

Video Semantic Communication with Major Object Extraction and Contextual Video Encoding

Haopeng Li*, Haonan Tong*, Sihua Wang*, Nuocheng Yang*, Zhaohui Yang[†], and Changchuan Yin*

* Beijing Laboratory of Advanced Information Network, Beijing University of Posts and Telecommunications, Beijing, China

[†] College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China

Emails: {hpli, hntong, sihuawang, yangnuocheng, ccyin}@bupt.edu.cn, yang_zhaohui@zju.edu.cn

Abstract—This paper studies an end-to-end video semantic communication system for massive communication. In the considered system, the transmitter must continuously send the video to the receiver to facilitate character reconstruction in immersive applications, such as interactive video conference. However, transmitting the original video information with substantial amounts of data poses a challenge to the limited wireless resources. To address this issue, we reduce the amount of data transmitted by making the transmitter extract and send the semantic information from the video, which refines the major object and the correlation of time and space in the video. Specifically, we first develop a video semantic communication system based on major object extraction (MOE) and contextual video encoding (CVE) to achieve efficient video transmission. Then, we design the MOE and CVE modules with convolutional neural network based motion estimation, contextual extraction and entropy coding. Simulation results show that compared to the traditional coding schemes, the proposed method can reduce the amount of transmitted data by up to 25% while increasing the peak signal-to-noise ratio (PSNR) of the reconstructed video by up to 14%.

Index Terms—Semantic communications, video transmission, major object extraction, contextually encode.

I. INTRODUCTION

The development of B5G/6G communication has promoted the vigorous rise of new broadband interactive applications (e.g. metaverse), which need to collect and transmit massive sensing data in different modals [1]. For example, in the interactive application such as immersive video conference, the transmitter needs to send information including the characters, the actions, and the background to the receiver. However, the interactive application requires a large amount of data transmission [2] and strict transmission performance, such as stringent delay [3] and high quality of experience, which brings a huge challenge to the existing wireless communication system with limited resources [4]. Therefore, future interactive applications call for efficient video transmission technology, and semantic communication [5] is emerging as a high-efficiency transmission technology to improve the broadband interactive application performance.

Previous works have studied semantic communication techniques for different data types. In [6], the authors proposed

This work was supported in part by Beijing Natural Science Foundation under Grant L223027, in part by the National Natural Science Foundation of China under Grants 61671086 and 61629101, supported by National Key RD Program of China (Grant No.2023YFB2904804), in part by BUPT Excellent Ph.D. Students Foundation under Grant CX2021114, and China Scholarship Council.

a deep learning (DL) based semantic communication system to reduce the amount of text information data. Furthermore, the work in [7] presented the generative adversarial networks (GANs) based image semantic codec to reduce the amount of image transmission data by transmitting semantic information rather than original symbols. Extending the single user scenario, the authors in [8] proposed a DL based multi-user semantic communication system to transmit multi-modal data. However, the semantics used for transmitting text and image in [6]–[8] only considered the data properties for one moment, which ignored the semantics from the temporal correlation in the data stream. Different from text and image, video semantic extraction is challenging due to the correlations among multiple moments within the video data streams.

The prior works [9]–[11] have studied efficient video semantic communication systems. In [9], the authors proposed an end-to-end DL based video communication model that obtained the motion information among video frames to encode the video. The work in [10] presented a semantic video conferencing (SVC) model based on face key points transmission to express the face motions. The authors in [11] proposed a scheme that collected the strong video temporal correlation provided by the feature domain context under the deep video semantic transmission (DVST) model to achieve more efficient video transmission. However, for interactive applications such as video conferences, the static background information is often redundant and unnecessary to be transmitted in each frame. Therefore, removing redundant background information according to the need of reconstructing content in video transmission can reduce the amount of transmission data and decrease transmission delay, which has not been considered in [9]–[11].

