

Attention-based UNet enabled Lightweight Image Semantic Communication System over Internet of Things

Guoxin Ma, Haonan Tong, Nuocheng Yang, and Changchuan Yin

Beijing Laboratory of Advanced Information Network, Beijing University of Posts and Telecommunications, Beijing, China
Emails: {guoxinma, hntong, yangnuocheng, and ccyin}@bupt.edu.cn

Abstract—This paper studies the problem of the lightweight image semantic communication system that is deployed on Internet of Things (IoT) devices. In the considered system model, devices must use semantic communication techniques to support user behavior recognition in ultimate video service with high data transmission efficiency. However, it is computationally expensive for IoT devices to deploy semantic codecs due to the complex calculation processes of deep learning (DL) based codec training and inference. To make it affordable for IoT devices to deploy semantic communication systems, we propose an attention-based UNet enabled lightweight image semantic communication (LSSC) system, which achieves low computational complexity and small model size. In particular, we first let the LSSC system train the codec at the edge server to reduce the training computation load on IoT devices. Then, we introduce the convolutional block attention module (CBAM) to extract the image semantic features and decrease the number of downsampling layers thus reducing the floating-point operations (FLOPs). Finally, we experimentally adjust the structure of the codec and find out the optimal number of downsampling layers. Simulation results show that the proposed LSSC system can reduce the semantic codec FLOPs by 14%, and reduce the model size by 55%, with a sacrifice of 3% accuracy, compared to the baseline. Moreover, the proposed scheme can achieve a higher transmission accuracy than the traditional communication scheme in the low channel signal-to-noise (SNR) region.

Keywords—Semantic communication, Image segmentation, Attention Mechanism, Internet of Things

I. INTRODUCTION

Semantic communication is a potentially vital technology to enable efficient data transmission for 6G intelligent connection, with the reduced amount of transmission data over ubiquitous intelligence applications [1]–[3]. To deploy a semantic communication system, Internet of Things (IoT) devices need Deep Learning (DL) based codecs for real-time semantic extraction and signal processing. However, training the codec on IoT devices and deploying the DL-based codec is computationally expensive, which can bring huge overhead for IoT devices with limited energy. Therefore, designing a lightweight semantic codec to reduce the costs of DL models on IoT devices is significant for deployment on IoT devices that support ubiquitous intelligence.

Existing works such as [4]–[6] have studied the efficient transmission of semantic communication for various data types

This work was supported in part by Beijing Natural Science Foundation under Grant L223027, the National Natural Science Foundation of China under Grant 61671086, China 973 Program under Grant 2009CB320407, in part by BUPT Excellent Ph.D. Students Foundation under Grant CX2021114, and China Scholarship Council.

over the Internet of Things. The authors in [4] proposed a DL-based semantic communication system, named DeepSC, for text transmission, which is more robust to channel variation compared with the traditional communication system. The work in [5] proposed a DL-enabled semantic communication system for speech signals (DeepSC-S) to improve the transmission efficiency significantly. Moreover, in [6], the authors proposed a federated learning-based audio semantic communication system over wireless networks, which can converge effectively compared to a traditional coding scheme. However, the works in [4]–[6] have left the field of image semantic communication, which is challenging due to the complexity of image data structure and various downstream tasks.

The prior arts [7], [8] have studied lightweight image semantic communication along with the development of image segmentation techniques. The work in [7] proposed a lightweight distributed semantic communication system by pruning the model redundancy and decreasing the weight resolution. The authors in [8] developed a semantic communication framework that reduces the computational complexity to run on edge devices with limited power. However, now that the works in [7], [8] introduce lightweight semantic codecs. When using a lightweight semantic codec, the model accuracy inescapably gets worse. To solve this, in [9], the authors designed a feature compression module based on the channel attention module that selects the most important features. Furthermore, the work in [10] proposed a combination of attention mechanism and dilated convolutions, to improve the segmentation performance of the image encoder. The authors in [11] proposed an efficient attention mechanism to learn the dynamic distribution of attention weights in both channel and spatial dimensions. However, most of these works [9]–[11] utilized the attention mechanism to improve the accuracy without considering the model size. Therefore, it is necessary to develop a lightweight semantic communication system considering both high accuracy and small model size.

