens for comprehensive information integration. During this process, skip connections are also implemented to retain information and optimize performance, following the principles of the vanilla attention mechanism. In the second stage, the decoder focuses on aligning the image features with the information from the query tokens to improve the precision of the target mask query via cross-attention. Furthermore, the attention mechanism functions at the feature level of the FPN encoder, empowering the model to learn relationships among individual features and enhancing the handling of long-range dependencies within the QTSeg framework. These two stages are iterated with h blocks before transitioning to the final self-attention process to generate the final query tokens and feature



(a) The overall architecture (b) Query mask decoder

Fig. 3. The proposed query token-based hybrid architecture for efficient 2D medical image segmentation.

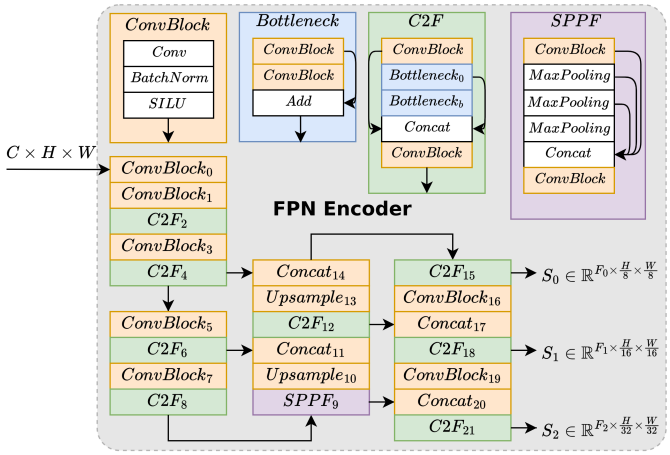


Fig. 4. Convolutional Neural Network Encoder with Feature Pyramid Network for Multi-Level Feature Extraction.

embeddings. The aligned feature embeddings and query tokens will be employed in the subsequent stage of the decoder. Algorithm 1 illustrates the complete QT Decoder pipeline. The feature and query token outputs of the current QM Decoder will be upsampled with convolutional transpose and aligned with the MLP layer for feeding to the next QM Decoder, respectively.

D. Feature Embeddings and Query Tokens

The query tokens play a vital role in our architecture. At every level of feature embeddings, we can obtain the predicted mask by utilizing the dot product, following the formula:

$$PredictMask_i = MLP(Q_i)^T \cdot UPSAMPLE(S_i) \quad (1)$$

where S_i and Q_i are the outputs of the MQM Decoder at the i^{th} stage. However, QTSeg only extracts the mask at the final stage of its learning process. After each stage, each feature embedding will be upsampled to the prior stage using the convolution transpose block. This block comprises the convolution transpose layer, followed by layer normalization and GELU activation. The output of this block will be combined with the feature embeddings from the preceding block. In addition, to mitigate computational complexity, the output of feature embeddings in the final layer will be upsampled twice to acquire higher-resolution features. While the typical approach after upsampling involves concatenating this feature with the output of the previous stage, similar to the U-shaped architecture, this method often demands higher computational resources as demonstrated in Section IV-D. To reduce the computational burdens, we employ the addition operation in this context rather than the concatenate operation to maintain a similar feature embedding dimension at each stage.

Similar to the feature embeddings, the query tokens also require adjustments to align their features appropriately for the subsequent stage. To accomplish this, we incorporate an additional MLP layer after the output of the query mask decoder to transform the feature from F_i to F_{i-1}. This process enables the query tokens to access information from all feature levels within the model. Subsequently, the outcome of the dot product operation between feature embeddings and query

Algorithm 1 QM Decoder pipeline.
Total number of classes N, embedding dimension F_i, height H_i and width W_i of the input, query tokens $Q_i \in \mathbb{R}^{F_i \times N}$, image features $S_i \in \mathbb{R}^{F_i \times H_i \times W_i}$, image feature positional embeddings $FE_i \in \mathbb{R}^{H_i \times W_i}$, and the number of hidden blocks h. The Img2Seq and Seq2Img operations represent the reshaping processes that convert images and sequences accordingly.

