Real-Time Polyp Detection in Colonoscopy using Lightweight Transformer

Youngbeom Yoo[1,*], Jae Young Lee[1,*], Dong-Jae Lee[1], Jiwoon Jeon[2], and Junmo Kim[1]
[1]School of Electrical Engineering, KAIST, South Korea
bum8552@hufs.ac.kr, {mcneato, jhtwosun, junmo.kim}@kaist.ac.kr
[2]AINex, South Korea jiwoon.jeon@ainex.io

Abstract

Colorectal cancer (CRC) represents a major global health challenge, and early detection of polyps is crucial in preventing its progression. Although colonoscopy is the gold standard for polyp detection, it has limitations, such as human error and missed detection rates. In response, computer-aided detection (CADe) systems have been developed to enhance the efficiency and accuracy of polyp detection. As deep learning gained prominence, the incorporation of Convolutional Neural Networks (CNNs) into CADe systems emerged as a breakthrough approach. However, CADe systems based on CNNs often demand significant computational resources, making them unsuitable for deployment in resource-constrained environments. To mitigate this, we propose a novel and lightweight polyp detection model that integrates a Transformer layer into the You Only Look Once (YOLO) architecture, focusing on optimizing the neck part responsible for feature fusion and rescaling. Our model demonstrates a substantial reduction in computational complexity and the number of parameters, without compromising detection performances. The lightweight model makes it accessible and feasibly deployable in medically underserved regions, serving a significant public interest by potentially expanding the reach of critical diagnostic tools for CRC prevention. By optimizing the architecture to reduce resource requirements while maintaining performance, our model becomes a practical solution to assist healthcare professionals in the real-time identification of polyps, even with resource-constraint devices.

1. Introduction

Colorectal cancer (CRC) imposes a substantial healthcare burden worldwide, given its status as a prevalent form of cancer. According to WHO [55], CRC was the third most common cancer type with nearly 2 million cases, and the second leading cause of cancer-related deaths, accounting for approximately 1 million fatalities annually in 2020. One of the precursors to CRC is the presence of polyps, which are generally benign growths that can, over time, develop into malignant tumors [6].

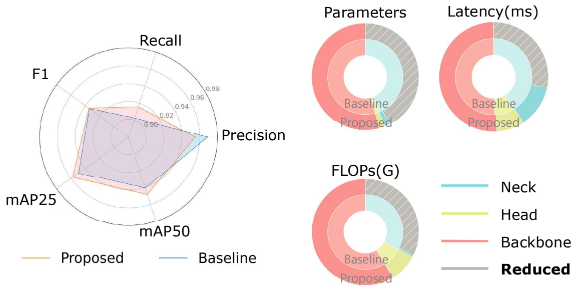

Figure 1. Comparison between YOLOv5m (baseline) and the proposed YOLOv5m-TST model with corresponding model size to the baseline at image size 320. YOLOv5m-TST has fewer parameters, fewer FLOPs, and lower latency. YOLOv5m-TST shows comparable performances.

Colonoscopy is a diagnostic procedure that involves inserting an endoscope through the anus to visualize the colon, allowing for the diagnosis of inflammation, polyps, tumors, and other abnormalities within the colon [56]. Colonoscopy serves not only as a means of examination but also permits immediate biopsy of suspicious areas for histopathological analysis. Additionally, it enables direct removal of polyps if present and provides the capability to control bleeding if encountered, thereby serving both diagnostic and therapeutic purposes.

Colonoscopy remains the gold standard in both the detection and removal of polyps, consequently playing a critical role in CRC prevention [2]. So detecting and removing polyps at an early stage plays a crucial role in reducing incidence and mortality rates associated with CRC [57, 58]. However, conventional colonoscopy relies on the expertise of the endoscopist, and even in skilled hands, polyp detection is not always guaranteed. Studies have reported miss

[*]equal contribution

rates of 16.8% to 28% for polyps during colonoscopy examinations [22], emphasizing the need for tools to increase polyp detection accuracy.

To improve the efficiency and accuracy of polyp detection, computer-aided detection (CADe) systems have been developed [50]. In the earlier CADe, hand-crafted filters and feature extraction techniques were prominent [1, 12]. These traditional methods primarily focused on analyzing images to detect patterns that could signify the presence of polyps [3, 13, 21, 47]. However, they required manual fine-tuning and were often not generalizable or scalable. The growing utilization of deep learning, particularly Convolutional Neural Networks (CNNs), has shown impressive performance in polyp detection [4, 16, 39, 40, 46]. Examples of CNN-based approaches like You Only Look Once (YOLO) [41] and region-based convolutional neural networks (R-CNN) have shown remarkable potential in polyp detection. However, it should be emphasized that the effectiveness of CADe relies on its ability to not only accurately detect but also work in real-time.

The need and significance of real-time polyp detection cannot be overstated, and its impact on patient outcomes is substantial. Real-time CADe systems empower medical professionals to take immediate therapeutic actions during colonoscopy procedures, which is crucial for early intervention and improving patient prognosis [8]. This aspect is particularly vital when we consider the inherent limitations in the manual review of colonoscopy data. It becomes labor-intensive and impractical for endoscopists to meticulously review extensive colonoscopy video recordings. This limitation is critical because, in practice, a second analysis of these videos is often bypassed due to time constraints and the cumbersome nature of the task [14]. This omission can have significant consequences, as it may lead to missing early-stage polyps that are vital for taking preventive measures against CRC. Thus, several studies proposed the real-time polyp detection algorithms [20, 26, 33, 34, 36–38, 50, 52, 60, 61]. However, while these high-performance CADe systems have revolutionized polyp detection, they are not without their limitations.

One significant drawback is the considerable computational power. They tend to be resource-intensive, often necessitating expensive, high-performance GPUs for real-time operation. While these GPUs can provide excellent performance, their cost and size limit the accessibility and ubiquity of advanced polyp detection technologies. In medical settings, there is often a need for portable devices, which have limited computational resources. As such, there is an imperative need for lightweight models that can operate efficiently on these devices without compromising accuracy or performance. Moreover, the deployment of such high-powered, GPU-based systems may not always be feasible, especially in resource-constrained environments, such as rural clinics or mobile healthcare units. Therefore, reconciling the conflicting requirements of performance and practicality in CADe systems is a significant challenge.

To address these challenges, we propose a lightweight polyp detection model. Our primary objective is to optimize the computational costs in the neck part of the YOLO model. The neck is crucial as it combines features across various scales and abstraction levels to enhance detection performance. We hypothesized that the neck part in YOLO can be further optimized using Transformer [51]. We integrate the Transformer into the YOLO architecture. The Transformer is well-regarded for its ability to capture broader contextual information using attention mechanisms. This characteristic greatly aids in improving the fusion of global and local features, resulting in more efficient processing. Additionally, by employing cross-attention, where local and global information are used as queries and keys/values respectively, it becomes possible to obtain global information attended to by local information. It enables the model to capture intricate relationships between different spatial scales and abstractions.

For integrating the Transformer into YOLO, we incorporate the Token-Sharing Transformer (TST) [25] in place of the CNN-based neck into the YOLOv5 framework [17]. The TST layer consists of multi-head cross-attention and a feed-forward network, capturing multi-level features with global information. Using TST, our model uses both local and global features from the backbone, leading to effective feature fusion. As a result, our model reduces both computational burden and the number of parameters.