The main contribution of this paper is a novel image semantic segmentation communication system that can be deployed on IoT devices with low floating-point operations (FLOPs) and small model size while ensuring high segmentation performance. Our key contributions include the following.

- We develop a lightweight image semantic communication (LSSC) system that transmits the trained semantic codec and semantic features between the edge server and IoT devices. The semantic codec consists of a semantic

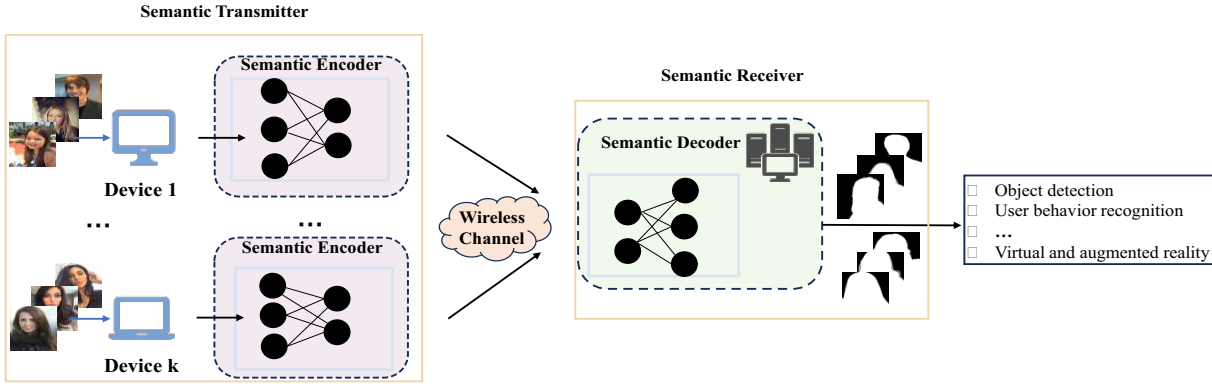


Fig. 1. Architecture of the LSSC system.

encoder that extracts the image data at the transmitter and a decoder that at the receiver decodes the image information, respectively.

- We design a lightweight attention-based UNet enabled network architecture with low computational complexity and small model size by decreasing the number of downsampling layers. Meanwhile, we introduce the convolutional block attention module (CBAM), which utilizes the attention module consisting of spatial and channel attention modules to evaluate the importance of individual pixels and channels for image segmentation, respectively.
- Simulation results show that the proposed LSSC system can reduce the FLOPs by 14% and model size by 55% with only a 3% mean Intersection over Union (mIoU) loss compared to traditional UNet. Moreover, our proposed model achieves higher mIoU than the baseline using the traditional method under different wireless channel conditions, especially in the low channel signal-to-noise (SNR) region.

The remainder of this paper is organized as follows. The system model and problem formulation are presented in Section II. In section III, we give a detailed description of the proposed lightweight semantic encoder and decoder. The simulation results are presented and analyzed in Section IV. The conclusion is drawn in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider an IoT network to deploy the LSSC system as shown in Fig. 1. The LSSC system consists of multiple devices and an edge server. To support wireless data transmission, IoT devices extract the image semantic information and transmit it. Then, the image semantic segmentation information obtained at the receiver will be used to support the downstream tasks, such as object detection, user behavior recognition, and virtual reality. Since the trained codec is deployed on IoT devices with finite memory, we must design a semantic segmentation codec that is lightweight enough to be deployed on devices. Moreover, since the computational capability of the device is limited, the semantic codec must be trained on the server with huge enough computational capability. As shown in Fig. 2, the edge server trains the codec and multiple devices transmit the

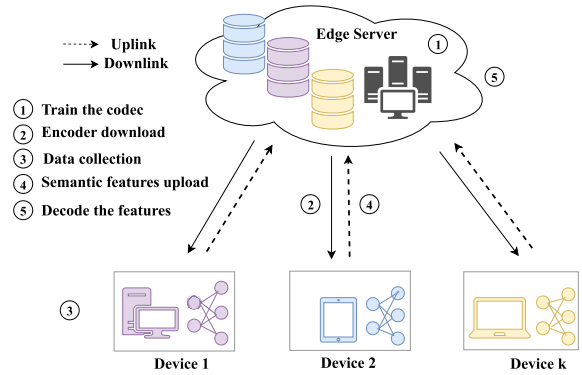


Fig. 2. The framework of semantic communication for networks.