To address the above issue, in this paper, we propose a video semantic communication system, which extracts the major object of the video for character reconstruction at the receiver. The main contributions of this paper are as follows:

(1) We develop a video semantic communication system through major object extraction (MOE)-contextual video encoding (CVE). In the considered system, the transmitter extracts the semantic information of the major object in the video, through removing the redundant static background, and then sends the semantic information of the major object to the receiver. The receiver decodes the semantic information and

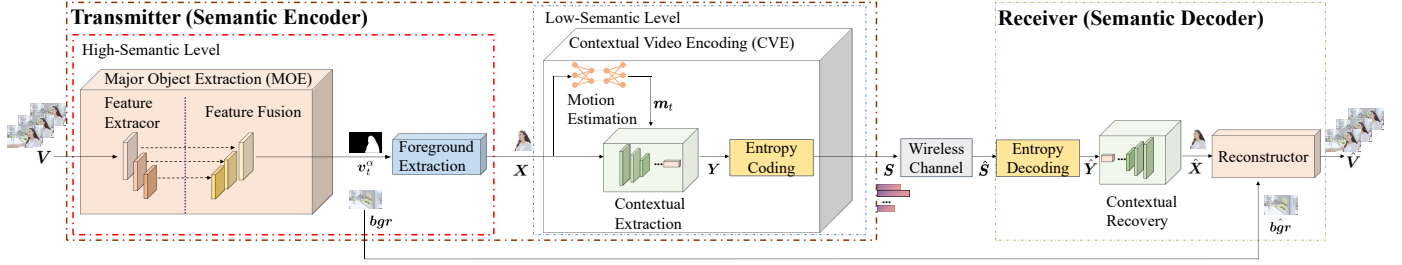


Fig. 1. The overall flowchart of the proposed MOE-CVE scheme.

synthesizes the background.

(2) We propose an efficient video semantic extraction method using hybrid high-level and low-level semantic information, where the high-level semantic is the major object information, and the low-level semantic is the spatial structure in the video frame. To achieve the above goal, we design the MOE module to extract the major object, and the CVE module to semantically encode video at a low-semantic level, in which context based spatial structure is extracted and encoded by entropy coding.

(3) Simulation results demonstrate that the proposed MOE-CVE scheme can reduce the amount of transmitted data by up to 25% while increasing peak signal-to-noise ratio (PSNR) by up to 14%, compared to the traditional coding schemes.

The rest of this paper is organized as follows. Section II introduces the system model. Section III provides detailed descriptions of the proposed network architectures. Simulation results are presented in Section IV. Conclusion is drawn in Section V.

II. SYSTEM MODEL

We consider a video semantic communication system to support interactive video conference, which includes a transmitter [12] and a receiver as shown in Fig. 1. The receiver needs to accurately reconstruct the character information, while the static background information does not need to be updated in time, and only limited wireless bandwidth is available to support the massive communication. To this end, the system requires efficient video transmission techniques by extracting and transmitting video semantic information. To achieve this goal, we consider a hybrid high-level and low-level semantic extraction by removing the redundant static video background information, which mainly includes three steps. First, we extract the major object of the video at a high-semantic level, where video frames are divided into foreground and background, the static background is transmitted only once, and the foreground needs to be updated in real time. Second, the foreground is further encoded through temporal and spatial correlations at a low-semantic level. Finally, given the transmitted foreground semantic feature, the receiver reconstructs the video frames by combining the character in the foreground and the one-frame static background into a complete video.

Next, we first introduce the MOE at the high-semantic level. Then, we present our CVE at the low-semantic level.

Subsequently, we introduce our channel model. Finally, we introduce the method for decoding and reconstructing video.

A. Major Object Extraction

Denote the raw video data sensed at the transmitter by $V = \{v_1, v_2, \dots, v_T\}$, where $v_t \in \mathbb{R}^{H \times W \times C}$ is the t -th video frame. H and W are the height and width of the video frame, respectively. C is the number of channels. The MOE module is used to extract the major object from V , and the major object is the foreground component. Denote the foreground estimation tensor by $v_t^\alpha \in \mathbb{R}^{H \times W \times 1}$, and v_t^α can be given by

$$v_t^\alpha = f_{MOE}(v_t), \quad (1)$$

where $f_{MOE}(\cdot)$ is the function of the MOE module.

Subsequently, we employ the foreground extraction to obtain the foreground series $X = \{x_1, x_2, \dots, x_T\}$ with v_t^α , where the t -th frame $x_t \in \mathbb{R}^{H \times W \times C}$ undergoes background pixel removal, and can be given by

$$x_t = v_t^\alpha \odot v_t + (1 - v_t^\alpha) \odot b, \quad (2)$$

where b is a white pixel tensor to represent the blank background, and \odot is the Hadamard product. Given v_t and x_t , the background can be extracted with $bgr = v_t - x_t$, where the background $bgr \in \mathbb{R}^{H \times W \times C}$ can be further completed by image generation techniques.