Procedure QM DECODER(Q_i, S_i, FE_i, h)
$query \leftarrow Q_i$
$key \leftarrow$ Img2Seq$(S_i + FE_i)$
$value \leftarrow$ Img2Seq(S_i)
for $t \leftarrow 0$ to h **do**
  Image features to tokens
  if $t = 0$ **then**
    $q \leftarrow$ LayerNorm(Self-Attention$(query)$)
    $query \leftarrow q + query$ {Skip at first layer}
  else
    $query \leftarrow$ LayerNorm(Self-Attention$(query)$)
  end if
  $attn\_out \leftarrow$ Cross-Attention$(query, key, value)$
  $query \leftarrow$ LayerNorm$(query + attn_out)$
  $query \leftarrow$ LayerNorm(MLP$(query) + query$)
  Tokens to image features
  $attn\_out \leftarrow$ Cross-Attention$(key, query, query)$
  $key \leftarrow$ LayerNorm$(key + attn_out)$
end for
Image features to tokens {Final attention}
$q \leftarrow query + Q_i$
$attn\_out \leftarrow$ Cross-Attention$(q, key, value)$
$Q_i \leftarrow$ LayerNorm$(query + attn_out)$
$S_i \leftarrow$ Seq2Img(key)
End Procedure

tokens will be rescaled to the original target size to compute the objective loss function.

E. Multi-Level Feature Fusion

In general, the current architecture of QTSeg demonstrates high performance with low computational complexity. However, its performance still lags behind recent approaches that need further improvement. As shown in Table V, employing a single decoder head does not yield performance as high as when utilizing multiple decoder heads. This highlights the significant contributions of features extracted from different stages to the query mask decoder. We hypothesize that the overall prediction quality can be enhanced by ensuring each query mask decoder receives a sufficient amount of valuable features for mask prediction computation. Based on this assumption, we designed an MLFF approach, which combines features containing information from all encoder stages and produces new features at each stage.

The concept of MLFF is similar to FPN architecture, leveraging ConvBlocks to downsample high-level features to low-level ones and concatenate them with the existing low-level features. Furthermore, the low-level feature undergoes upsampling through the ConvTranspose block ($ConvT$) and concatenates with the high-level feature. The ConvTranspose block is designed similarly to ConvBlock by replacing the Conv layer with the ConvTranspose layer. In each stage, the current stage comprises a larger proportion of features than the other stages, with 50% of feature size F for the current stage and 25% of feature size F for the remaining stages. This distribution is selected to ensure that all features, which have feature size divisible by 4, can be effectively divided into three outputs of the FPN Encoder. As a result, only a 1:1:2 ratio is suitable for dividing and concatenating the FPN Encoder's output. Subsequently, the outputs of the MLFF can be obtained in the following manner:

$$\mathcal{SF}_0 = Concat(Conv_{S00}(\mathcal{S}_0), ConvT_{S01}(\mathcal{S}_1), ConvT_{S02}(\mathcal{S}_2)) \quad (2)$$

$$\mathcal{SF}_1 = Concat(Conv_{S10}(\mathcal{S}_0), Conv_{S11}(\mathcal{S}_1), ConvT_{S12}(\mathcal{S}_2)) \quad (3)$$

$$\mathcal{SF}_2 = Concat(Conv_{S20}(\mathcal{S}_0), Conv_{S21}(\mathcal{S}_1), Conv_{S22}(\mathcal{S}_2)) \quad (4)$$

where $\mathcal{SF}_i$ is the output fusion features at stage i^{th} which have the same dimension as $\mathcal{S}_i$. The inclusion of MLFF to reorganize the current features results in enhanced overall performance with minimal computational demand, as shown in Setting 6 of Table V.

IV. Experimental Results and Discussion

A. Datasets

1) Breast Ultrasound Image Dataset (BUSI): The BUSI [24] dataset was initially introduced in 2018 and comprises a collection of breast ultrasound images obtained from 600 female patients aged between 25 and 75 years. This dataset includes a total of 780 images categorized into three classes: normal, benign, and malignant. As the normal images do not exhibit any lesions, a new dataset was obtained by excluding these normal images. The refined dataset exclusively contains cases classified as benign (437 images) and malignant (210 images), which were then partitioned using the K-fold strategy. To ensure the reproducibility of results and facilitate future comparisons, we first sort all benign and malignant samples based on their sample names. Subsequently, the benign samples were segmented into K folds by array slicing. A similar process is utilized for the malignant samples. Finally, each training and testing fold for benign and malignant cases, identified by the same slicing index, were merged together. In our experimental setup, we executed the models across five folds and subsequently reported the averaged metrics for a comprehensive evaluation.

2) Skin Lesion Segmentation Dataset (ISIC2016): The ISIC2016 [25] dataset was made public during the ISIC Challenge in 2016, with a specific focus on enhancing melanoma diagnosis through the utilization of high-quality, human-validated datasets comprising skin lesion images. The challenge provided official training and testing datasets featuring 900 dermoscopic lesion images for training and 379 for testing. These images were accompanied by ground truth masks labeled by experts and saved in binary mask format.

TABLE I
THE DETAIL SETTINGS FOR QTSEG ARCHITECTURE.