image semantic features to the edge server. In this scenario, the overall training and inference procedure of the LSSC system is given as follows:

- 1) The edge server trains the semantic codec, which includes an encoder and a decoder.
- 2) The Edge server transmits the trained semantic encoder to each IoT device.
- 3) Each device collects the images and extracts image semantic features which are transmitted to the edge server.
- 4) The decoder at the edge server obtains the image segmentation which can be utilized for the downstream tasks.

A. LSSC Encoder

The LSSC encoder extracts the different levels of semantic features from the input image data. The processed image is put into the proposed encoder to extract the semantic features. Then, the semantic features are sent through the wireless channel.

We assume that the input image of the LSSC system is $S \in \mathbb{R}^{H \times W \times C}$, with H , W , and C being the image width, height, and the number of channels, respectively. We simplify the LSSC encoder parameter as α , thus, the relationship between the transmitted semantic features x and the input image S can be given by

$$x = R_{\alpha}(S), \quad (1)$$

where $x = R_{\alpha}(\cdot)$ indicates the function of the semantic encoder.

B. Wireless Channel

When transmitted over a wireless channel, the encoded semantic features will suffer channel fading and noise. We consider the condition of the wireless channel, where the received semantic features \mathbf{y} at the LSSC decoder can be given by

$$\mathbf{y} = h \cdot \mathbf{x} + \mathbf{n}, \quad (2)$$

where h represents the channel gain between the transmitter and the receiver, and $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I})$ is additive white Gaussian noise (AWGN) with variance σ_n^2 being noise variance and \mathbf{I} being identity matrix.

C. LSSC Decoder

The LSSC decoder is used to recover the received semantic feature \mathbf{y} . The decoded image segmentation is $\hat{\mathbf{S}}$. The decoder needs to get the image segmentation $\hat{\mathbf{S}}$ with the same height and width of the input image. $\hat{\mathbf{S}}$ classifies the pixels in \mathbf{S} into N categories with N being the number of object categories concerned in \mathbf{S} (portraits and background, etc.). To this end, the association between the output image segmentation $\hat{\mathbf{S}}$ and the received semantic features \mathbf{y} can be given by

$$\hat{\mathbf{S}} = R_\beta^{-1}(\mathbf{y}), \quad (3)$$

where $R_\beta^{-1}(\cdot)$ is the semantic decoder network with parameter β and the superscript represents the inverse function operation.

D. LSSC Objective

The objective of the LSSC system is to decode the image segmentation of the image containing portraits at the receiver. Since the LSSC system focuses on semantic-level image recovery, we use the mIoU metric rather than the bit error rate in traditional communications to evaluate the system performance, which is given by

$$mIoU = \frac{1}{N} \sum_{i=1}^N \frac{P \cap G}{P \cup G}, \quad (4)$$

where P is the set of pixel regions predicted by the LSSC decoder for a specific category object, and G is the actual set of pixel regions for this category object.

To compress the image semantic segmentation models for transmission and implementation on devices, we propose a lightweight image semantic segmentation model. Generally, the DL model training and inference run upon FLOPs, which means that the input and output require a large number of computation cycles. The number of FLOPs required by model feedforward measures the computation complexity of the model. During forward propagation in a model, operations such as convolution, pooling, BatchNorm, ReLU, and Up-sample, etc., are performed, with convolution being the most computationally intensive. Therefore, we mainly consider the decrease in convolution operation for low FLOPs. In order to deploy the efficient semantic codec on IoT devices, we need to reduce the FLOPs. The number of FLOPs in LSSC codec is defined as G , which can be given by

$$G = \sum_{l=1}^L l \times W_l \times H_l \times C_l^{\text{in}} \times C_l^{\text{out}} \times (K_l^2 + 1), \quad (5)$$