As shown in Fig. 1, our model (YOLOv5m-TST) has significantly reduced both the number of Parameters and FLOPs in the neck part of the architecture. Vision Transformer (ViT) [9] models are usually slower than CNNs due to various factors including a large number of parameters and increased computational burden [28, 32, 54]. However, experimental results show that even with the utilization of a Transformer, our model reduces latency compared to the YOLO model. We conduct extensive experiments to demonstrate the effectiveness of our lightweight approach in comparison to the YOLO models and versions. Experiments show that our Transformer-based approach achieves comparable or improved accuracy while significantly reducing the computational requirements. This enables real-time performance even on resource-constrained hardware platforms, which can make low-cost and high-quality examination possible even in medically underserved regions.

2. Related Work

2.1. Polyp Detection

Polyp detection using CNN has been actively studied in recent years. Shin *et al.* [46] adopted region-based two-stage CNN for polyp detection. Qadir *et al.* [39] devised a two stage-stage process, initially proposing regions of interest through CNN-based object detector networks and subsequently employing a unit to reduce false positives. Qadir *et al.* [40] proposed 2D Gaussian masks and single-shot feed-forward fully convolutional neural networks. Jia *et al.* [16] presented a two-stage approach, called PLPNet, using advanced residual networks and feature pyramids. Zhang *et al.* [60] constructed Single Shot MultiBox Detector (SSD) based architecture, which is called SSD-GPNet. Zheng *et al.* [61] applied YOLO [41] for polyp detection and Urban *et al.* [50] used a variation of YOLO for better localizing single objects. Lee *et al.* [26] developed a polyp detection model using YOLOv2 [42], and applied median filtering to reduce the number of false positives in the video analysis. Nogueira *et al.* [34] utiliezed YOLOv3 architecture and incorporated post-processing step to minimize false positives. Jha *et al.* [14] devised a model called ColonSegNet, which focuses more on speed rather than accuracy and showed that YOLOv4 [5] was faster compared with YOLOv3+spp [43], EfficientDet [48], RetinaNet [29], and also six times faster than Faster R-CNN [44]. Misawa *et al.* [33] applied YOLOv3 and proposed SUN Dataset. Pacal and Karaboga [37] proposed a YOLOv4-based model applying the Cross Stage Partial Networks(CSPNet) [53] to the whole architecture and used Transformer [51] in the last block on the backbone. Pacal *et al.* [38] scaled YOLOv3 and YOLOv4 with CSPNet and they tested various activation functions and loss functions to optimize their model. Wan *et al.* [52] proposed a YOLOv5 model based on a self-attention mechanism for polyp target detection. Karaman *et al.* [20] used YOLOv5 along with the Artificial Bee Colony (ABC) [19] optimization algorithm, where YOLOv5 is utilized for polyp detection and ABC is deployed to enhance the model performance by finding the optimal activation functions and hyper-parameters. Ou *et al.* [36] devised Polyp-YOLOv5-Tiny, which reduced the number of convolutional kernels by half and removed the part of the head.

With minimal architectural modifications, the existing polyp detection methods primarily relied on YOLO-based frameworks, which consist of three parts: backbone, neck, and head. Unlike the existing polyp detection methods, we replace the neck part of YOLO using the Transformer. In contrast to the existing models using self-attention based Transformers for certain parts of the YOLO model, we employed a multi-head cross-attention between tokens from different layers. Through this change, we not only reduce the computational cost but also manage to make a lighter model while minimizing performance drop.

2.2. Lightweight Transformer

ViT [9] showed that Transformer [51] architecture can show the success on the image classification task. With attention mechanism, it enables the model to learn global information. After ViT, Transformer-based architectures were actively studied for various applications. DeiT [49] used distillation learning to reduce the needs of large datasets. DETR [7] used a Transformer encoder-decoder architecture for object detection. D-DETR [62] used attention modules that focus on a small set of key sampling points. Swin-Transformer [31] introduced a hierarchical structure with shifted windows to efficiently address the challenges of scale variation and high resolution in images. However, these models have a large number of parameters and heavy computation complexity. Therefore, various studies have been made for real-time use and to make lightweight models. LeViT [10] used multi-stage Transformer architecture using attention as a downsampling mechanism. MobileViT [32] reduced computational costs based on MobileNetV2 [45] backbone with repeated CNN-Transformer blocks. Efficientformer [28] showed that Vision Transformer can operate at MobileNetV2 speed on mobile devices. Topformer [59] proposed token pyramid pooling to reinforce the model's representation ability. TST [25] proposed global token sharing to inject global information into the multi-level features.

Previous works have successfully employed lightweight Transformer architectures for tasks such as classification, segmentation, and depth estimation. For polyp detection, we adopted TST as a lightweight Transformer in the proposed method. By utilizing TST in the YOLO model, we achieved computational efficiency and elevated real-time performance without a significant performance drop.

3. Proposed Method

3.1. Design Concept

To design an architecture for lightweight polyp detection on edge devices, we start by revisiting the original YOLOv5 [17] model and understanding the functional principles of the lightweight Transformer (i.e., TST [25]).

YOLOv5 architecture is divided into three parts: backbone, neck, and head. The backbone efficiently processes the input image and generates feature maps at different scales, enabling the model to extract multi-level features. In YOLOv5, the CSP-Darknet53 architecture [5, 53] is utilized as the backbone. The Neck performs feature fusion and transformation to generate more informative and context-rich representations. In YOLOv5, the SPPF [11] and CSP-PAN [30, 53] is utilized as the neck. The head predicts the bounding box coordinates, objectness scores, and class probabilities for multiple objects across different scales and aspect ratios. In YOLOv5, the head part follows the YOLOv3 head [43]. Particularly, our primary focus is to optimize the neck. In object detection, the neck plays an instrumental role as it integrates feature maps of differ-