B. Contextual Video Encoding

To take advantage of the temporal and spatial correlations of the foreground to further reduce the amount of data required for transmission, we transform x_t from the pixel domain to the semantic feature y_t in the latent domain and then encode y_t to the variable-length code words s_t for transmission. In the CVE module, we first use the motion estimation network to learn the motion vector $m_t \in \mathbb{R}^{H \times W \times (C-1)}$ between the current frame x_t and the previous frame x_{t-1} , which is given by

$$m_t = f_{Est}(x_t, x_{t-1}), \quad (3)$$

where $f_{Est}(\cdot)$ denotes the motion estimation function. Since neural network can effectively refine the video features in the latent domain [13], given the motion vector m_t , we can refine the original video foreground series X into the semantic feature series $Y = \{y_1, y_2, \dots, y_T\}$, which is in a latent domain representation with smaller data size as shown in Fig. 1. The relationship between y_t and x_t can be given by

$$y_t = f_{CE}(x_t, m_t), \quad (4)$$

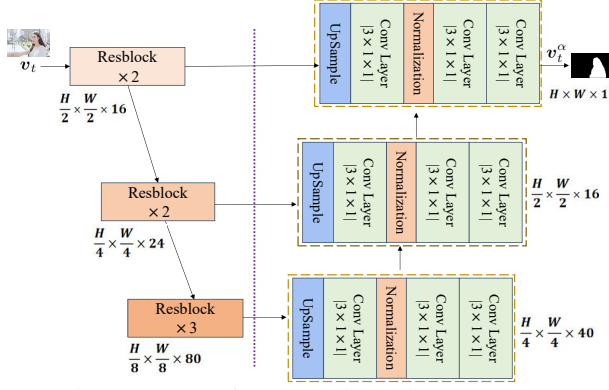


Fig. 2. The structure of MOE, where the Conv Layer means the convolutional layer with $[kernel \times stride \times padding]$, and the Resblock means the residual block.

where $f_{CE}(\cdot)$ denotes the contextual extraction function, and $\mathbf{y}_t \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 96}$ is the semantic feature of a single frame. Finally, \mathbf{y}_t is encoded into code words streaming $\mathcal{S} = \{s_1, s_2, \dots, s_T\}$ by entropy coding algorithm to transmit through wireless channel, which can be given by

$$s_t = f_{en}(\mathbf{y}_t), \quad (5)$$

where $f_{en}(\cdot)$ denotes the function of encoding, and $s_t \in \mathbb{R}^{k_t}$ is the variable-length code words with k_t being the length of s_t . We use $R = \frac{1}{T} \sum_{t=1}^T \frac{k_t}{H \times W \times C}$ as the channel bandwidth ratio (CBR) [11] to describe the average coding rate of \mathcal{X} .

C. Wireless Channel

When transmitted over a wireless channel, encoded code words streaming \mathcal{S} suffers transmission impairments that include distortion and noise. Assume that the video transmission uses a single end-to-end wireless link, the received signal can be given by

$$\hat{\mathcal{S}} = h\mathcal{S} + \sigma, \quad (6)$$

where $\hat{\mathcal{S}}$ is the received semantic feature code words streaming with transmission impairment, h is the fading channel coefficient, and $\sigma \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ denotes Gaussian channel noise with σ^2 being noise variance and \mathbf{I} being identity matrix.

D. Receiver

The receiver includes the entropy decoding, the contextual recovery, and the reconstructor modules. The entropy decoding is used to decode the received code words streaming $\hat{\mathcal{S}}$ to the semantic feature $\hat{\mathbf{Y}}$, which is given by

$$\hat{\mathbf{Y}} = f_{de}(\hat{\mathcal{S}}). \quad (7)$$

Then the contextual recovery is used to obtain $\hat{\mathbf{X}}$ from semantic feature $\hat{\mathbf{Y}}$, given by

$$\hat{\mathbf{X}} = f_{CR}(\hat{\mathbf{Y}}). \quad (8)$$

The reconstructor is used to combine the static background that is directly provided at the receiver. In the considered system, the static background can be reused for multiple video frames. As a result, we transmit one static background frame and use it as the background of the reconstructed video \hat{bgr} . Finally, we combine the video frames back to reconstructed video $\hat{\mathbf{V}} = \{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_T\}$, as shown in Fig. 1.