where W_l and H_l are the width and height of the input feature map of one downsampling or upsampling layer l , respectively. K_l is the kernel size. C_l^{in} and C_l^{out} are the number of channels of the input and output feature maps of one layer, respectively. $L = L_{en} + L_{de}$ is the total number of convolutional and deconvolutional layers, with L_{en} being the number of convolutional layers of LSSC encoder and L_{de} being the number of deconvolutional layers of LSSC decoder. Meanwhile, the performance of a codec improves as the number of parameters increases. However, high computational complexity and model size results in high energy consumption. Given the finite memory of devices, a model with a smaller size is favored for the advantages, including improved memory conservation, enhanced operational efficiency, and faster calculations. In the LSSC system, the model size of codec M is given by

$$M = \sum_{l=1}^L l \times C_l^{\text{in}} \times C_l^{\text{out}} \times K_l^2. \quad (6)$$

Moreover, in order to exploit the trade-off between the model accuracy and model complexity, we design a lightweight model for the semantic communication system. Assuming the LSSC system, we denote the segmentation metric achieved by the LSSC codec as Q , which is given by

$$Q = \lambda \cdot mIoU - \mu \cdot G - \nu \cdot M, \quad (7)$$

where λ , μ , and ν are coefficients that characterize the importance of accuracy, model computational complexity, and model size.

The objective of the LSSC system is to maximize the mIoU, while with lower FLOPs and smaller model size under the proposed scenario. Therefore, we formulate the following optimization problem. The expression can be given by

$$\max_{G, M} Q, \quad (8)$$

$$\text{s.t. } G \leq g, \quad (8a)$$

$$M \leq m, \quad (8b)$$

where the g and m are the FLOPs and model size of the baseline of the existing semantic codec. From (5), (6), and (7), we can see that the L , C_l^{in} , C_l^{out} and K_l determine the G and M , thus determining the optimization objective Q . In order to reduce the FLOPs and model size, we decrease the number of L to make the G and M smaller. Since compressing the semantic segmentation model makes it possible to run the DL model on IoT devices, it is necessary to design the architecture of the LSSC encoder and the LSSC decoder. To this end, we introduce an attention-based model to extract the image semantic information.

III. LSSC ENCODER AND DECODER DESIGN

To solve the problem (8), we design the architecture of the neural network to optimize the stated objective. We first propose an improved network architecture to extract image information with high efficiency. Then, we provide a detailed exposition of the LSSC encoder and decoder based on UNet,

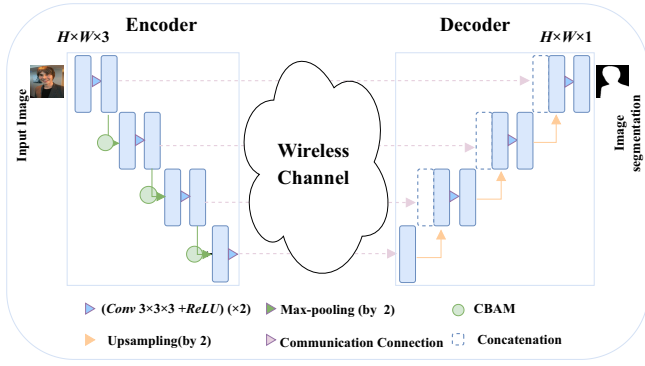


Fig. 3. UNet based Semantic Codec Architecture in LSSC system.

along with the improvements made to it and we utilize the CBAM module to improve the extraction performance. Subsequently, we introduce the training strategy of our proposed LSSC system.

A. UNet based Semantic Codec Architecture

For semantic segmentation tasks, the detailed high-level segmentation feature information is valuable. Based on the symmetric encoder-decoder structure of the UNet, with a cross-layer feature reconstruction module, the important high-level feature information can be preserved. Meanwhile, skip-connection can provide multi-scale and multi-level information for image segmentation, thus obtaining a more precise segmentation performance. Most importantly, UNet is better than simply using the codec framework, where much detailed information that is helpful is compressed and lost during the coding process [12]. Therefore, we determined to develop a lightweight model based on UNet.