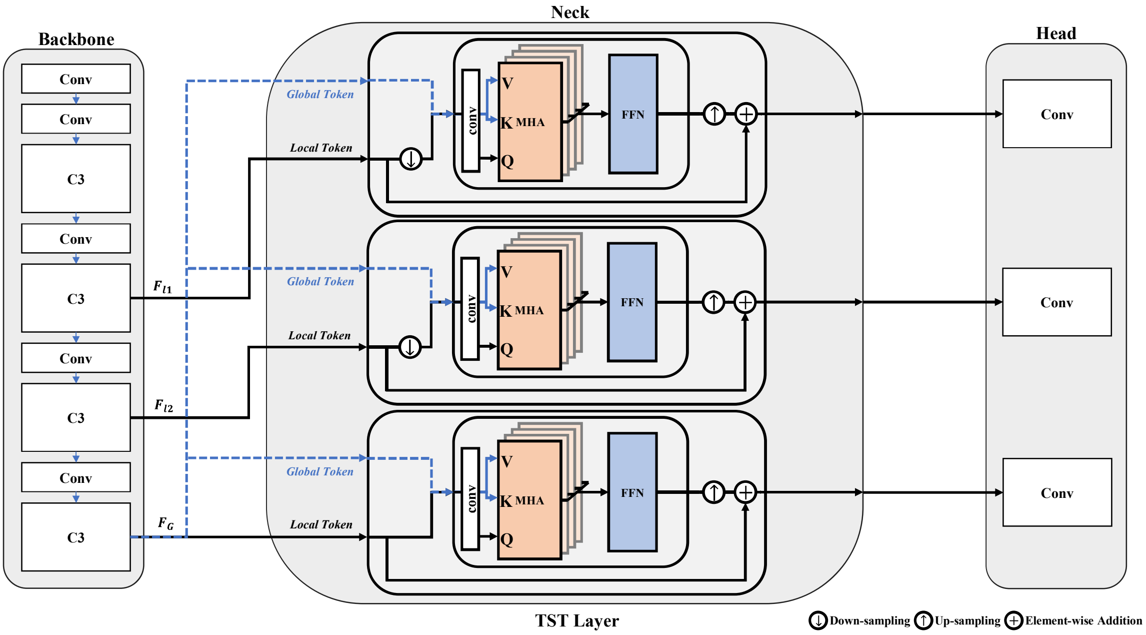

Figure 2. The Overall Architecture of our proposed YOLOv5-TST model.

ent scales and resolutions from the backbone, allowing the model to capture both low-level and high-level visual features. This fusion and transformation within the neck are essential for generating richer representations that can greatly benefit the object detection process.

TST [25] is devised for the lightweight monocular depth estimation task. TST focused on learning the multi-level features containing global information, with global token sharing. TST was incorporated into the encoder-decoder shortcut, replacing the conventional skip connection. To handle multi-level features, they down-sampled the local features to match the size of the global features. To share global information with each level of the features, they used the global token as a query and the local token as a key and value. After passing through the Transformer block, they up-sampled the TST output to restore the feature size to the original level. The result shows high throughput, without performance drop.

Given that the CNN-based neck is crucial in feature fusion, we hypothesized that by leveraging the attention mechanism, which excels at capturing global information, the TST could effectively replace the neck. Specifically, if the query is set to the local token and the key and value are set to the global token, the cross-attention result can obtain the global token that has been attended to by the local token. By adding the output of the Transformer block to the local token with residual connection, the global information deemed important in the local context gets weighted during learning, potentially making feature fusion more efficient. Through our experiments, we observed a functional similarity between the role of the CNN-based neck in feature fusion and the capabilities of the TST.

Based on this consideration, we replace the SPPF and CSP-PAN with TST to devise a lighter model. By substituting the neck with TST, we achieved a significant reduction in the number of parameters while maintaining performance comparable to the existing model. Furthermore, the real-time detection performance, measured in edge devices, showed a substantial improvement.

3.2. Overall Architecture

The TST layer consists of a convolution block, multi-head cross-attention (MHA), and a feed-forward network (FFN). Fig. 2 shows the detailed architecture of our model.

Considering an input image $I \in \mathbb{R}^{H \times W \times C}$, where H, W, and C represent the height, width, and RGB channels of the image, respectively. Backbone extracts a set of multi-resolution feature maps, denoted as $\mathbf{F} = \{F_{l1}, F_{l2}, F_G\}$, where $F_{l1} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times C_{l1}}$, $F_{l2} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_{l2}}$, $F_G \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_G}$. These feature maps serve as local tokens F_{l1}, F_{l2}, F_G and a global token F_G.

Within the TST layer, the local tokens are first down-sampled to match the resolution of the global token. Specifically, the local tokens from $l1$ and $l2$ are down-sampled to $F_{l1}^{\downarrow} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_{l1}}$ and $F_{l2}^{\downarrow} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_{l2}}$, respectively. Following this, the down-sampled local tokens and the global token are input into the Transformer block.

In the MHA process, we aimed to use the local tokens as queries Q, and global tokens, which serve as information-sharing tokens, as keys K and values V. We encountered an issue as the channel dimensions of the local tokens were different from those of the global tokens, making it impossible to directly use them through the MHA. To address this issue, we employed a convolution block to transform the input dimensions, enabling us to obtain queries, keys, and values with compatible dimensions for efficient processing within the MHA. The output from the MHA is then passed through FFN and is up-sampled to match the resolution of each local token. This is followed by the Transformer block, which effectively fuses the local and global tokens. Residual connection is utilized to integrate the information from the global tokens that the MHA process deemed important back into the original local feature map. This yields a synergistic effect by adding global information to the local information. Finally, the enriched feature map is forwarded to the head. In polyp detection, when the data passes through the detection head, bounding boxes are produced around areas that might contain polyps, along with confidence scores.

3.3. Token-Sharing Transformer Layer

In the convolution block, we perform batch normalization after the dimension transformation. For MHA, the dimensions of the queries and keys are set to 4, while the dimension for values is set to 16. In each Transformer block, the number of heads is set to 4. The FFN first processes the MHA output through a ReLU activation function, and then performs a dimension transformation to ensure that the output dimensions match the channel size of the local tokens that were initially fed into the layer. Additionally, batch normalization is applied to the FFN output.

4. Experiments

4.1. Datasets

We use three colonoscopy datasets for our experiments. To test the generalizability of the experiment, we combine three datasets: SUN [33], KUMC [27], and Kvasir-SEG (Kvasir) [15]. Also, we conduct experiments for each dataset separately. Using the datasets, the training and test sets are divided into different cases based on patient numbers, approximately 8:2 as follows. Furthermore, we conduct experiments using a 5-fold cross-validation approach.

SUN includes 49,136 polyp images taken from different 100 cases, which are fully annotated with bounding boxes. We use 80 cases (38,249 images) as a training set and 20 cases (10,887 images) as a test set.

KUMC includes 37,900 polyp images taken from different 153 cases, which are fully annotated with bounding boxes. We use 124 cases (29,261 images) as a training set and 29 cases (8,639 images) as a test set.

Kvasir includes 1,000 polyp images, which are fully annotated with bounding boxes and masks. We use 800 images as a training set and 200 images as a test set.

4.2. Hardware Platform

We evaluated model performance on embedded devices: NVIDIA Jetson Nano, NVIDIA Jetson TX2, and NVIDIA Jetson AGX Xavier. Jetson Nano has 128 CUDA cores Maxwell architecture GPU with a Quad-core ARM A57 MPCore CPU and 4GB of RAM. Jetson TX2 has 256 CUDA cores Pascal architecture GPU with Dual-Core Denver, Quad-core ARM A57 MPCore CPU, and 8GB of RAM. Jetson AGX Xavier has 512 CUDA cores, 64 Tensor cores Volta architecture GPU with 8-core Carmel Armv8.2 CPU, and 32GB of RAM. All evaluation results are reported on 10W, 15W, and 30W power modes for Jetson Nano, Jetson TX2, and Jetson AGX Xavier, respectively.

4.3. Evaluation Metrics

For the polyp detection task, we use evaluation metrics to measure the performance of our model, following the previous methods [14, 27, 36, 38].

1) **Floating-point Operations Per Second (FLOPs)** measures the computational cost of a model by counting the number of floating-point operations during inference.

2) **Parameters (Params)** refer to the trainable weights and biases in a model. The number of parameters indicates the model's capacity and complexity.

3) **Frames Per Second (FPS)** measures the speed of the model, indicating how many images the model can process.