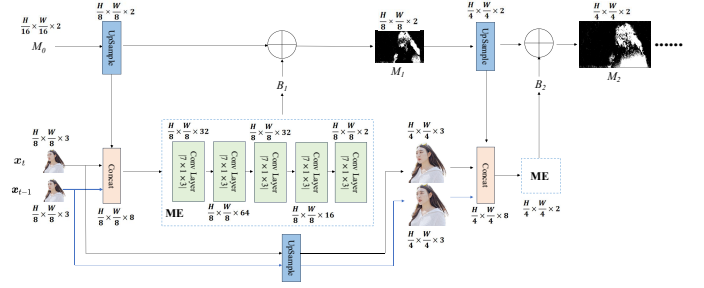


Fig. 3. Optical flow estimation network.

III. NETWORK ARCHITECTURES

In this section, we design the system module to implement the proposed system jointly considering accurate semantic extraction and various channel conditions. In particular, we first extract the high-level semantic in each single frame via neural network based MOE in semantic encoder. Then, we extract the low-level semantic through CVE in semantic encoder, considering correlations between frames, and adaptive various length coding, thus reducing the transmitted data amount. Finally, the semantic decoder and the whole training algorithm are illustrated.

A. MOE in Semantic Encoder

The MOE module is deployed to perform character extraction from the input video frames. It first recognizes the major object, and then removes the static background. In this way, we consider the object-level content as the high-level semantics. To accurately recognize the major object, we extract features at different scales which include object shapes and locations. We use MOE module to generate v_t^α in (1). As shown in Fig. 2, we use MobileNetV3-Large [14] as the backbone of MOE module, which is composed of a feature extractor and a feature fusion. The feature extractor consists of residual blocks to extract features at $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ scales of the video frame, respectively, to accurately locate video major object. The feature fusion part fuses features with different scales via convolutional layers to generate v_t^α , as shown in Fig. 2.

To improve the accuracy of foreground estimation, the loss function of MOE module to output v_t^α is given by

$$\mathcal{L}_{MOE} = \|v_t^\alpha - v_t^{\alpha*}\|_1 + \mathcal{L}_{lap}^\alpha + 5 \left\| \frac{dv_t^\alpha}{dt} - \frac{dv_t^{\alpha*}}{dt} \right\|_2, \quad (9)$$

where $v_t^{\alpha*}$ is the ground-truth of v_t^α , and

$$\mathcal{L}_{lap}^\alpha = \sum_{i=1}^5 \frac{2^{i-1}}{5} \|L_{pyr}^i(v_t^\alpha) - L_{pyr}^i(v_t^{\alpha*})\|_1 \quad (10)$$

is the Laplacian loss [15] with $L_{pyr}^i(\cdot)$ being the output of the i -th pyramid layer [15].

B. CVE in Semantic Encoder

In CVE, we first use motion vector m_t to refine the semantic feature \mathbf{y}_t from x_t with contextual extraction. Then \mathbf{y}_t is encoded as variable length code words s_t , based on the entropy estimation of \mathbf{y}_t . Next, we will introduce the motion

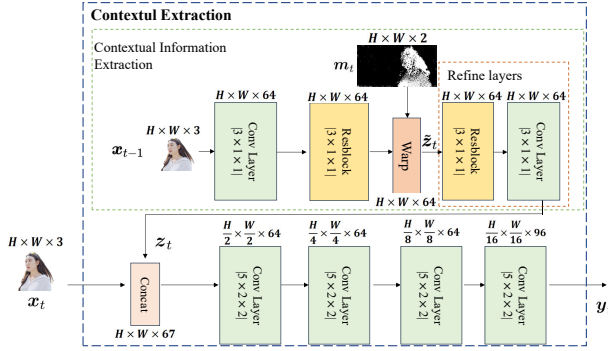


Fig. 4. The structure of contextual extraction.

estimation, the contextual extraction, and the entropy encoding in turn.

1) *Motion Estimation*: Since the foreground has strong temporal and spatial correlations between successive video frames, the low-level semantic features (temporal and spatial correlations) are further extracted by CVE module to reduce transmission overhead. To obtain the contextual information $z_t \in \mathbb{R}^{H \times W \times 64}$ which is the combination of temporal and spatial correlations between successive video frames, we first use convolutional neural network (CNN) to estimate motion vector m_t . Compared with traditional motion estimation (ME) methods, neural network can integrate multi-dimensional information such as color, texture, and depth from videos to improve the accuracy of motion estimation, so we estimate the motion vector m_t with CNN. ME is position change prediction of the pixel blocks. In the ME process, we use the optical flow estimation network [16] to estimate the motion vector m_t between the previous frame x_{t-1} and the current frame x_t . Concretely, the optical flow estimation network consists of four ME modules to form four pyramid levels [16]. As shown in Fig. 3, the optical flow estimation uses a coarse-to-fine spatial pyramid structure to learn residual flow $B_k \in \mathbb{R}^{2^{4-k} \times \frac{W}{2^{4-k}} \times 2}$ at each pyramid level, and the optical flow $M_k \in \mathbb{R}^{2^{4-k} \times \frac{W}{2^{4-k}} \times 2}$ at the k -th pyramid level is given by