Layer names	Hyper parameters
FPN Encoder	
	$n = 16$
$ConvBlock_0$	$c = n \times 1$, $k = 3$, $s = 2$, $p = same$
$ConvBlock_1$	$c = n \times 2$, $k = 3$, $s = 2$, $p = same$
$ConvBlock_3$	$c = n \times 4$, $k = 3$, $s = 2$, $p = same$
$ConvBlock_5$	$c = n \times 8$, $k = 3$, $s = 2$, $p = same$
$ConvBlock_7$	$c = n \times 16$, $k = 3$, $s = 2$, $p = same$
$ConvBlock_{16}$	$c = n \times 4$, $k = 3$, $s = 2$, $p = same$
$ConvBlock_{19}$	$c = n \times 8$, $k = 3$, $s = 2$, $p = same$
$C2F_2$	$c = n \times 2$, $b = 1$
$C2F_4$	$c = n \times 4$, $b = 2$
$C2F_6$	$c = n \times 8$, $b = 2$
$C2F_8$	$c = n \times 16$, $b = 1$
$C2F_{12}$	$c = n \times 8$, $b = 1$
$C2F_{15}$	$c = n \times 4$, $b = 1$
$C2F_{18}$	$c = n \times 8$, $b = 1$
$C2F_{21}$	$c = n \times 16$, $b = 1$
$SPPF_9$	$c = n \times 16$, $MaxPooling(k = 5, s = 1)$
MLFF	
$Conv_{S00}$	$c = n \times 2$, $k = 1$, $s = 1$, $p = same$
$ConvT_{S01}$	$c = n \times 1$, $k = 2$, $s = 2$, $p = same$
$ConvT_{S02}$	$c = n \times 1$, $k = 4$, $s = 4$, $p = same$
$Conv_{S10}$	$c = n \times 2$, $k = 3$, $s = 2$, $p = same$
$Conv_{S11}$	$c = n \times 4$, $k = 1$, $s = 1$, $p = same$
$ConvT_{S12}$	$c = n \times 2$, $k = 2$, $s = 2$, $p = same$
$Conv_{S20}$	$c = n \times 4$, $k = 3$, $s = 4$, $p = same$
$Conv_{S21}$	$c = n \times 4$, $k = 3$, $s = 2$, $p = same$
$Conv_{S22}$	$c = n \times 8$, $k = 1$, $s = 1$, $p = same$
MQM Decoder	
$attention_2$	$head = 8$, $h_{blocks} = 3$, $F_2 = n \times 16$
$attention_1$	$head = 8$, $h_{blocks} = 2$, $F_1 = n \times 8$
$attention_0$	$head = 8$, $h_{blocks} = 1$, $F_0 = n \times 4$
MLP_2	$h_{layers} = 3$, $h_{dim} = 2048$, $out_{dim} = n \times 8$
MLP_1	$h_{layers} = 3$, $h_{dim} = 2048$, $out_{dim} = n \times 4$
MLP_0	$h_{layers} = 3$, $h_{dim} = 2048$, $out_{dim} = n \times 4$
$ConvTranspose_2$	$c = n \times 8$, $k = 2$, $s = 2$, $p = same$
$ConvTranspose_1$	$c = n \times 4$, $k = 2$, $s = 2$, $p = same$
$ConvTranspose_{0_0}$	$c = n \times 2$, $k = 2$, $s = 2$, $p = same$
$ConvTranspose_{0_1}$	$c = n \times 4$, $k = 2$, $s = 2$, $p = same$

3) BKAI-IGH NeoPolyp Dataset: The BKAI-IGH NeoPolyp [26] dataset was released by the BKAI Research Center at Hanoi University of Science and Technology in collaboration with the Institute of Gastroenterology and Hepatology (IGH) in Vietnam. This dataset comprises 1,200 images, with a division of 1,000 images for training and 200 images for testing purposes. Within the dataset, polyps are categorized into neoplastic or non-neoplastic classes, represented in red and green, respectively. While an official split of 1,000 training and 200 testing samples exists, the official test ground truth is not public due to the ongoing challenge. Thus, we partition the training set into five folds to evaluate the model's performance comprehensively. Two experiments were conducted using this dataset: binary segmentation and multi-class segmentation.

B. Implementation Details and Evaluation Metrics

1) Implementation Details: To reduce model overfitting during training, we adopted the augmentation process outlined in MISSFormer [8]. Our QTSeg model underwent training for 350 epochs on each dataset, utilizing a batch size of 32 and the AdamW optimizer with a learning rate set to 0.001 and a weight decay of 0.0001. The learning rate was adjusted through a scheduler that reduces the current learning rate by multiplying it by 0.1 every 50 epochs. However, the learning rate was not allowed to drop below 0.00001. Input images were resized to 512 x 512 and normalized to a value range of [0,1] by dividing by the maximum value within the image type. To ensure a fair comparison, no post-processing was applied to the output results. All experiments were conducted on an NVIDIA GeForce RTX 3080ti GPU operating on the Debian 12 system. The details of our model parameters are described in Table I.