4) **Precision** is a metric that measures the accuracy of positive predictions made by the model.

5) **Recall** measures the ability of the model to correctly detect positive instances.

6) **F1 score** is the harmonic mean of Precision and Recall. It provides a balanced measure of the model's performance by considering both Precision and Recall.

7) **mean Average Precision (mAP)** measures the accuracy of object localization and detection. mAP25 and mAP50 represent the mAP at an Intersection over Union threshold of 0.25 and 0.5, respectively. mAP50:95 represents the mAP across a range of IoU thresholds from 0.5 to 0.95.

4.4. Implementation Details

For our experiments, we primarily adopt the settings from YOLOv5 [17] and YOLOv8 [18]. The YOLOv3 model was trained within the YOLOv8 framework. However, we made some modifications to the settings. We changed the learning rate to 0.001. When training on the combined datasets, we set the batch size to 128 and the number of epochs was set to 100. When training on each of the SUN, KUMC, and Kvasir datasets, we set the

Size	Model	FLOPs(G) ↓	Params ↓	FPS ↑			Metrics ↑					
				Nano	TX2	Xavier	Precision	Recall	F1	mAP25	mAP50	mAP50:95
320 × 320	YOLOv5s	4.0	7.0M	47.3	124.5	351.1	**0.9295**	0.8472	0.8863	**0.9207**	**0.8800**	**0.5464**
	YOLOv5s-TST	**2.5** (-37.5%)	**3.7M** (-46.9%)	**64.8** (+37.0%)	**162.4** (+30.4%)	**437.5** (+24.6%)	0.9234	**0.8557**	**0.8881**	0.9186	0.8773	0.5243
	YOLOv5m	12.1	20.9M	18.6	53.9	177.0	0.9302	0.8329	0.8786	0.9082	0.8712	0.5440
	YOLOv5m-TST	**7.6** (-37.2%)	**11.0M** (-47.2%)	**26.0** (+39.8%)	**76.0** (+41.0%)	**241.7** (+36.6%)	**0.9336**	**0.8627**	**0.8967**	**0.9272**	**0.8902**	**0.5446**
	YOLOv5l	27.1	46.1M	9.9	29.1	119.9	0.9332	0.8449	0.8867	0.9156	0.8827	**0.5589**
	YOLOv5l-TST	**17.0** (-37.3%)	**24.4M** (-47.2%)	**15.2** (+53.5%)	**43.2** (+48.5%)	**167.9** (+40.0%)	**0.9344**	**0.8587**	**0.8947**	**0.9293**	**0.8902**	0.5518
384 × 384	YOLOv5s	5.7	7.0M	34.3	91.0	290.5	**0.9357**	**0.8533**	**0.8923**	**0.9271**	**0.8889**	**0.5532**
	YOLOv5s-TST	**3.7** (-35.1%)	**3.7M** (-46.9%)	**42.4** (+23.6%)	**117.7** (+29.3%)	**357.7** (+23.1%)	0.9203	0.8447	0.8807	0.9135	0.8745	0.5445
	YOLOv5m	17.4	20.9M	14.3	38.9	147.6	**0.9404**	0.8375	0.8857	0.9118	0.8779	**0.5520**
	YOLOv5m-TST	**11.0** (-36.8%)	**11.0M** (-47.2%)	**20.0** (+39.9%)	**55.1** (+41.6%)	**195.2** (+32.2%)	0.9256	**0.8662**	**0.8947**	**0.9274**	**0.8917**	0.5508
	YOLOv5l	39	46.1M	7.7	21.8	87.7	0.9359	0.8473	0.8893	0.9180	0.8824	**0.5629**
	YOLOv5l-TST	**24.5** (-37.2%)	**24.4M** (-47.2%)	**12.1** (+57.1%)	**31.6** (+45.0%)	**129.1** (+47.2%)	**0.9364**	**0.8711**	**0.9024**	**0.9355**	**0.8972**	0.5599
480 × 480	YOLOv5s	9.0	7.0M	23.1	46.7	226.3	**0.9338**	**0.8542**	**0.8921**	**0.9266**	**0.8918**	**0.5804**
	YOLOv5s-TST	**5.7** (-36.7%)	**3.7M** (-46.9%)	**30.8** (+33.3%)	**82.5** (+76.7%)	**260.9** (+15.3%)	0.9098	0.8438	0.8753	0.9138	0.8711	0.5414
	YOLOv5m	27.1	20.9M	9.9	28.2	105.5	**0.9381**	0.8447	0.8889	0.9215	0.8849	0.5590
	YOLOv5m-TST	**17.1** (-36.9%)	**11.0M** (-47.2%)	**14.2** (+43.4%)	**40.3** (+42.9%)	**138.3** (+31.1%)	0.9352	**0.8603**	**0.8958**	**0.9265**	**0.8937**	**0.5602**
	YOLOv5l	60.9	46.1M	5.6	15.7	69.3	**0.9398**	0.8507	0.8927	0.9220	0.8862	**0.5690**
	YOLOv5l-TST	**38.3** (-37.1%)	**24.4M** (-47.2%)	**7.6** (+35.7%)	**22.8** (+45.2%)	**95.0** (+37.1%)	0.9293	**0.8704**	**0.8988**	**0.9324**	**0.8958**	0.5632

Table 1. Comparison of the performance between our proposed method and various sizes of the YOLOv5 [17] models. Each model is evaluated on the combined datasets of SUN, KUMC, and Kvasir. The best scores are bold-faced.

Model	FLOPs(G) ↓	Params ↓	FPS ↑		
			Nano	TX2	Xavier
YOLOv5m	12.1	20.9M	18.6	53.9	177.0
ENDOMIND	12.9	21.5M	15.9	48.7	160.0
YOLOv5m-Tiny	11.0	15.4M	21.1	57.5	210.7
YOLOv5m-TST	**7.6**	**11.0M**	**26.0**	**76.0**	**241.7**

Model	Dataset	Metrics					
		Precision	Recall	F1	mAP25	mAP50	mAP50:95
YOLOv5m	SUN	0.9183	0.8204	0.8663	0.8981	0.8633	0.5196
ENDOMIND		0.9082	0.8194	0.8612	0.8925	0.8458	0.4974
YOLOv5m-Tiny		0.9064	0.8336	0.8683	0.9084	0.8713	**0.5200**
YOLOv5m-TST		**0.9223**	**0.8409**	**0.8794**	**0.9157**	**0.8736**	0.5089
YOLOv5m	KUMC	0.9236	0.8467	0.8835	0.9187	0.8835	0.5780
ENDOMIND		0.9286	0.8237	0.8727	0.9026	0.8609	0.5566
YOLOv5m-Tiny		0.9191	0.8464	0.8809	0.9188	0.8808	**0.5879**
YOLOv5m-TST		**0.9387**	**0.8554**	**0.8951**	**0.9286**	**0.8881**	0.5728
YOLOv5m	Kvasir	0.9323	0.8906	0.9109	0.9370	0.8988	0.6631
ENDOMIND		0.9332	0.8873	0.9095	0.9410	0.9048	0.6729
YOLOv5m-Tiny		0.9072	0.8889	0.8977	0.9374	0.9008	0.6672
YOLOv5m-TST		**0.9369**	**0.8915**	**0.9134**	**0.9428**	**0.9150**	**0.6855**