Based on the UNet symmetric encoder-decoder structure, from the expressions (5) and (6), we can see that when the number of downsampling and upsampling operations decreases, it consequently influences the number of convolutional layers, resulting in changes to both FLOPs and model size. Since the computational capability of the IoT device is limited, UNet imposes a significant computational burden on IoT devices. Therefore, we need to reduce the computational complexity by ignoring some semantic features in order to lessen the burden on IoT devices. Furthermore, in order to address the issue of class ambiguity resulting from the reduction of high-level features, the CBAM module is employed in the semantic encoder, enabling adaptive recalibration of feature mappings.

The proposed CBAM-based network architecture is shown in Fig. 3. From Fig. 3, we can see that the proposed network architecture is composed of a semantic encoder, wireless channel, and semantic decoder, which extract semantic information and recover semantic features, respectively. To extract the important image data information in the downsampling process, the CBAM module is employed following each downsampling layer while ignoring useless features.

B. Module Enhancement Methods

Fig. 4(a) shows the CBAM module in the semantic encoder of the LSSC system. The CBAM module can assign different

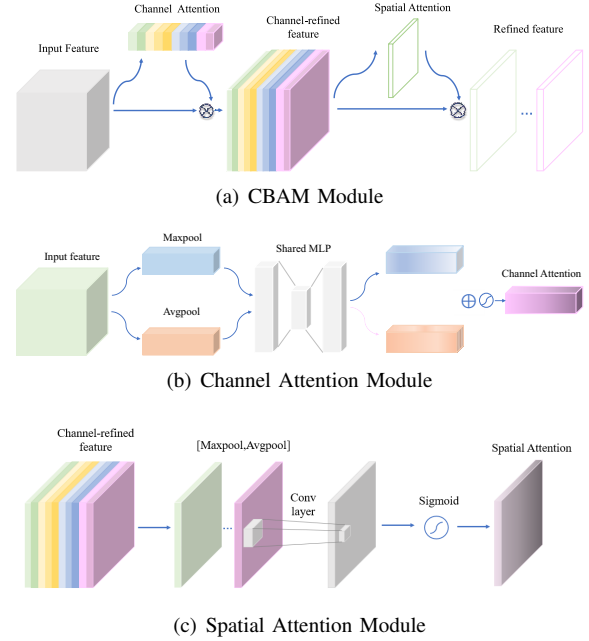


Fig. 4. Comparisons of different attention modules

weights to different channels or regions in space, thereby helping the codec to focus on extracting more important information. CBAM module is proposed as a lightweight attention module, which can infer attention maps along two separate dimensions, channel and space. Subsequently, we describe two dimensions of the CBAM module, the channel attention module, and the spatial attention module, respectively. Since CBAM is a lightweight general-purpose module, it can be seamlessly integrated into any network architecture with negligible overhead, and it can be end-to-end trainable [13].

The channel attention module focuses on which channels play an important role in determining the final inference results of the LSSC codec, i.e., it selects the channels that are decisive for prediction. As shown in Fig. 4(b), the input feature map $F \in \mathbb{R}^{H \times W \times C}$ is initially fed into both maximum pooling and average pooling operations along the channel dimension. The outputs of the pooling operations are then fed into a multilayer perceptron (MLP) layer, followed by element-wise summation and activation using a sigmoid function to generate the channel attention weight map $M_c \in \mathbb{R}^{H \times W \times C}$. The detailed process is described as

$$\begin{aligned}
 M_c(F) &= \sigma(MLP(Avgpool(F)) + MLP(Maxpool(F))) \\
 &= \sigma(W_1(W_0(F_{avg}^c) + W_0(F_{max}^c))),
 \end{aligned} \tag{9}$$

where σ indicates the sigmoid function, MLP represents the multilayer perceptron, and Avgpool and Maxpool denote the pooling operations. The channel attention weight map M_c will be used as the input of the spatial attention module. F_{avg}^c and F_{max}^c represent the feature after global avgpooling and maxpooling layer, W_0 and W_1 denote the two-layer weights of the MLP. The channel attention weight map M_c will be used as the input of the spatial attention module.