2) Evaluation Metrics: We utilized standard metrics for evaluating segmentation models, including Mean Absolute Error (MAE), Accuracy (Acc), Intersection-over-Union (IoU), and Dice Similarity Coefficient (Dice).

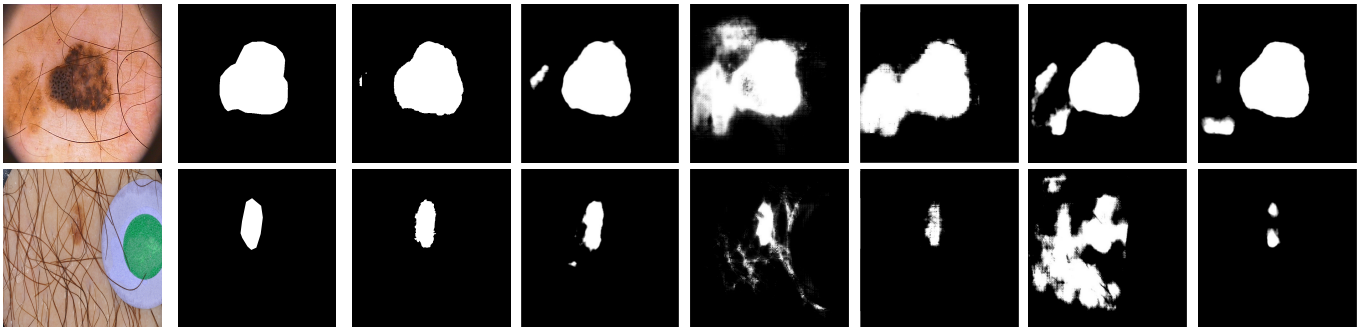
TABLE II
PERFORMANCE COMPARISON BETWEEN QTSEG AND OTHER STATE-OF-THE-ART METHODS ON THE ISIC2016 DATASET.

Method	**MAE↓**	**Acc↑**	**Dice↑**	**IoU↑**
U-Net (2015) [1]	4.98	94.52	89.56	83.61
DeepLabv3+ (2017) [27]	4.42	95.43	90.12	84.35
UNet++ (2020) [2]	5.12	94.21	89.46	83.63
nnUnet (2021) [15]	4.35	95.59	90.45	84.52
EGE-UNet (2023) [28]	5.68	94.31	88.98	81.74
MISSFormer (2023) [8]	5.45	94.36	88.37	80.75
Swin-Unet (2023) [5]	4.74	95.03	90.12	83.21
H2Former (2023) [7]	3.80	96.31	92.41	86.45
MALUNet (2023) [29]	4.55	95.45	92.01	85.20
TransUNet (2024) [6]	4.23	95.75	91.31	84.96
MET-Net (2024) [30]	4.06	95.94	91.67	85.66
MHorUNet (2024) [31]	4.74	95.26	91.13	84.91
FSCA-Net (2024) [32]	5.01	94.99	90.51	84.09
VM-UNetV2 (2024) [33]	4.43	95.57	92.14	85.43
QTSeg (Ours)	**3.59**	**96.41**	**92.42**	**86.74**

C. Comparisons with Other Methods

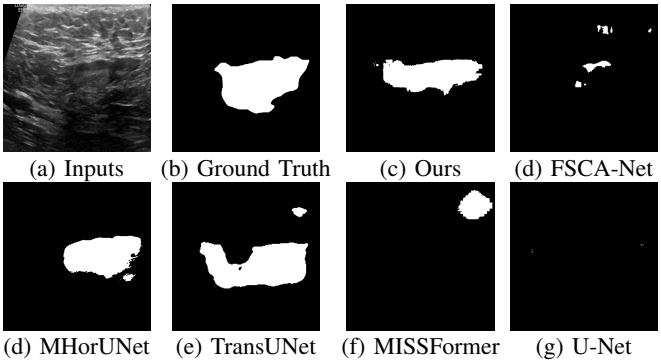
1) Results on Skin Lesion Segmentation: Table II showcases the performance of QTSeg on the skin lesion segmentation dataset, surpassing all recent methods in all evaluation metrics. Specifically, our model achieves impressive scores of 92.42% and 86.74% in Dice and IoU metrics, respectively. Furthermore, as depicted in Table VI, QTSeg demonstrates great potential by delivering outstanding results with significantly lower FLOPs and parameter counts compared to other high-accuracy methods. It is noteworthy that the parameter count of QTSeg is four times less than that of H2Former while achieving superior performance across all metrics. Additionally, in Fig. 5, the visualization of our model's predictions compared to other methods reveals that QTSeg exhibits the lowest error, with the mask closely resembling the ground truth mask. These results underscore the efficiency of integrating the mask decoder block with CNN features for medical image segmentation.