$$M_k = u(M_{k-1}) + B_k, \quad k = 1, 2, 3, 4, \quad (11)$$

where $u(\cdot)$ is the upsampling operation, B_k is the residual flow output by the ME module, and $M_0 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 2}$ is a zero tensor as the initial value. We use the ME module to calculate the residual optical flow B_k of the current level (k -th level). The residual optical flow B_k is input successively to rectify the optical flow M_k at each level. ME is the last step of optical flow estimation, and finally output the last optical flow as the motion vector, given by $m_t = M_4$.

2) *Contextual Extraction*: As shown in Fig. 4, we use the obtained m_t and the previous frame to extract the initial contextual information $\tilde{z}_t \in \mathbb{R}^{H \times W \times 64}$ with convolutional layers and residual blocks. Here, to compensate for the discontinuity [11], we use refine layers to refine the final contextual information $z_t \in \mathbb{R}^{H \times W \times 64}$ which contains temporal and spatial correlations. Then we use z_t to extract the semantic feature y_t with CNN layers, where y_t is the mapping feature of x_t in the latent domain. Instead of directly encoding the video

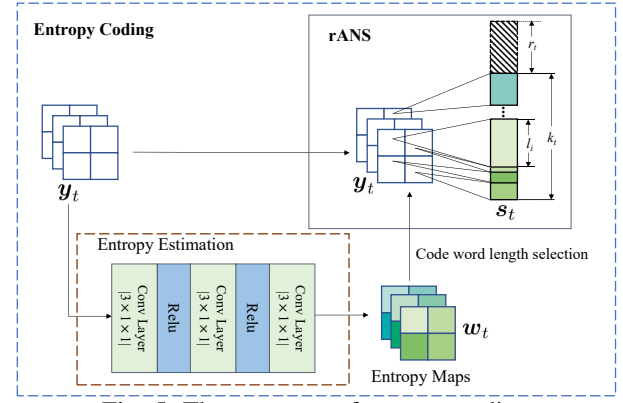


Fig. 5. The structure of entropy coding.

frame in the pixel domain in traditional methods, we encode semantic feature y_t taking into account both the temporal and spatial correlations in the regions across video frames, which is more conducive to the subsequent allocation of coding weights.

3) *Entropy Coding*: Given y_t , we encode the semantic feature y_t into the variable-length code words s_t with entropy coding. As shown in Fig. 5, the entropy coding consists of the entropy estimation and a range asymmetric numeral system (rANS) arithmetic encoder [17]. The entropy estimation is composed of the convolutional layers and the relu layers, which assigns weights to the elements of semantic feature y_t and generates a weight vector $w_t \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 96}$ [18]. Each element in w_t represents a weight that maps the elements in y_t to the code words space $\Omega \in \mathbb{R}^{64 \times 103}$. In detail, each element in Ω follows the Laplacian distribution. The element in w_t determines the mapped position in Ω of the element in y_t , thus deciding the code word length k_t . In this way, a greater estimated entropy in y_t is mapped by w_t to a longer code length in Ω . The code word length k_t of s_t is the sum length of the code words of all the elements in y_t , which is given by $k_t = \sum_{i=1}^{\frac{H}{16} \times \frac{W}{16} \times 96} l_i$, where l_i is the code word length of i -th element in y_t . Then the code words streaming S is transmitted to the wireless channel. As shown in Fig. 5, we only encode characters for transmission, saving the code word length r_t required for transmitting the redundant background.