TABLE I
SIMULATION PARAMETERS

Module	Layer Name	Block	Value
Semantic Encoder	3xDownsampling	(ConvBN+ReLU)x2	kernel size=3 stride=1
		Maxpool(k=2,s=2)	kernel size=2 stride=2
	CBAM	AdaptiveAvgPool2d AdaptiveMaxPool2d	output_size=1
		Conv2d(k=7,s=1) sigmoid	kernel size=7 stride=1
AWGN	AWGN	SNR (dB)	1-20
Semantic Decoder	3xUpsampling	Upsampling	scale_factor=2
		(ConvBN+ReLU)x2	kernel size=2 stride=1

The spatial attention module is designed to identify which pixels in an RGB image are most informative for making classification [14]. This is achieved by using a similar mechanism as the channel attention module but applied at the pixel level, yielding a pixel-level attention weight map to represent the importance of each pixel. The implementation is illustrated in Fig. 4(c), where the selected informative channels tensor F' is first subjected to a channel-based maximum and mean pooling operation, and then two outputs are concatenated and passed through a 7×7 convolutional layer. Finally, the resulting feature map is activated using the sigmoid function to generate the spatial attention weight map $M_s(F') \in \mathbb{R}^{H \times W \times C}$, which is given by

$$\begin{aligned} M_s(F') &= \sigma(f^{7 \times 7}([AvgPool(F'); MaxPool(F')])) \\ &= \sigma(f^{7 \times 7}([F_{avg}^s; F_{max}^s])), \end{aligned} \quad (10)$$

where $f^{7 \times 7}$ and $[\cdot; \cdot]$ is the convolutional operation and concatenation operation, respectively. The F_{avg}^s and F_{max}^s of two-dimensional maps are generated through avgpooling and maxpooling. The spatial attention weight map $M_s(F')$ is used to select the informative pixels and as the input of the following downsampling layer.

The objective of the decoder is to accurately classify each pixel. Therefore, we utilize the cross entropy of the multi-class classification for each pixel as the loss function. For a batch of frames, the loss function of the entire LSSC system can be given by

$$L_{LSSC} = - \frac{\sum_{b=1}^B \sum_{l=1}^{H_l \times W_l} \sum_{i=1}^N (p_{b,l,i} \log(\hat{p}_{b,l,i}))}{S \times H_l \times W_l}, \quad (11)$$

where B is the batch size, H_l and W_l is the input image feature of one layer. $\hat{p}_{b,l,i}$ is the predicted probability that the pixel is classified as a category i in the b batch at layer l . $p_{b,l,i}$ is the classification indicator with a value of 0 or 1 and N is the number of categories. By the loss function, the LSSC system is trained by the end-to-end methods.

IV. SIMULATION AND PERFORMANCE ANALYSIS

To evaluate the performance of the proposed LSSC system, we use the dataset of Deep Automatic Portrait Matting for

training and testing [15]. In our simulation, the coefficients λ , μ , and ν that lead to the appropriate results are selected respectively 1, 0.1, and 0.2. We train the model using the training set from the image dataset, which contains 1530 training images and 170 test images, and every image is in RGB format. The learning rate η is 10^{-3} . The proposed LSSC system is trained under AWGN with the SNR varying randomly between 1 and 20 dB and the RMSprop algorithm is employed to optimize the model. The simulation parameters are listed in Table I. For comparison purposes, we use four baselines: a) UNet, b) Attention-UNet with an attention gate (AG) module proposed in [16], c) CBAM-UNet with a CBAM module in each downsampling process and d): the traditional modular wireless transmission mode, which adopts PNG for source coding, Low-Density Parity-Check Codes (LDPC) for channel coding, and QAM for modulation [17]. At the receiver, we employ the UNet to get the received image segmentations.

In Fig. 5, we show the relationship between the number of downsampling layers and the optimization objective Q of our proposed model. From Fig. 5, we can see that when the number of downsampling layers is 3, the optimization objective Q of the proposed model reaches the peak. When the number of downsampling layers is 2, the mIoU is lower which can not achieve high accuracy of image segmentation. And when the number of downsampling layers is 4, we can observe that the FLOPs and model size are bigger, which increases the burden on IoT devices. Fig. 5 demonstrates that in our proposed model architecture, adopting three number of downsampling layers can exploit the trade-off between accuracy and complexity, and can achieve the highest accuracy with limited calculations.