2) Results on Breast Ultrasound Image: To showcase the effectiveness of our proposed approach, we conducted experiments on the breast ultrasound image (BUSI) dataset, as depicted in Table III. QTSeg demonstrates superior performance across all metrics when evaluated using a five-fold assessment.



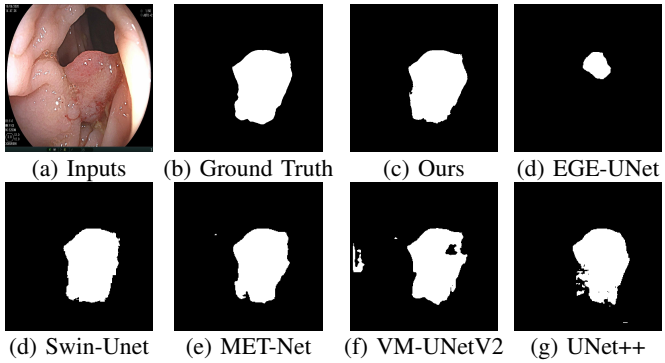
(a) Inputs (b) Ground Truth (c) **Ours** (d) H2Former (e) MISSFormer (f) Swin-Unet (g) TransUnet (h) DeepLabv3+

Fig. 5. The comparison of visualization prediction between QTSeg and other methods on the ISIC2016 dataset.



(a) Inputs (b) Ground Truth (c) Ours (d) FSCA-Net (d) MHorUNet (e) TransUNet (f) MISSFormer (g) U-Net

Fig. 6. Visualization of a failure case for the Malignant sample (108) from the BUSI dataset.



(a) Inputs (b) Ground Truth (c) Ours (d) EGE-UNet (d) Swin-Unet (e) MET-Net (f) VM-UNetV2 (g) UNet++

Fig. 7. Prediction comparison of QTSeg and other models on the 22af6b2da43f71d4dc3ddad260bfcb44 sample from the BKAI-IGH NeoPolyp dataset.

TABLE III
PERFORMANCE COMPARISON BETWEEN QTSEG AND OTHER STATE-OF-THE-ART METHODS ON THE BUSI DATASET.

Method	MAE↓	Acc↑	Dice↑	IoU↑
U-Net (2015) [1]	4.09	95.91	77.17	68.47
EGE-UNet (2023) [28]	4.85	95.15	68.24	57.11
MISSFormer (2023) [8]	4.25	95.75	78.63	69.66
Swin-Unet (2023) [5]	3.80	96.20	81.35	72.90
MALUNet (2023) [29]	4.72	95.28	75.14	60.26
TransUNet (2024) [6]	3.64	96.36	81.25	73.10
MET-Net (2024) [30]	3.98	96.02	78.54	69.80
MHorUNet (2024) [31]	4.11	95.89	77.24	67.89
FSCA-Net (2024) [32]	**3.61**	**96.39**	81.25	72.95
VM-UNetV2 (2024) [33]	5.87	94.13	67.80	51.63
QTSeg (Ours)	3.64	96.35	**82.09**	**73.85**

TABLE IV
PERFORMANCE COMPARISON BETWEEN QTSEG AND OTHER STATE-OF-THE-ART METHODS ON THE BKAI-IGH NEOPOLYP DATASET.

Method	Binary			Multi-class		
	MAE↓	Dice↑	IoU↑	MAE↓	Dice↑	IoU↑
U-Net (2015) [1]	0.99	87.25	81.33	0.92	67.74	64.11
UNet++ (2020) [2]	0.92	88.44	82.89	0.86	68.51	64.82
EGE-UNet (2023) [28]	3.20	59.98	49.28	-	-	-
MISSFormer (2023) [8]	0.95	87.98	81.89	0.87	70.00	66.40
Swin-Unet (2023) [5]	0.73	90.80	85.24	0.66	79.80	76.48
MALUNet (2023) [29]	1.84	82.09	69.67	-	-	-
TransUNet (2024) [6]	0.69	91.37	86.25	0.70	77.62	74.89
MET-Net (2024) [30]	0.74	91.05	85.99	0.71	75.96	73.07
FSCA-Net (2024) [32]	0.89	89.45	83.99	0.81	73.65	70.41
VM-UNetV2 (2024) [33]	1.26	87.71	78.16	-	-	-
QTSeg (Ours)	**0.60**	**93.13**	**88.94**	**0.59**	**79.88**	**77.54**