C. Semantic Decoder

We first decode \hat{s}_t to \hat{y}_t by arithmetic decoding in rANS as shown in Fig. 1, which is the inverse process of entropy coding. Then we use the contextual recovery which consists of the convolutional layers, the upsample layers, and the residual blocks to recover \hat{y}_t to \hat{x}_t . The convolutional layers and the residual blocks are used to reconstruct the video frames from features, and the upsample layers are used to gradually recover to the same size as the original video. Finally the reconstructor combines the final video \hat{V} by $\hat{v}_t = \hat{x}_t \odot \hat{x}_t^\alpha + (1 - \hat{x}_t^\alpha) \odot \hat{b}gr$ which is similar to (2). The semantic decoder is trained offline end-to-end with the CVE module. In order to control the trade-off between the distortion d_t and the bit cost k_t , the loss of the semantic decoder is

$$\mathcal{L}_{CVE} = k_t + \lambda \cdot d_t, \quad (12)$$

Algorithm 1 Proposed MOE-CVE Algorithm

- 1: **Initialize** the parameters of MOE (θ) and CVE (ω).
 - 2: train MOE with (9):
 - 3: **for** each MOE training step **do**
 - 4: $g_t \leftarrow \nabla_{\theta} \mathcal{L}_{MOE}(\theta_t)$ ▷ calculate gradient
 - 5: $e_t \leftarrow \beta_1 e_{t-1} + (1 - \beta_1) g_t$ ▷ first moment estimation
 - 6: $s_t \leftarrow \beta_2 s_{t-1} + (1 - \beta_2) g_t^2$ ▷ second moment estimation
 - 7: $\hat{e}_t \leftarrow \frac{e_t}{1 - \beta_1^t}$ ▷ deviation correction
 - 8: $\hat{s}_t \leftarrow \frac{s_t}{1 - \beta_2^t}$ ▷ deviation correction
 - 9: $\theta_{t+1} \leftarrow \theta_t - \frac{\xi}{\sqrt{\hat{s}_t + \epsilon}} \hat{e}_t$ ▷ update parameter
 - 10: **end for**
 - 11: train CVE with (12):
 - 12: **for** each CVE training step **do**
 - 13: $\omega_{t+1} \leftarrow \omega_t - \eta \nabla_{\omega} \mathcal{L}_{CVE}(\omega_t)$
 - 14: **end for**
 - 15: Receiver reconstructs the video frames.
 - 16: **Output:** The reconstructed video \hat{V} .
-

where λ is the coefficient to control the trade-off between the distortion d_t and the bit cost k_t , d_t is peak signal-to-noise ratio (PSNR) of \hat{v}_t and v_t . The PSNR can be given by

$$PSNR(\hat{v}_t, v_t) = 10 \log_{10} \left(\frac{1}{MSE(\hat{v}_t, v_t)} \right), \quad (13)$$

where $MSE(\hat{v}_t, v_t) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n [\hat{v}_t(i, j) - v_t(i, j)]^2$, with m and n respectively being the number of pixels horizontally and vertically.

D. MOE-CVE Training Algorithm

In the proposed system, each module is independent, and we train the MOE module with parameter θ and the CVE module with parameter ω separately. In MOE training, we use Adam optimizer, where g_t is the gradient of \mathcal{L}_{MOE} ; e_t is the first-moment estimation; s_t is the second-moment estimation; ϵ is 10^{-8} . The learning rate ξ and η for both of MOE and CVE modules is set to 10^{-4} . After the semantic feature \mathbf{Y} is transmitted through the AWGN channel, the receiver uses the received noisy feature $\hat{\mathbf{Y}}$ for training. The entropy decoding and the contextual recovery are simultaneous training with CVE in an end-to-end method. The overall training process is summarized in Algorithm. 1.

IV. SIMULATION AND PERFORMANCE ANALYSIS

In our simulations, a video semantic communication system consisting of a transmitter and a receiver is considered. Details of all network layers are shown in Figs.2-5. The MOE model is trained with video dataset VideoMatte240K [19]. The dataset provides 484 video clips, and we divide the dataset into 475/4/5 clips for train/val/test splits. The CVE module uses Vimeo-90k septuplet dataset [20] as the training data. The testing data uses the character dataset provided by [14] which contains a variety of character video frames with $H=1920$, $W=1024$ and $C=3$. We train 4 models with different $\lambda \in \{256, 512, 1024, 2048\}$ with fading channel with $h=0.9$ and $SNR=15\text{dB}$ AWGN. For comparison purposes, we experiment three baselines: a) Advanced

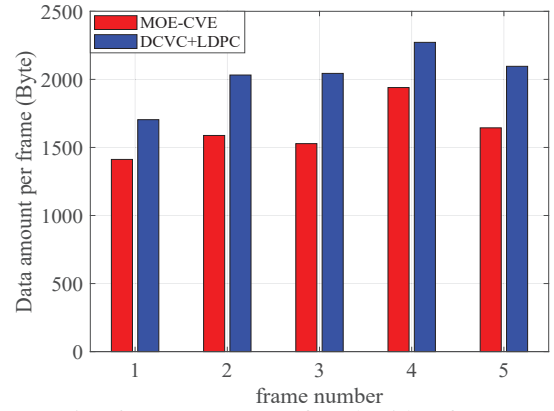


Fig. 6. Data amount of each video frame.