In Tabel II, we show the FLOPs, model size, and mIoU of different schemes. From Table II, we can see that the FLOPs of our proposed model is about 13 % lower than UNet. Reducing the computational complexity of the semantic segmentation model makes it possible for IoT devices to run the semantic codec. Meanwhile, the model size of our proposed codec is about 55 % less than UNet, because of the decrease in the number of downsampling layers. Table II also indicates that, among the compared methods, the proposed codec in this paper, achieves the minimal computational complexity and lowest model size while maintaining high segmentation performance. Meanwhile, Table II also shows that the size of the CBAM module is almost negligible, which can improve the model segmentation performance.

Fig. 6 shows how the mIoU changes with different channel SNR. We can see that as the channel SNR increases, the mIoU of both the LSSC system and the traditional schemes increase, and the LSSC system has a better performance at low SNR region because the semantic features contain contextual information. We can also observe that the mIoU of the LSSC system is more robust with the steadier mIoU, in contrast to the traditional schemes. The line of the traditional scheme sharply drops when the channel quality deteriorates because the damaged PNG image cannot be segmented. Besides, the reason for the steep line of the traditional scheme is that it encodes the image into flag and data bits, which would be

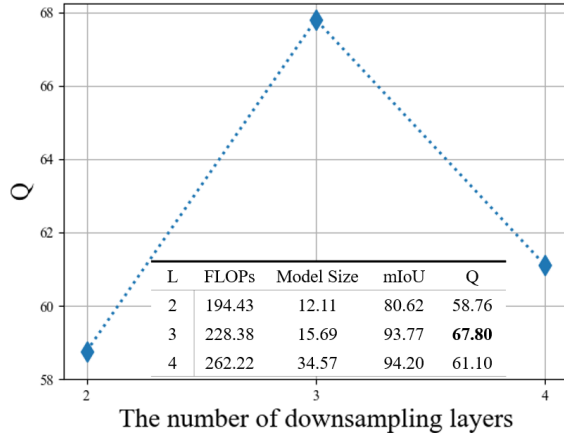


Fig. 5. The relationship between the number of downsampling layers and objective Q .

TABLE II
THE RELATIONSHIP BETWEEN mIoU AND FLOPs UNDER LSSC SYSTEM AND DIFFERENT MODELS

Methods	FLOPs	Model Size	mIoU(%)
UNet	262.12	34.53	94.07
Attention-UNet	266.54	34.88	94.18
CBAM-UNet	262.22	34.57	94.20
LSSC(Ours)	228.38	15.69	93.77

transmitted incorrectly in the low SNR region. The wrong flag bits will severely compromise the image reconstruction and damage the accuracy of image segmentation, which is known as the “cliff effect” [18]. From Fig. 6, we see that the “cliff effect” is avoided by the LSSC system. This is because the LSSC system can effectively extract image semantic features which contain contextual information, thus achieving more robust transmission against channel variation.

V. CONCLUSION

In this paper, we proposed an attention-based UNet enabled lightweight image communication (LSSC) system over the Internet of Things, which has lower computational complexity and smaller model size. To lessen the burden of IoT devices, the LSSC system transmits the trained semantic codec and semantic features between the edge server and IoT devices. Specifically, by employing the CBAM module in the semantic encoder, the proposed LSSC codec efficiently learns informative image information by identifying the most important channels and pixels. Simulation results show that compared to the traditional UNet, our system achieves lower FLOPs and smaller model size with a sacrifice of 3% accuracy. Moreover, the LSSC system performs better in the low SNR region than the traditional communication scheme. Therefore, our proposed LSSC system is a promising candidate for IoT devices, especially in the low SNR region.

REFERENCES

[1] S. E. Calvanese and B. Sergio, “6G networks: Beyond Shannon towards semantic and goal-oriented communications,” *Computer Networks*, vol. 190, pp. 107930, 2021.