In particular, our QTSeg model achieves remarkable results with 82.09% Dice and 73.85% IoU, outperforming FSCA-Net by 0.84% and 0.90%, respectively. However, our QTSeg falls slightly behind FSCA-Net on the MAE and Acc metrics. Compared to the other methods, QTSeg gains notable improvements of 0.15-2.22% in Acc, 0.74-14.29% in Dice, and 0.75-22.22% in IoU. Notably, QTSeg achieves these outcomes with only around 9.41 million (M) parameters, which accounts for approximately 21.63% of the parameters used by FSCA-Net with about 43.50 million parameters. Fig. 6 illustrates a failure case on the BUSI dataset predicted by QTSeg and other methods. Our QTSeg model demonstrates minimal error, closely resembling the shape and area of the ground truth in comparison to the other methods. Despite the unclear features in the sample, QTSeg successfully segments a portion of the sample, whereas models such as U-Net [1] struggle to perform the segmentation accurately.

3) Results on BKAI-IGH NeoPolyp: The experimental results of our QTSeg model on the BKAI-IGH Neopolyp dataset are presented in Table IV. The table clearly demonstrates that our QTSeg model achieves the highest scores across all metrics and tasks (binary and multi-class). Notably, QTSeg obtains the MAE, Acc, Dice, and IoU of 0.60, 99.40%, 93.13%, and 88.94%, and 0.59, 99.41%, 79.88%, and 77.54%, for the binary and multi-class tasks, respectively. Although EGE-UNet and MALUNet are characterized by their smaller parameter sizes, they struggle to converge on the poly-segmentation

TABLE V
ABLATION STUDIES ON ISIC2016 DATASET.

Setting	Components					MAE↓	Acc↑	Dice↑	IoU↑	Params↓	FLOPs↓
	Method	Encoder	Decoder	MLFF	Feature aggregation						
Baseline	MedSAM	ViT Base	MaskDecoder	-	-	-	-	-	-	93.74 M	488.24 G
1	MedSAM	TinyViT	MaskDecoder	-	-	3.67	96.33	**92.54**	**86.91**	9.79 M	39.91 G
2	MedSAM	FPN Encoder	MaskDecoder	-	-	4.00	96.00	92.14	86.31	**4.63 M**	**1.89 G**
3	QTSeg	FPN Encoder	MQM Decoder	-	Concatenation	3.77	96.25	92.51	86.79	11.90 M	2.67 G
4	QTSeg	FPN Encoder	MQM Decoder	✓	Concatenation	4.21	95.79	91.96	86.18	12.18 M	2.78 G
5	QTSeg	FPN Encoder	MQM Decoder	-	Addition	3.87	96.13	91.92	85.87	9.41 M	2.19 G
6	QTSeg	FPN Encoder	MQM Decoder	✓	Addition	**3.59**	**96.41**	92.42	86.74	9.69 M	2.29 G

dataset due to inherent design limitations and model parameter constraints. In contrast, our QTSeg model demonstrates superior performance while maintaining competitive parameter values and lower FLOPs than the alternative methods. Regarding binary task, our QTSeg model achieves notable improvements of 1.76-33.15% in Dice and 2.69-39.66% in IoU. In terms of multi-class tasks, our QTSeg model gains notable improvements of 0.08-12.14% in Dice and 1.05-13.42% in IoU. The comparison in Fig. 7 showcases our model's predictions alongside those of other methods. It is evident that our approach achieves more precise segmentation of the poly object with minimal error compared to the other methods.

TABLE VI
MODEL COMPLEXITY AND INFERENCE TIME.

Method	Params↓	FLOPs↓	Inference Time↓
U-Net (2015) [1]	23.63 M	33.39 G	28.87 ms
DeepLabv3+ (2017) [1]	26.19 M	33.89 G	29.62 ms
UNet++ (2020) [1]	24.38 M	35.60 G	31.30 ms
EGE-UNet (2023) [28]	**0.05 M**	**0.33 G**	18.31 ms
MISSFormer (2023) [8]	42.33 M	109.45 G	92.86 ms
Swin-Unet (2023) [5]	27.27 M	36.98 G	34.82 ms
H2Former (2023) [7]	33.71 M	33.56 G	29.02 ms
MALUNet (2023) [29]	0.18 M	0.37 G	18.30 ms
TransUNet (2024) [6]	109.54 M	56.66 G	48.65 ms
MET-Net (2024) [30]	18.65 M	15.60 G	**16.62 ms**
MHorUNet (2024) [31]	4.96 M	2.38 G	28.00 ms
FSCA-Net (2024) [32]	43.50 M	32.95 G	22.89 ms
VM-UNetV2 (2024) [33]	22.77 M	4.40 G	29.56 ms
QTSeg (Ours)	9.41 M	2.19 G	22.55 ms