Table 2. Comparison on SUN, KUMC, and Kvasir datasets with 320 × 320. For each dataset, comparisons among YOLOv5m, YOLOv5m-TST, YOLOv5m-ENDOMIND (ENDOMIND) [23], and modified Polyp-YOLOv5-Tiny (YOLOv5m-Tiny) [36] are presented. The best scores are bold-faced.

batch size to 64 and the number of epochs was set to 90, 120, and 3000, respectively. In addition, we compare the performance of our models using input image sizes of 320, 384, and 480, considering the impact of different image sizes on the model's performance. All models were trained from scratch using NVIDIA RTX™ A6000 GPU. We used different sizes of the YOLOv5 model, namely s, m, and l. In these models, the sizes of the channels $\{C_{l1}, C_{l2}, C_G\}$ in each feature map are $\{128, 256, 512\}$, $\{192, 384, 768\}$, $\{256, 512, 1024\}$, respectively. In the YOLOv8m model, the sizes of the channels in each feature map are $\{192, 384, 576\}$. In the YOLOv3 model, the sizes of the channels in each feature map are $\{256, 512, 1024\}$. The results in Tab. 1, Tab. 2, Tab. 3 and Tab. 5 are averaged from 5-fold cross-validation. In supplementary material, the full experimental results are presented.

4.5. Results

We hypothesized that the TST layer could serve as a suitable replacement for the CNN-based neck, both in terms of devising a lightweight model and potentially enhancing feature fusion. The following experiments demonstrate the validity of our hypothesis. We set the YOLOv5m model as our baseline model and conducted experiments accordingly.

Tab. 1 presents a performance comparison of the TST layer based on the size of the YOLOv5 model. When comparing our YOLOv5m-TST with YOLOv5m, we observed a reduction of 37.2% in FLOPs and a decrease of 47.2% in the number of parameters. This trend of reduction is consistent, yielding similar results regardless of the model size. We achieved an average reduction of 47.1% in the number of parameters regardless of the model size. Furthermore, we confirmed an average reduction of 36.9% in FLOPs, even when the input image size changed. In terms of performance metrics, TST layer showed minor performance drop or even better performance. The TST layer has significantly contributed to performance improvements when deployed on edge devices.

For real-time object detection, achieving at least 30 FPS is considered essential [25]. In cases where previous models could not meet this FPS requirement due to their larger size, our more compact and efficient model successfully attained an FPS above 30. Specifically, with an input size of 384, YOLOv5l-TST leaped, increasing FPS from 21.8 to 31.6 at TX2. Furthermore, the performance is increased by approximately 40.36%, 44.51%, and 31.91% on the NVIDIA Jetson Nano, TX2, and Xavier, respectively. These improvements emphasize the effectiveness of replacing the neck part with the TST layer, particularly in scenarios demanding real-time detection, by not only meeting but surpassing the 30 FPS threshold across varying image sizes.

Size	Model	FLOPs(G) ↓	Params ↓	FPS ↑			Metrics ↑					
				Nano	TX2	Xavier	Precision	Recall	F1	mAP25	mAP50	mAP50:95
320 × 320	YOLOv3	70.7	103.7M	4.9	13.8	68.7	0.9342	0.8795	0.9056	0.9438	0.9157	**0.6170**
	YOLOv3-TST	**55.3** (-21.8%)	**75.4M** (-27.3%)	**5.9** (+20.4%)	**17.1** (+23.9%)	**83.8** (+21.9%)	**0.9347**	**0.8873**	**0.9102**	**0.9500**	**0.9210**	0.6093
	YOLOv8m	19.8	25.9M	12.7	37.1	130.1	**0.9365**	0.8651	0.8989	0.9363	0.9077	**0.6119**
	YOLOv8m-TST	**13.5** (-31.8%)	**15.0M** (-42.1%)	**18.1** (+42.5%)	**51.8** (+39.6%)	**173.2** (+33.1%)	0.9316	**0.8775**	**0.9036**	**0.9460**	**0.9139**	0.5991
	YOLOv5m	12.1	20.9M	18.6	53.9	177	0.9302	0.8329	0.8786	0.9082	0.8712	0.5440
	YOLOv5m-TST	**7.6** (-37.2%)	**11.0M** (-47.2%)	**26.0** (+39.8%)	**76.0** (+41.0%)	**241.7** (+36.6%)	**0.9336**	**0.8627**	**0.8967**	**0.9272**	**0.8902**	**0.5446**
384 × 384	YOLOv3	101.9	103.7M	X	11.2	48.0	**0.9398**	0.8890	**0.9133**	0.9491	0.9197	**0.6210**
	YOLOv3-TST	**79.7** (-21.8%)	**75.4M** (-27.3%)	**5.0**	**12.4** (+10.7%)	**52.6** (+9.6%)	0.9316	**0.8902**	0.9101	**0.9499**	**0.9204**	0.6127
	YOLOv8m	28.5	25.9M	10.4	29.7	111.9	0.9351	0.8627	0.8971	0.9363	0.9069	**0.6161**
	YOLOv8m-TST	**19.5** (-31.6%)	**15.0M** (-42.1%)	**15.0** (+44.2%)	**40.1** (+35.0%)	**143.4** (+28.2%)	**0.9381**	**0.8848**	**0.9104**	**0.9507**	**0.9197**	0.6069
	YOLOv5m	17.4	20.9M	14.3	38.9	147.6	**0.9404**	0.8375	0.8857	0.9118	0.8779	**0.5520**
	YOLOv5m-TST	**11.0** (-36.8%)	**11.0M** (-47.2%)	**20.0** (+39.9%)	**55.1** (+41.6%)	**195.2** (+32.2%)	0.9256	**0.8662**	**0.8947**	**0.9274**	**0.8917**	0.5508
480 × 480	YOLOv3	159.2	103.7M	X	7.6	38.2	**0.9434**	0.8803	0.9105	0.9480	0.9229	**0.6296**
	YOLOv3-TST	**124.5** (-21.8%)	**75.4M** (-27.3%)	**3.4**	**9.2** (+21.1%)	**45.0** (+15.1%)	0.9342	**0.8954**	**0.9143**	**0.9528**	**0.9257**	0.6123
	YOLOv8m	44.5	25.9M	7.5	20.4	76.1	**0.9398**	0.8683	0.9024	0.9413	0.9142	**0.6189**
	YOLOv8m-TST	**30.5** (-31.5%)	**15.0M** (-42.1%)	**9.6** (+28.0%)	**27.1** (+32.8%)	**106.7** (+40.2%)	0.9327	**0.8847**	**0.9078**	**0.9481**	**0.9180**	0.6097
	YOLOv5m	27.1	20.9M	9.9	28.2	105.5	**0.9381**	0.8447	0.8889	0.9215	0.8849	0.5590
	YOLOv5m-TST	**17.1** (-36.9%)	**11.0M** (-47.2%)	**14.2** (+43.4%)	**40.3** (+42.9%)	**138.3** (+31.1%)	0.9352	**0.8603**	**0.8958**	**0.9265**	**0.8937**	**0.5602**

Table 3. Comparison of the performance between our proposed method applied to YOLOv3 [43] and YOLOv8m [18] for examining the compatibility of the model. Each model is evaluated on the combined datasets of SUN, KUMC, and Kvasir. The best scores are bold-faced.