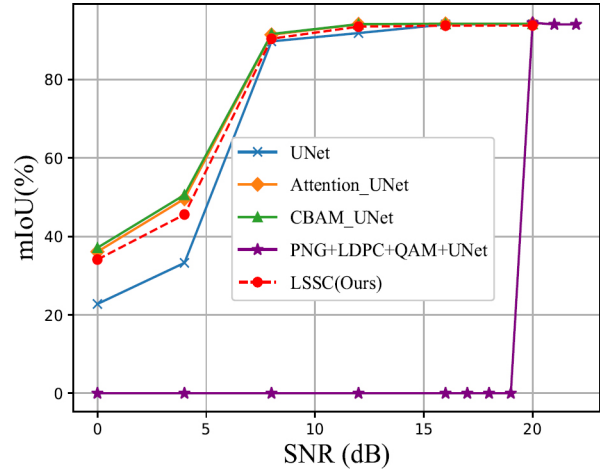


Fig. 6. mIoU comparison of the LSSC system.

[2] V. Sandeep Narayanan Kadan, M. Mohammad, H. Andreas, Y. Emre A. T. Tuomas, B. Johan, and W. Y-P Eric, “Toward smaller and lower-cost 5G devices with longer battery life: an overview of 3GPP release 17 redcap,” *IEEE Communications Standards Magazine*, vol. 6, no. 3, 2022.

[3] Wentao Gong, Haonan Tong, Sihua Wang, Zhaohui Yang, Xinxin He, and Changchuan Yin, “Adaptive bitrate video semantic communication over wireless networks,” 2023.

[4] H. Xie, Z. Qin, L. Ye, and J. Biing-Hwang, “Deep learning enabled semantic communication systems,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[5] Z. Weng and Z. Qin, “Semantic communication systems for speech transmission,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.

[6] H. Tong, Z. Yang, S. Wang, Y. Hu, S. Walid, and C. Yin, “Federated learning based audio semantic communication over wireless networks,” in *2021 IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, Dec. 2021, IEEE.

[7] H. Xie and Z. Qin, “A lite distributed semantic communication system for Internet of things,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, 2020.

[8] B. Du, H. Du, H. Liu, N. Dusit, X. Peng, J. Yu, M. Qi, and Y. Tang, “Yolo-based semantic communication with generative AI-aided resource allocation for digital twins construction,” *IEEE Internet of Things Journal*, 2023.

[9] N. Li, I. Alexandros, and Q. Zhang, “Attention-based feature compression for CNN inference offloading in edge computing,” *arXiv preprint arXiv:2211.13745*, 2022.

[10] D. Li, S. Yin, Y. Lei, J. Qian, C. Zhao, and L. Zhang, “Segmentation of white blood cells based on CBAM-DC-UNet,” *IEEE Access*, vol. 11, pp. 1074–1082, 2022.

[11] H. Huang, Z. Chen, Y. Zou, M. Lu, and C. Chen, “Channel prior convolutional attention for medical image segmentation,” *arXiv preprint arXiv:2306.05196*, Jun. 2023.

[12] R. Olaf, F. Philipp, and B. Thomas, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention—MICCAI Munich, Germany, October, 2015*.

[13] W. Sanghyun, P. Jongchan, L. Joon-Young, and K. I. So, “Cbam: Convolutional block attention module,” in *Proc. of the European conference on computer vision (ECCV)*, 2018.

[14] M. Guo, c. Lu, Q. Hou, Z. Liu, and M. Cheng, “Segnext: Rethinking convolutional attention design for semantic segmentation,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 1140–1156, 2022.

[15] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia, “Deep automatic portrait matting,” in *Computer Vision – ECCV 2016*, Oct. 2016.

[16] O. Ozan, S. Jo, F. L. Le, L. Matthew, H. Mattias, M. Kazunari, M. Kensaku, M. Steven, et al., “Attention u-net: Learning where to look for the pancreas,” *arXiv preprint arXiv:1804.03999*, Apr. 2018.

[17] Qiang Pan, Haonan Tong, Jie Lv, Tao Luo, Zhilong Zhang, Changchuan Yin, and Jianfeng Li, “Image segmentation semantic communication over internet of vehicles,” in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2023.

[18] B. Eirina, K. D. Burth, and G. Deniz, “Deep joint source-channel coding for wireless image transmission,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, 2019.