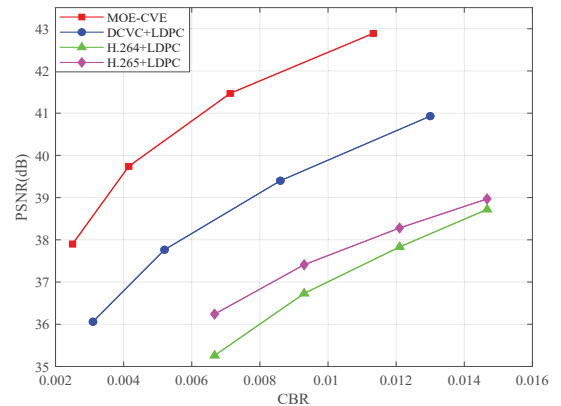


Fig. 7. PSNR of different algorithms under different CBR at SNR = 15dB.

video coding (H.264) + low-density parity-check (LDPC): H.264 for source coding combined with 1/2 rate LDPC code for channel coding, b) High-efficiency video coding (H.265) + LDPC and c) Deep contextual video compression (DCVC) + LDPC provide the reference here is good if possible.

In Fig. 6, we show the transmitted data amount of different video frames in our method and DCVC+LDPC. From Fig. 6, we can observe that both methods use a small amount of data to transmit the original video frame, while our method reduces more data amount. From Fig. 6, we see that our method reduces the data amount of each frame of video by nearly 25%, compared to DCVC+LDPC. This is due to the fact that MOE removes the redundant background information and focuses on the major object. Fig. 6 demonstrates that the proposed MOE-CVE method significantly reduces the amount of transmitted data.

In Fig. 7, we show how the PSNR changes as the CBR varies. From Fig.7 we can observe that, as the transmission CBR increases, the transmission PSNR increases as expected, while our MOE-CVE increases more fastly. From Fig.7 We can also observe that, when the CBR is 0.012, the PSNR of our method is 14% higher than H.264+LDPC, and 12% higher than H.265+LDPC. Moreover, compared to DCVC+LDPC, our method also achieves a improvement in PSNR. The improvement is due to the fact that the other algorithms encode all the details of the video frame while our MOE-CVE concentrates

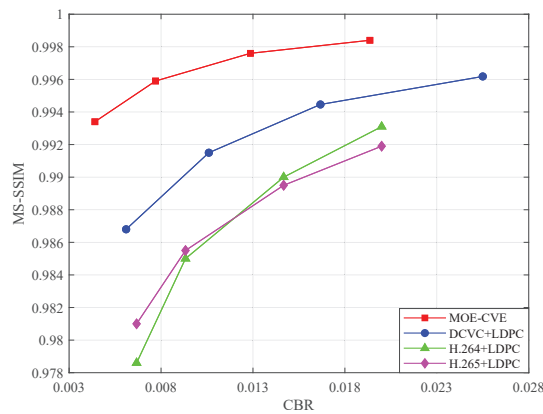


Fig. 8. MS-SSIM of different algorithms under different CBR at SNR = 15dB.

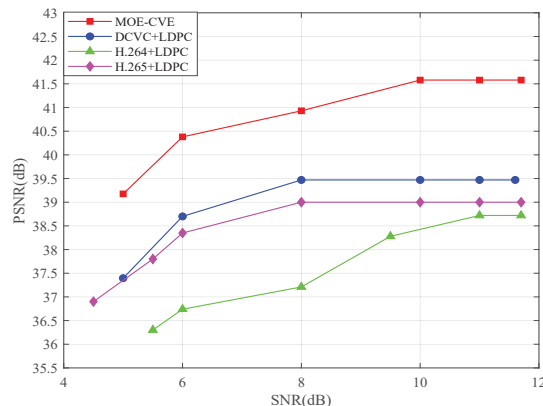


Fig. 9. PSNR of different algorithms under different SNR with CBR=0.015 for H.264/H.265 schemes, and 0.008 for others.

on the semantic features of the major object thus achieving better reconstruction performance under the same CBR. Fig. 7 demonstrates that the proposed method can improve the video reconstruction accuracy.