Shape	# Frames	Model	Precision	Recall	F1	mAP25	mAP50	mAP50:95
IIa	939	YOLOv5m	0.9492	0.8168	0.8781	0.8805	0.8629	**0.4838**
		YOLOv5m-TST	**0.9627**	**0.8573**	**0.9070**	**0.9284**	**0.8959**	0.4818
Ip	264	YOLOv5m	0.9994	**0.9962**	**0.9978**	**0.9950**	**0.9950**	**0.8599**
		YOLOv5m-TST	**1.0000**	0.9917	0.9958	**0.9950**	**0.9950**	0.8514
Is	7693	YOLOv5m	**0.9700**	0.9122	0.9402	0.9626	0.9550	**0.5929**
		YOLOv5m-TST	0.9671	**0.9410**	**0.9539**	**0.9797**	**0.9724**	0.5881
Isp	799	YOLOv5m	0.9674	0.8473	0.9034	0.9248	0.9108	0.6254
		YOLOv5m-TST	**0.9692**	**0.9050**	**0.9360**	**0.9587**	**0.9514**	**0.6609**

Table 4. Comparison between YOLOv5m and YOLOv5m-TST based on polyp shape in the SUN test dataset with 320 × 320 resolution. The best scores are bold-faced.

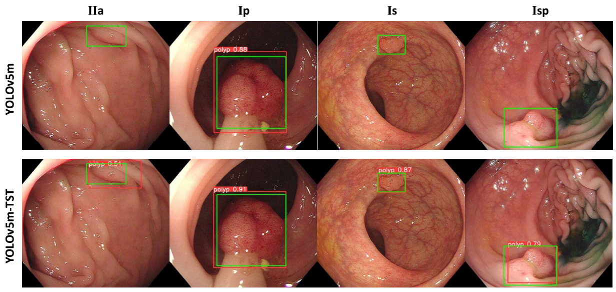

Figure 3. Detection results of YOLOv5m and YOLOv5m-TST on the SUN test set based on polyp shapes. The red and green boxes signify the prediction results and ground truth, respectively.

Tab. 2 presents a comparison of our baseline models, YOLOv5m-TST with YOLOv5m, YOLOv5-ENDOMIND (shortly ENDOMIND) [23], and modified Polyp-YOLOv5-Tiny (shortly YOLOv5m-Tiny) [36] on each dataset – SUN, KUMC, and Kvasir. The ENDOMIND is an advanced version of the baseline YOLOv5 model. The original Polyp-YOLOv5-Tiny utilized YOLOv5s. However, for the purpose of conducting a fair comparison in our experiments, we re-implemented it to align with the YOLOv5m model and conducted our experiments accordingly. YOLOv5m-Tiny, derived from the original YOLOv5 architecture, removes the large-object detection part of the head. For each dataset, YOLOv5m-TST outperforms other models except mAP50:95. Therefore, our YOLOv5m-TST demonstrates a lightweight yet generalized model performance.

In Tab. 3, to check compatibility and compare our method with the baseline YOLO models, we experimented with YOLOv3 and YOLOv8. The YOLOv3 and YOLOv5 models were used as baseline models in previous studies [20, 33, 34, 36, 38, 52]. In the YOLOv3 architecture, the FPS of YOLOv3-TST on TX2 and Xavier increases by approximately 18.6% and 15.5%, respectively. In Nano, the YOLOv3-TST operates successfully, but for YOLOv3, the model does not work with image sizes 384 and 480. In the YOLOv8m, the FPS of YOLOv8m-TST on Nano, TX2, and Xavier increases by approximately 38.23%, 35.8%, and 33.83%, respectively. Furthermore, in the cases of YOLOv3 and YOLOv8, the model with the TST demonstrates reductions in computational cost, leading to 21.8% and 31.6% reduction in FLOPs, along with a decrease in parameters by 27.3% and 42.1%, respectively. Performance metrics, such as Precision, F1, and mAP50:95 are slightly dropped. Notwithstanding these minor decreases, the YOLOv3-TST and YOLOv8m-TST demonstrated slightly higher or nearly similar performance in other metrics. Through our method, we could leverage YOLOv3 architecture on Nano. Moreover, the TST layer demonstrates efficiency and adaptability even in newer architectures like YOLOv8m.

Tab. 4 presents the evaluation based on the type of polyp using SUN test set. In SUN dataset, polyps are categorized into four shapes according to the Paris Classification [24, 35]: IIa, Ip, Is, and Isp. We compare the performance of YOLOv5m and YOLOv5m-TST based on these shapes.

Size	Key Dim.	FLOPs(G)	Params	FPS			Metrics					
				Nano	TX2	Xavier	Precision	Recall	F1	mAP25	mAP50	mAP50:95
320 × 320	4	7.6 (-37.2%)	11.0M (-47.2%)	26.0 (+39.8%)	76.0 (+41.0%)	241.7 (+36.6%)	0.9336	0.8627	0.8967	0.9272	0.8902	0.5446
	64	8.5 (-29.8%)	15.4M (-26.3%)	23.5 (+26.3%)	66.8 (+23.9%)	217.1 (+22.7%)	0.9364	**0.8686**	**0.9011**	0.9336	*0.9006*	**0.5602**
	128	9.4 (-22.3%)	20.1M (-3.8%)	21.9 (+17.7%)	60.5 (+12.2%)	189.4 (+7.0%)	*0.9370*	0.8645	0.8991	*0.9341*	0.8990	0.5575
	256	11.3 (-6.6%)	29.4M (+40.7%)	18.4 (-1.1%)	47.5 (-11.9%)	137.1 (-22.5%)	**0.9381**	*0.8659*	*0.9004*	**0.9355**	**0.9014**	*0.5601*
	YOLOv5m	12.1	20.9M	18.6	53.9	177	0.9302	0.8329	0.8786	0.9082	0.8712	0.5440

Table 5. Comparison of the performance between our proposed method with varying key dimension sizes and the original YOLOv5m model. Result of models evaluated on the combined combined datasets of SUN, KUMC, and Kvasir. The best, second-best, and third-best scores are highlighted in bold, italics, and underlined, respectively.

Across most shapes, YOLOv5m-TST shows better metric performance. However, except for Isp, the YOLOv5m model shows better mAP50:95 performance. Fig. 3 shows examples of each shape along with the detection results for each model. There are various cases where the YOLOv5m-TST model detects polyps but YOLOv5m misses.

mAP25 and mAP50 of our models generally show comparable or slightly better performance than those of existing models. Furthermore, when evaluating Precision, Recall, and F1 Score, our model shows similar performance to the existing models in most cases, with only a few exceptions. For real-time polyp detection, the primary goal is to help endoscopists not miss polyps during the examination. Early-stage polyps might be extremely small or have irregular shapes. So, the ability to detect small or irregularly shaped polyps is vital for early detection and diagnosis, facilitating the prompt delivery of proper treatment to patients. Although determining the exact location and size of polyps would be desirable, our main focus is on alerting the endoscopist to the presence of polyps. From our experiments, YOLOv5-TST demonstrates proficiency in detecting polyps, but it might occasionally struggle to accurately localize polyps compared to the YOLOv5. It can be considered a trade-off between computational cost and accurate prediction. YOLOv5 excels in mAP50:95 performance, yet it requires a more computational cost than YOLOv5-TST. On the other hand, YOLOv5-TST effectively reduces the computational cost and achieves better mAP25 and mAP50. Thus, to assist endoscopists, YOLOv5-TST can be desirable in resource-constrained devices because it can effectively reduce computational costs while showing high performance at mAP25 and mAP50.