D. Ablation Studies

To investigate the effectiveness of our architecture, we conducted several ablation studies on the QTSeg architecture. Due to resource constraints, we did not evaluate the baseline MedSAM with the ViT base architecture. Instead, we replaced the ViT base with TinyViT to assess the model's performance within the MedSAM framework, as depicted in Setting 1 of Table V. This setting makes MedSAM have the same architecture as MobileSAM [34], which shows high performance on general segmentation tasks. Furthermore, we replaced TinyViT with our FPN Encoder model without altering the MedSAM architecture (Setting 2). Comparing Settings 1 and 2, the transition from TinyViT to the FPN Encoder led to a decrease in both model accuracy and complexity. To enhance model performance to meet a similar baseline standard, we incorporated the U-Shape architecture outlined in Section III and Settings 3, 4, 5, and 6 in Table V. Settings 3 and 4 demonstrate the utilization of feature concatenation on the channel axis instead of element-wise addition for the skip connections from the encoder to the decoder. The table illustrates that implementing the U-shape architecture resulted in an overall performance boost but increased model complexity compared to Setting 2. Interestingly, adding the MLFF module alongside feature concatenation did not yield performance improvements and decreased model performance for this setting.

To address the complexity of QTSeg, we replaced feature concatenation with element-wise addition, as shown in Setting 5. This adjustment reduced complexity while enhancing accuracy. Furthermore, incorporating the MLFF module into Setting 5, as in our proposed QTSeg method (Setting 6), resulted in further enhancements. Notably, with parameters similar to Setting 1, QTSeg achieved a substantial reduction in model complexity from 39.91 GFLOPs to 2.29 GFLOPs while maintaining or even surpassing performance metrics such as MAE and Acc.

E. Model Complexity

Table VI provides a detailed comparison of the computational complexity of our proposed QTSeg method with several other approaches, considering parameters, FLOPs, and inference time on the NVIDIA GeForce RTX 3090 GPU. It should be noted that the input size was set to 512 x 512 for our model in this evaluation. Notably, FSCA-Net stands out with 43.50 million parameters and 32.95 GFLOPs, making it approximately five times larger in scale compared to our proposed method. Similarly, MISSFormer and H2Former also exhibit high parameter counts and FLOPs due to their reliance on a pure transformer structure featuring global self-attention and hybrid transformer blocks, leading to substantial computational demands. On the other hand, EGE-UNet and MALUNet showcase lower parameter counts and FLOPs, but their inference times are marginally slower than our models. This discrepancy can be attributed to the design of our methods, which are optimized for parallel computing on GPUs, resulting in faster inference times. However, in scenarios with limited resources or when computing on CPUs, our approach may lag behind these models. It is worth noting that EGE-UNet and MALUNet do not consistently achieve high metrics across all experiments due to their parameter constraints. In contrast, our hybrid architecture strikes a balance between delivering highly accurate results and maintaining a competitive computational cost, making it a more effective solution overall. More specifically, QTSeg significantly outperforms

all the existing methods in terms of IoU and FLOPs on the ISIC2016 dataset (Fig. 1), except for EGE-UNet, MALUNet, and MHorUNet in terms of FLOPs. Based on Table VI, Fig. 1, and all the other experiments, it can be seen that our model offers the best trade-off between computational complexity and performance.

V. Conclusion

In this study, we harnessed the strengths of CNN models by incorporating the FPN architecture to generate multi-level features. By combining this approach with a mask decoder inspired by the lightweight decoder in SAM, we introduced a query token-based hybrid architecture for 2D medical image segmentation known as QTSeg. QTSeg leverages CNN as an image encoder and transformer as a decoder, offering a unique fusion of these two powerful models. Furthermore, we introduced an MLFF module to effectively contribute to the multi-level feature at each stage, resulting in enhanced prediction of the MQM decoder. QTSeg exhibited impressive results across all experimental datasets while maintaining a competitive parameter count. Our research demonstrates that our proposed architecture strikes an optimal balance between model complexity and segmentation performance on common medical image segmentation tasks such as poly segmentation, lesion segmentation, and breast cancer segmentation. In the future, we will extend the application of this architecture to general tasks such as image segmentation and semantic segmentation with flexible prompts for segmenting diverse objects. This architecture can streamline the complexity of SAM while maintaining performance levels comparable to those of the baseline model.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[2] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 1856–1867, 2020.