In Fig. 8, we show how the Multi-scale structural similarity (MS-SSIM) [18] changes as the CBR varies. From Fig. 8 we observe that as the CBR increases, the transmission MS-SSIM increases as expected. From Fig. 8 we can also observe that our method increases transmission MS-SSIM by nearly 0.014, compared to H.264+LDPC. Besides, the MS-SSIM of our method varies flatter than that of the other methods. This is because our method removes the static background, and encodes taking into the correlations, thus improving the MS-SSIM significantly.

Fig. 9 shows how the PSNR changes as the wireless channel SNR varies. From Fig. 9, we see that the PSNR of our method is higher than that of other methods. As the SNR increases, the PSNR of both our method and the baselines increase, while our MOE-CVE method increases more fastly. This is due to the fact that our method focuses on the major object, and all data amount is used to transmit the semantic feature of the major object, rather than to transmit both the background information. From Fig. 9, we can also observe that the PSNR of the H264+LDPC remains unchanged when SNR is larger than 10 dB. When the SNR is greater than 8 dB, the PSNR

of our method remains unchanged. This demonstrates that our method is more robust against channel noise, especially under low SNR regions.

V. CONCLUSION

In this paper, we proposed a MOE-CVE for video semantic communication. In particular, we designed a MOE module to extract the major object of the video and a CVE module to encode the semantic information of the major object. Simulation results demonstrated that the proposed MOE-CVE scheme can reduce by up to 25% transmission data while maintaining PSNR and MS-SSIM, compared to the traditional coding schemes. Moreover, the proposed MOE-CVE scheme exhibited better robustness to channel impairments than baseline methods.

REFERENCES

- [1] Z. Yang, M. Chen, Z. Zhang, and C. Huang, "Energy efficient semantic communication over wireless networks with rate splitting," *IEEE Journal on Selected Areas in Communications*, May 2023.
- [2] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, "Recent advances of resource allocation in network function virtualization," *IEEE Transactions on Parallel and Distributed Systems*, Feb. 2021.
- [3] W. Gong, H. Tong, S. Wang, Z. Yang, X. He, and C. Yin, "Adaptive bitrate video semantic communication over wireless networks," *arXiv preprint arXiv:2308.00531*, Aug. 2023.
- [4] A. Pratap, R. Gupta, V. S. S. Nadendla, and S. K. Das, "Bandwidth-constrained task throughput maximization in IoT-enabled 5G networks," *Pervasive and Mobile Computing*, Nov. 2020.
- [5] H. Tong, Z. Yang, S. Wang, Y. Hu, W. Saad, and C. Yin, "Federated learning based audio semantic communication over wireless networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2021.
- [6] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, Apr. 2021.
- [7] D. Huang, X. Tao, F. Gao, and J. Lu, "Deep learning-based image semantic coding for semantic communications," in *Proc. Global Communications Conference (GLOBECOM)*, Dec. 2021.
- [8] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE Journal on Selected Areas in Communications*, Sept. 2022.
- [9] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "Dvc: An end-to-end deep video compression framework," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.
- [10] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless semantic communications for video conferencing," *IEEE Journal on Selected Areas in Communications*, Jan. 2023.
- [11] S. Wang, J. Dai, Z. Liang, K. Niu, Z. Si, C. Dong, X. Qin, and P. Zhang, "Wireless deep video semantic transmission," *IEEE Journal on Selected Areas in Communications*, Jan. 2023.
- [12] P. Si, J. Zhao, K.-Y. Lam, and Q. Yang, "Uav-assisted semantic communication with hybrid action reinforcement learning," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Dec. 2023.
- [13] J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, and P. Zhang, "Nonlinear transform source-channel coding for semantic communications," *IEEE Journal on Selected Areas in Communications*, Aug. 2022.
- [14] S. Lin, L. Yang, I. Saleemi, and S. Sengupta, "Robust high-resolution video matting with temporal guidance," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022.
- [15] Q. Hou and F. Liu, "Context-aware image matting for simultaneous foreground and alpha estimation," in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.
- [16] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Aug. 2017.
- [17] J. Duda, "Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding," *arXiv preprint arXiv:1311.2540*, Feb. 2014.
- [18] Z. Bao, H. Liang, C. Dong, C. Li, X. Xu, and P. Zhang, "Mdvsc-wireless model division video semantic communication," *arXiv preprint arXiv:2308.05338*, May. 2023.
- [19] S. Lin, A. Ryabtsev, S. Sengupta, B. L. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Real-time high-resolution background matting," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Aug. 2021.
- [20] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, Feb. 2019.