4.6. Ablation Study

We hypothesize that if we configure our model to have a similar number of FLOPs as the original YOLO model, we would observe improvements across various evaluation metrics. Accordingly, we experimented with YOLOv5m-TST by adjusting the key dimension, which is initially set to 4, and aims for the number of FLOPs that is roughly equivalent to that of YOLOv5m. By setting the key dimension to 256, we could achieve comparable FLOPs to the conventional YOLO model. We evaluate the performance of the TST layer by increasing the key dimension to 64, 128, and 256. Although this increases the number of parameters, as can be seen in Tab. 5, our model outperforms the standard YOLOv5m across most metrics. As the number of key dimensions increases, the overall performances are improved. Particularly, when the key dimensions are set to 256, the model demonstrates the best performance in mAP25 and mAP50. We aim to design a model that is lightweight yet does not compromise significantly on performance. So, we devise a model with key dimensions set to 4. In real-time polyp detection, latency is a critical consideration. Setting key dimensions to 4 not only results in superior FPS performance compared to 256 but also does not have a significant compromise in evaluation metrics performance. This observation suggests that the TST layer is capable of more efficiently fusing features at a lower dimension, and consequently enhances the performance of the model with less computational demand.

5. Conclusion

In this paper, we propose a novel and efficient model for polyp detection by using a lightweight Transformer within the neck of the YOLO framework. Specifically, we replace the CNN-based neck with a Transformer-based TST layer. The TST layer employs global token sharing for effective feature fusion. The TST layer reduces both the number of parameters and FLOPs, ultimately leading to significant improvements in FPS without substantial performance drops. Our experiments show consistent performance improvement across various YOLOv5 model sizes and input sizes. We also confirmed a similar trend with the recently proposed YOLOv8m. For CRC prevention, our model makes it feasible for deployment in medically underserved regions, serving the public interest by potentially expanding the reach of critical diagnostic tools.

Acknowledgements. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub)

References

[1] Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. Texture-based polyp detection in colonoscopy. In Bildverarbeitung für die Medizin 2009: Algorithmen—Systeme—Anwendungen Proceedings des Workshops vom 22. bis 25. März 2009 in Heidelberg, pages 346–350. Springer, 2009. 2

[2] Melina Arnold, Mónica S Sierra, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. Global patterns and trends in colorectal cancer incidence and mortality. Gut, 66(4):683–691, 2017. 1

[3] Jorge Bernal, Javier Sánchez, and Fernando Vilarino. Towards automatic polyp detection with a polyp appearance model. Pattern Recognition, 45(9):3166–3182, 2012. 2

[4] Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sanchez, Bogdan J Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjørn Rustad, Ilangko Balasingham, et al. Comparative validation of polyp detection methods in video colonoscopy: results from the miccai 2015 endoscopic vision challenge. IEEE transactions on medical imaging, 36(6):1231–1249, 2017. 2

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020. 3

[6] John H Bond, Practice Parameters Committee of the American College of Gastroenterology, et al. Polyp guideline: diagnosis, treatment, and surveillance for patients with colorectal polyps. Official journal of the American College of Gastroenterology— ACG, 95(11):3053–3063, 2000. 1

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pages 213–229. Springer, 2020. 3

[8] Douglas A Corley, Christopher D Jensen, Amy R Marks, Wei K Zhao, Jeffrey K Lee, Chyke A Doubeni, Ann G Zauber, Jolanda de Boer, Bruce H Fireman, Joanne E Schottinger, et al. Adenoma detection rate and risk of colorectal cancer and death. New england journal of medicine, 370(14):1298–1306, 2014. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 2, 3

[10] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In Proceedings of the IEEE/CVF international conference on computer vision, pages 12259–12269, 2021. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9):1904–1916, 2015. 3

[12] Sae Hwang, JungHwan Oh, Wallapak Tavanapong, Johnny Wong, and Piet C De Groen. Polyp detection in colonoscopy video using elliptical shape feature. In 2007 IEEE International Conference on Image Processing, volume 2, pages II–465. IEEE, 2007. 2

[13] Dimitris K Iakovidis, Dimitris E Maroulis, and Stavros A Karkanis. An intelligent system for automatic detection of gastrointestinal adenomas in video endoscopy. Computers in biology and medicine, 36(10):1084–1103, 2006. 2

[14] Debesh Jha, Sharib Ali, Nikhil Kumar Tomar, Håvard D Johansen, Dag Johansen, Jens Rittscher, Michael A Riegler, and Pål Halvorsen. Real-time polyp detection, localization and segmentation in colonoscopy using deep learning. Ieee Access, 9:40496–40510, 2021. 2, 3, 5

[15] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In MultiMedia Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II 26, pages 451–462. Springer, 2020. 5

[16] Xiao Jia, Xiaochun Mai, Yi Cui, Yixuan Yuan, Xiaohan Xing, Hyunseok Seo, Lei Xing, and Max Q-H Meng. Automatic polyp recognition in colonoscopy images using deep learning and two-stage pyramidal feature prediction. IEEE Transactions on Automation Science and Engineering, 17(3):1570–1584, 2020. 2, 3

[17] Glenn Jocher. Ultralytics yolov5, 2020. 2, 3, 5, 6

[18] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. 5, 7

[19] Dervis Karaboga et al. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer . . . , 2005. 3

[20] Ahmet Karaman, Ishak Pacal, Alper Basturk, Bahriye Akay, Ufuk Nalbantoglu, Seymanur Coskun, Omur Sahin, and Dervis Karaboga. Robust real-time polyp detection system design based on yolo algorithms by optimizing activation functions and hyper-parameters with artificial bee colony (abc). Expert Systems with Applications, 221:119741, 2023. 2, 3, 7

[21] Stavros A Karkanis, Dimitrios K Iakovidis, Dimitrios E Maroulis, Dimitris A. Karras, and M Tzivras. Computer-aided tumor detection in endoscopic video using color wavelet features. IEEE transactions on information technology in biomedicine, 7(3):141–152, 2003. 2

[22] Nam Hee Kim, Yoon Suk Jung, Woo Shin Jeong, Hyo-Joon Yang, Soo-Kyung Park, Kyuyong Choi, and Dong Il Park. Miss rate of colorectal neoplastic polyps and risk factors for missed polyps in consecutive colonoscopies. Intestinal research, 15(3):411, 2017. 2

[23] Adrian Krenzer, Michael Banck, Kevin Makowski, Amar Hekalo, Daniel Fitting, Joel Troya, Boban Sudarevic, Wolfgang G Zoller, Alexander Hann, and Frank Puppe. A real-time polyp-detection system with clinical application in colonoscopy using deep convolutional neural networks. Journal of Imaging, 9(2):26, 2023. 6, 7