[3] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[4] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[5] H. Cao *et al.*, "Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation," in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds. Cham: Springer Nature Switzerland, 2023, pp. 205–218.

[6] J. Chen *et al.*, "TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers," *Medical Image Analysis*, p. 103280, 2024.

[7] A. He, K. Wang, T. Li, C. Du, S. Xia, and H. Fu, "H2Former: An Efficient Hierarchical Hybrid Transformer for Medical Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 42, no. 9, pp. 2763–2775, 2023.

[8] X. Huang, Z. Deng, D. Li, X. Yuan, and Y. Fu, "MISSFormer: An Effective Transformer for 2D Medical Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 42, no. 5, pp. 1484–1494, 2023.

[9] H.-Y. Zhou *et al.*, "nnFormer: Volumetric Medical Image Segmentation via a 3D Transformer," *IEEE Transactions on Image Processing*, vol. 32, pp. 4036–4045, 2023.

[10] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[11] R. Varghese and S. M., "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024, pp. 1–6.

[12] A. Kirillov *et al.*, "Segment anything," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 3992–4003.

[13] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 357–366.

[14] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, "H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes," *IEEE Transactions on Medical Imaging*, vol. 37, no. 12, pp. 2663–2674, 2018.

[15] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021.

[16] O. Oktay *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[17] Q. Zhang and Y.-B. Yang, "Rest: An efficient transformer for visual recognition," *Advances in neural information processing systems*, vol. 34, pp. 15 475–15 485, 2021.

[18] D. T. Tran *et al.*, "SwinTExCo: Exemplar-based video colorization using Swin Transformer," *Expert Systems with Applications*, vol. 260, p. 125437, 2025.

[19] D.-H. Hoang, A.-K. Tran, D. N. M. Dang, P.-N. Tran, H. Dang-Ngoc, and C. T. Nguyen, "RBBA: ResNet - BERT - Bahdanau Attention for Image Caption Generator," in *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, 2023, pp. 430–435.

[20] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, no. 1, p. 654, 2024.

[21] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, "Transreid: Transformer-based object re-identification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 013–15 022.

[22] A. Vaswani *et al.*, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[24] W. Al-Dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, "Dataset of breast ultrasound images," *Data in Brief*, vol. 28, p. 104863, 2020.

[25] D. Gutman *et al.*, "Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1605.01397*, 2016.

[26] P. Ngoc Lan *et al.*, "NeoUNet : Towards Accurate Colon Polyp Segmentation and Neoplasm Detection," in *Advances in Visual Computing*, G. Bebis, V. Athitsos, T. Yan, M. Lau, F. Li, C. Shi, X. Yuan, C. Mousas, and G. Bruder, Eds. Cham: Springer International Publishing, 2021, pp. 15–28.

[27] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 833–851.

[28] J. Ruan, M. Xie, J. Gao, T. Liu, and Y. Fu, "EGE-UNet: An Efficient Group Enhanced UNet for Skin Lesion Segmentation," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, and R. Taylor, Eds. Cham: Springer Nature Switzerland, 2023, pp. 481–490.

[29] J. Ruan, S. Xiang, M. Xie, T. Liu, and Y. Fu, "MALUNet: A Multi-Attention and Light-weight UNet for Skin Lesion Segmentation," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022, pp. 1150–1156.

[30] A. Iqbal and M. Sharif, "Memory-efficient transformer network with feature fusion for breast tumor segmentation and classification task,"

Engineering Applications of Artificial Intelligence, vol. 127, p. 107292, 2024.

[31] R. Wu *et al.*, "MHorUNet: High-order spatial interaction UNet for skin lesion segmentation," *Biomedical Signal Processing and Control*, vol. 88, p. 105517, 2024.

[32] D. Tan, R. Hao, X. Zhou, J. Xia, Y. Su, and C. Zheng, "A Novel Skip-Connection Strategy by Fusing Spatial and Channel Wise Features for Multi-Region Medical Image Segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 9, pp. 5396–5409, Sep. 2024.

[33] M. Zhang, Y. Yu, S. Jin, L. Gu, T. Ling, and X. Tao, "VM-UNET-V2: Rethinking Vision Mamba UNet for Medical Image Segmentation," in *Bioinformatics Research and Applications*, W. Peng, Z. Cai, and P. Skums, Eds. Singapore: Springer Nature Singapore, 2024, pp. 335–346.

[34] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster segment anything: Towards lightweight sam for mobile applications," *arXiv preprint arXiv:2306.14289*, 2023.