[24] Adrian Krenzer, Stefan Heil, Daniel Fitting, Safa Matti, Wolfram G Zoller, Alexander Hann, and Frank Puppe. Au-

tomated classification of polyps using deep learning architectures and few-shot learning. BMC Medical Imaging, 23(1):59, 2023. 7

[25] Dong-Jae Lee, Jae Young Lee, Hyunguk Shon, Eojindl Yi, Yeong-Hun Park, Sung-Sik Cho, and Junmo Kim. Lightweight monocular depth estimation via token-sharing transformer. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 4895–4901, 2023. 2, 3, 4, 6

[26] Ji Young Lee, Jinhoon Jeong, Eun Mi Song, Chunae Ha, Hyo Jeong Lee, Ja Eun Koo, Dong-Hoon Yang, Namkug Kim, and Jeong-Sik Byeon. Real-time detection of colon polyps during colonoscopy using deep learning: systematic validation with four independent datasets. Scientific reports, 10(1):8379, 2020. 2, 3

[27] Kaidong Li, Mohammad I Fathan, Krushi Patel, Tianxiao Zhang, Cuncong Zhong, Ajay Bansal, Amit Rastogi, Jean S Wang, and Guanghui Wang. Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations. Plos one, 16(8):e0255809, 2021. 5

[28] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. Advances in Neural Information Processing Systems, 35:12934–12949, 2022. 2, 3

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017. 3

[30] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8759–8768, 2018.

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision, pages 10012–10022, 2021. 3

[32] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178, 2021. 2, 3

[33] Masashi Misawa, Shin-ei Kudo, Yuichi Mori, Kinichi Hotta, Kazuo Ohtsuka, Takahisa Matsuda, Shoichi Saito, Toyoki Kudo, Toshiyuki Baba, Fumio Ishida, et al. Development of a computer-aided detection system for colonoscopy and a publicly accessible large colonoscopy video database (with video). Gastrointestinal endoscopy, 93(4):960–967, 2021. 2, 3, 5, 7

[34] Alba Nogueira-Rodríguez, Rubén Domínguez-Carbajales, Fernando Campos-Tato, Jesús Herrero, Manuel Puga, David Remedios, Laura Rivas, Eloy Sánchez, Agueda Iglesias, Joaquín Cubiella, et al. Real-time polyp detection model using convolutional neural networks. Neural Computing and Applications, 34(13):10375–10396, 2022. 2, 3, 7

[35] null Endoscopic Classification Review Group et al. Update on the paris classification of superficial neoplastic lesions in the digestive tract. Endoscopy, 37(06):570–578, 2005. 7

[36] Shimin Ou, Yixing Gao, Zebin Zhang, and Chenjian Shi. Polyp-yolov5-tiny: A lightweight model for real-time polyp detection. In 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), volume 2, pages 1106–1111. IEEE, 2021. 2, 3, 5, 6, 7

[37] Ishak Pacal and Dervis Karaboga. A robust real-time deep learning based automatic polyp detection system. Computers in Biology and Medicine, 134:104519, 2021. 2, 3

[38] Ishak Pacal, Ahmet Karaman, Dervis Karaboga, Bahriye Akay, Alper Basturk, Ufuk Nalbantoglu, and Seymanur Coskun. An efficient real-time colonic polyp detection with yolo algorithms trained by using negative samples and large datasets. Computers in biology and medicine, 141:105031, 2022. 2, 3, 5, 7

[39] Hemin Ali Qadir, Ilangko Balasingham, Johannes Solhusvik, Jacob Bergsland, Lars Aabakken, and Younghak Shin. Improving automatic polyp detection using cnn by exploiting temporal dependency in colonoscopy video. IEEE journal of biomedical and health informatics, 24(1):180–193, 2019. 2, 3

[40] Hemin Ali Qadir, Younghak Shin, Johannes Solhusvik, Jacob Bergsland, Lars Aabakken, and Ilangko Balasingham. Toward real-time polyp detection using fully cnns for 2d gaussian shapes prediction. Medical Image Analysis, 68:101897, 2021. 2, 3

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016. 2, 3

[42] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017. 3

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018. 3, 7

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 2015. 3

[45] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510–4520, 2018. 3

[46] Younghak Shin, Hemin Ali Qadir, Lars Aabakken, Jacob Bergsland, and Ilangko Balasingham. Automatic colon polyp detection using region based deep cnn and post learning approaches. IEEE Access, 6:40950–40962, 2018. 2

[47] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. IEEE transactions on medical imaging, 35(2):630–644, 2015. 2

[48] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10781–10790, 2020. 3

[49] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In International conference on machine learning, pages 10347–10357. PMLR, 2021. 3

[50] Gregor Urban, Priyam Tripathi, Talal Alkayali, Mohit Mittal, Farid Jalali, William Karnes, and Pierre Baldi. Deep learning localizes and identifies polyps in real time with 96% accuracy in screening colonoscopy. Gastroenterology, 155(4):1069–1078, 2018. 2, 3

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017. 2, 3

[52] Jingjing Wan, Bolun Chen, and Yongtao Yu. Polyp detection from colorectum images by using attentive yolov5. Diagnostics, 11(12):2264, 2021. 2, 3, 7

[53] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pages 390–391, 2020. 3

[54] Xudong Wang, Li Lyna Zhang, Yang Wang, and Mao Yang. Towards efficient vision transformer inference: a first study of transformers on mobile devices. In Proceedings of the 23rd Annual International Workshop on Mobile Computing Systems and Applications, pages 1–7, 2022.

[55] WHO. Cancer: https://www.who.int/news-room/fact-sheets/detail/cancer. Accessed:, Jun. 2023. 1

[56] Chr Williams and RH Teague. Colonoscopy. Gut, 14(12):990, 1973. 1

[57] Sidney J Winawer, Ann G Zauber, May Nah Ho, Michael J O'brien, Leonard S Gottlieb, Stephen S Sternberg, Jerome D Waye, Melvin Schapiro, John H Bond, Joel F Panish, et al. Prevention of colorectal cancer by colonoscopic polypectomy. New England Journal of Medicine, 329(27):1977–1981, 1993. 1

[58] Ann G Zauber, Sidney J Winawer, Michael J O'Brien, Iris Lansdorp-Vogelaar, Marjolein van Ballegooijen, Benjamin F Hankey, Weiji Shi, John H Bond, Melvin Schapiro, Joel F Panish, et al. Colonoscopic polypectomy and long-term prevention of colorectal-cancer deaths. N Engl J Med, 366:687–696, 2012. 1

[59] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12083–12093, 2022. 3

[60] Xu Zhang, Fei Chen, Tao Yu, Jiye An, Zhengxing Huang, Jiquan Liu, Weiling Hu, Liangjing Wang, Huilong Duan, and Jianmin Si. Real-time gastric polyp detection using convolutional neural networks. PloS one, 14(3):e0214133, 2019. 2, 3

[61] Yali Zheng, Ruikai Zhang, Ruoxi Yu, Yuqi Jiang, Tony WC Mak, Sunny H Wong, James YW Lau, and Carmen CY Poon. Localisation of colorectal polyps by convolutional neural network features learnt from white light and narrow band endoscopic images of multiple databases. In 2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC), pages 4142–4145. IEEE, 2018. 2, 3

[62] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020. 3